Enrico Corradini

got his Bachelor's Degree in Computer Science and Automation Engineering in 2017 and his Master's Degree in Computer Science and Automation Engineering (summa cum laude) in 2019 at the Polytechnic University of Marche. He attended the PhD course in Information Engineering at the Department of Information Engineering of the same University from 2019 to 2022.

The studies and research activities of Enrico Corradini were carried out in cooperation with high experienced researchers belonging to the Polytechnic University of Marche, University "G. D'Annunzio" of Chieti-Pescara, University of Calabria, and University of Pavia.

The main goal of his research is to demonstrate the possibility of uniformly modeling people and things through (Social) Network Analysis. To this end, different scenarios are taken into consideration, namely Online Social Networks, the Internet of Things, Blockchains, neural networks, and so forth. For each of them, several open problems in the literature are addressed and appropriate solutions are proposed, based on the use of concepts, measures and approaches derived from (Social) Network Analysis. The various solutions are tested and compared with the related literature highlighting, for each of them, the strengths, weaknesses, similarities, and differences.

Enrico Corradini is author of many scientific papers published in prestigious International Journals (such as Information Sciences, Information Processing and Management, Future Generation Computer Systems, Expert Systems with Applications).

Enrico Corradini has been a Member of the Technical Program Committee and Session Chair for some International Conferences. He is also a reviewer of scientific papers for international journals, conferences, and workshops (such as Information Processing and Management, Multimedia Tools and Applications and Computer Communications).

He has participated to several research projects with research institutions and companies and contributed to the realization of many prototype systems.

He has contributed to the didactic activities at the Polytechnic University of Marche. In particular, he has given several lectures in Mobile Programming and Software Engineering. He has been supervisor of several Bachelor Theses and some Master Theses. He also has held several courses of Big Data Analytics, Machine Learning and Social Network Analysis to different companies.

Finally, he has won several awards, thanks to his skills in mobile programming.

DOTTORATO di RICERCA in INGEGNERIA dell'INFORMAZIONE
SCUOLA DI DOTTORATO in SCIENZE dell'INGEGNERIA
Università Politecnica delle Marche

In recent years, information and networks have become the new core of economic development. They represent two of the most important and valuable assets for any public or private organization. Their presence in our society is declined in a variety of ways, ranging from Online Social Networks to IoT devices, from Blockchains to machine and deep learning systems. Managing all these scenarios is extremely complex, and many researchers are striving to address this challenging issue. This thesis aims to provide a contribution in this setting. It starts from the idea that the previous scenarios have a common root in networking. Therefore, complex network theory, and more generally graph theory, can provide the models and the techniques to uniformly represent and handle such scenarios that might seem extremely heterogeneous at a first glance. The applications of the results of this thesis are multiple and range from user profiling to information diffusion, from Industry 4.0 to the detection of possible organized groups speculating on cryptocurrencies, from the early discovery of dangerous challenges in TikTok to the efficient and effective management of privacy in a Multi-IoT context.

Enrico CORRADINI

Networking people and things: scenarios, models, and approaches

DOCTORAL SCHOOL IN ENGINEERING SCIENCE
POLYTECHNIC UNIVERSITY OF MARCHE

DRINF
DOTTORATO DI RICERCA IN INGEGNERIA DELL'INFORMAZIONE

Enrico Corradini

NETWORKING PEOPLE AND THINGS: SCENARIOS, MODELS, AND APPROACHES

Supervisor: Prof. Domenico Ursino
S.S.D. ING-INF/05
XXXV cycle

DOCTORAL SCHOOL IN ENGINEERING SCIENCE
POLYTECHNIC UNIVERSITY OF MARCHE

DEPARTMENT OF INFORMATION ENGINEERING
(DII)

PHD IN
INFORMATION ENGINEERING

S.S.D. ING-INF/05
XXXV CYCLE

NETWORKING PEOPLE AND THINGS: SCENARIOS, MODELS, AND APPROACHES

CANDIDATE
Dr. Enrico CORRADINI

SUPERVISOR
Prof. Domenico URSINO

COORDINATOR
Prof. Franco CHIARALUCE

ANCONA, FEBRUARY 2023

ENRICO CORRADINI

NETWORKING PEOPLE AND THINGS: SCENARIOS, MODELS, AND APPROACHES

The Teaching Staff of the PhD course in

INFORMATION ENGINEERING

consists of:

Franco CHIARALUCE (coordinator)
Marco BALDI
Andrea BONCI
Laura BURATTINI
Stefania CECCHI
Massimo CONTI
Paolo CRIPPA
Diego D'ADDA
Andrea DI DONATO
Claudia DIAMANTINI
Aldo Franco DRAGONI
Marco FARINA
Sandro FIORETTI
Oriano FRANCESCANGELI
Emanuele FRONTONI
Ennio GAMBI
Donato IACOBUCCI
Gianluca IPPOLITI
Sauro LONGHI
Liana LUCCHETTI
Adriano MANCINI
Valter MARIANI PRIMIANI
Fabrizio MARINELLI
Franco MOGLIE
Andrea MONTERIÙ
Antonio MORINI
Gianluca MORONCINI
Simone ORCIONI
Giuseppe ORLANDO
Valentina ORSINI
Francesco PIAZZA
Luca PIERANTONI
Paola PIERLEONI
Ornella PISACANE
Domenico POTENA
Paola RUSSO
David SCARADOZZI
Luca SPALAZZI
Susanna SPINSANTE
Stefano SQUARTINI
Claudio TURCHETTI
Domenico URSINO
Francesco VITA
Primo ZINGARETTI

It is perilous to confuse what you are made to do with what you choose to do.

(Sylvanas Windrunner)

Foreword

The 21st century society can be defined as "the information and network society". These two assets have now become pervasive and are the new engine of the economy, not only for services but also in agriculture and industry. Just to give some statistics of the phenomenon, we have that 4.7 billion people in the world access at least one social medium, and this number has increased by 227 million in the last 12 months. Moreover, 59% of the world's population uses social media, and this number rises to 75.5% if we consider people over 13 years old. Finally, 93.6% of Internet users access social media. At the same time, there are 7.74 billion IoT devices currently connected. Notably, the number of connected IoT devices surpassed the number of connected people in 2020. In 2025, it is expected to have 41.6 billion IoT devices connected in the world. Still, the blockchain industry has an annual growth rate of 56.3%, with a worth reaching $163.83 billion in 2029. There are currently 170 million blockchain wallets in the world and 10% of the world's population owns cryptocurrencies. In turn, all these connections generate a huge amount and variety of data, which is not even close to what past generations had to (and could) handle. In fact, it is estimated that by 2025 the world will generate 181 zettabytes of data. It is also estimated that 97.2% of organizations are investing in big data and artificial intelligence, with the total worth currently standing at $49 billion. Even more interestingly, by 2026 this worth will reach $268.4 billion, which is 12% of the Compound Annual Growth Rate (CAGR).

Managing these phenomena, which have a common root in networking, is increasingly complex and poses challenging issues to researchers in several areas including computer engineering, computer science, statistics, telecommunications, as well as sociology and economics.

Enrico Corradini's PhD thesis is set in this context and aims to provide a contribution in addressing this issue. It starts from the idea that graphs represent an extremely powerful and, at the same time, very flexible model to represent all networking phenomena. Moreover, graph theory, in particular complex network theory,

has developed in recent years a very large body of knowledge that can be used to solve open issues in this area. To demonstrate this assumption, Enrico Corradini's PhD thesis considers a wide variety of problems that can be summarized in two main strands, namely networking people and networking things. For each of these problems, the thesis provides a suitable modeling based on complex networks and one or more solutions that fully exploit that modeling.

In addition to the specific technical merits, which the reader can evaluate as she/he proceed to read the various chapters, Enrico Corradini's approach has the merit of defining a uniform modeling technique and a way of proceeding to handle seemingly very heterogeneous problems ranging from information diffusion to blockchain, from Industry 4.0 to privacy. For each problem considered, it provides a complete description of the state of the art, clearly describes the proposed approach to its solution, and presents an experimental campaign aimed at assessing the goodness of the latter.

For this reason, I consider Enrico Corradini's PhD thesis an excellent piece of work. The general approach is methodologically and scientifically sound, as evidenced by the numerous papers already published by the author and his colleagues in various journals. I think this thesis can be very useful both to researchers, who investigate the issues it addresses, and to practitioners, who can exploit the experimental results presented in it in the context in which they operate.

In my role as Advisor of Enrico Corradini's PhD thesis, I had the privilege of being able to follow the entire development of the research path that led Enrico to achieve the excellent results described in this thesis. And here, in writing this brief preface, I'm pleased to attest the quality, continuity and passion that Enrico has put, and continues to put, in his research activities. At the end of these three wonderful years, I feel I can say with certainty that Enrico has achieved all the goals we set beforehand, when we started this adventure.

Prof. Domenico Ursino,
Università Politecnica delle Marche

Preface

This book is my PhD thesis, which reports all the research done in three years at the Department of Information Engineering of the Polytechnic University of Marche from 2019 to 2022, under the supervision of Prof. Domenico Ursino.

During these three years, I had the opportunity to work with high experienced professors and researchers, such as Prof. Domenico Ursino himself, Prof. Antonino Nocera, Prof. Giorgio Terracina, Dr. Francesco Cauteruccio, Dr. Serena Nicolazzo and Dr. Alessia Amelio. I also had the opportunity to work with a wonderful team together with my colleagues (and friends) Dr. Luca Virgili, Dr. Gianluca Bonifazi and, in the last year, Dr. Michele Marchetti. During my experience as PhD student I also got to work with other colleagues met during off-site periods and conferences, which resulted in papers being published together.

The research described in this thesis starts from the observation that the sheer volume of data available today is staggering. We are constantly overwhelmed by more and more information. Consider, for instance, weather forecasts, recommended routes from navigation apps, news stories, data shared through social media (the average person has 8.4 social media accounts in 2020), the Internet of Things, and the list goes on. We are also generating new data all the time through devices like smart watches and fitness trackers. It can be very difficult to sift through all this information and find what is truly relevant. This is where models and approaches that can handle large amounts of data come in handy, allowing us to extract only the most important bits of information for a given domain. Ensuring the efficiency and effectiveness of extracting knowledge from all this data requires representing it in an adequate way.

In this thesis, we aim to provide a contribution in this setting and propose graph-based models that can uniformly represent and handle data coming from heterogeneous contexts. We believe that the connections between entities are important, and so we represent them through complex networks with nodes representing the domain entities and arcs modeling entity connections. These networks can also have

labels, weights, and a set of features attached to their arcs in order to store relevant information about interactions (e.g., number of common posts in a social network, amount of money exchanged between two wallets in a blockchain, number of transactions in an Internet of Things scenario, etc.). Once a complex network representing a scenario is built using our approach, any tools provided by Social Network Analysis can be applied to it, such as centrality measures, to derive the most important entities, or cliques, to determine the presence of strongly connected components, in order to gain insights into whatever scenario is being represented. In this way, our models offer great versatility since they can be used for any type of scenario with only minor adjustments.

The approaches, methods and results described are divided in two macro-categories that we call *Networking people* and *Networking things*. These two contexts focus on two different related entities: people and smart objects. In the former scenario we propose approaches aiming to find who are the most important users in an online social medium, to detect possible communities of information diffusers, to analyze specific behaviors of users and to classify them, and so on. In the latter one, we employ Social Network Analysis models to represent the interaction between objects, for example to detect anomalies in a Multi-IoT environment, to describe the behavior of smart objects in a workplace, to manage trust and reputation of smart objects, and so on.

There is a great deal of heterogeneity in the types of scenarios where Social Network Analysis can be used to represent and analyze. Each scenario presents its own unique challenges and issues. However, the concepts and approaches associated with Network Analysis in general are designed to deal with this heterogeneity. This means that a common methodology can be used to address different problems in different contexts.

I want to thank all the people who have helped me during my experience as PhD student. First, I want to express my gratitude to my supervisor Prof. Domenico Ursino, who helped me to conclude this PhD path in the best possible way. His advice will surely be helpful and useful for what is to come in the future.

I would also like to thank all the people I worked with. First of all my colleagues Luca Virgili, Gianluca Bonifazi, and Michele Marchetti, who have always allowed me to work and study with serenity and fun. My thoughts also go to Dr. Francesco Cauteruccio, who helped me understand many dynamics that were unknown to me, in order to make the most of my time as a PhD student. In addition, my experience as PhD student was surely enriched by all the colleagues I met during all the weeks I spent off-site.

Obviously, a special thanks goes to my parents who have always supported me during these three years. I would also like to thank my friends Emanuela, Francesco, Giovanni, Luca, and Matteo, who always brought me back to the real world every evening and reminded me that life is not just research and work.

A final thanks goes to myself. Although I have chosen one of the most difficult paths that my education could allow me, these three years of doctorate have allowed me to grow from many points of view. They allowed me to know realities that otherwise I would never have been able to know, together with a way of thinking that will surely be of great help to me in the future. I could not have achieved better human, social and educational growth in any other path.

November 2022 Enrico Corradini

Contents

Foreword . . . . . **I**

Preface . . . . . **III**

1 Introduction **1**

1.1 Motivations 1

1.1.1 Networking people 1

1.1.2 Networking things 11

1.2 General characteristics of the approach 17

1.2.1 Networking people 17

1.2.2 Networking things 26

1.3 Related works 30

1.3.1 Networking people 30

1.3.2 Networking things 40

1.4 Contributions 44

1.4.1 Networking people 44

1.4.2 Networking things 51

1.5 Outline of the thesis 54

Part I Networking people

2 Defining and detecting k-bridges **59**

2.1 Methods 59

2.1.1 A model for k-bridges and an approach to extract them 59

2.1.2 Investigating k-bridge properties 64

2.2 Results 76

2.2.1 Analysis of k-bridges and macro-categories in Yelp 76

2.2.2 Validation of k-bridge properties in other networks 84

2.2.3 Applications of k-bridges 88

3 Detecting user stereotypes and their assortativity 93
3.1 Methods 93
3.1.1 Dataset description 93
3.1.2 Stereotyping subreddits 100
3.1.3 Stereotyping authors 107
3.2 Results 110
3.2.1 Evaluating author assortativity 110
3.2.2 Correlation between subreddits and author stereotypes 115
3.2.3 Considerations about author stereotypes and assortativity . . . 118
3.2.4 Applications of stereotypes 120

4 Detecting backbones of information diffusers among different communities of a social platform 123
4.1 Methods 123
4.1.1 Network model 123
4.1.2 Detection of a backbone of information diffusers 124
4.2 Results 131
4.2.1 Dataset 131
4.2.2 Construction of the network $\mathcal{N}$ 135
4.2.3 Construction of $\hat{U}_Y$ 137
4.2.4 Construction of $\tilde{U}_Y$ and comparison with $\hat{U}_Y$ 139
4.2.5 Construction of the backbone of disseminator bridges 144

5 Investigating the NSFW phenomenon 147
5.1 Structural investigation 147
5.1.1 Methods 147
5.1.2 Results 157
5.2 Content investigation 171
5.2.1 Methods 171
5.2.2 Results 196

6 Investigating negative reviews and negative influencers 199
6.1 Methods 199
6.1.1 Definition of Yelp model 199
6.1.2 Definition of negative influencer stereotypes 200
6.1.3 Hypothesis definition 201
6.1.4 Preliminary analysis of negative influencers stereotypes 204
6.2 Results 207

6.2.1 Investigating the Hypothesis H1 . . . 207
6.2.2 Investigating the Hypothesis H2 . . . 209
6.2.3 Investigating the Hypothesis H3 . . . 212
6.2.4 Investigating the Hypothesis H4 . . . 215
6.2.5 Investigating the Hypothesis H5 and defining a profile of negative influencers in Yelp . . . 218
6.3 Discussion . . . 221
6.3.1 Reference context . . . 221
6.3.2 Main findings of the knowledge extraction process . . . 224
6.3.3 Theoretical contributions . . . 225
6.3.4 Practical implications . . . 227
6.3.5 Limitations and future research directions . . . 229

7 Investigating user behavior in a blockchain during a cryptocurrency speculative bubble . . . 233
7.1 Methods . . . 233
7.1.1 Dataset description . . . 233
7.1.2 Defining the user categories of interest . . . 234
7.1.3 Detecting the main features of the user categories of interest . . . 238
7.1.4 Generalizability of the proposed analyses . . . 241
7.2 Results . . . 244
7.2.1 Evaluating the existence of backbones linking users of a certain category . . . 245
7.2.2 Graphical backbone evaluations through k-trusses . . . 254
7.2.3 Defining the identikit of bubble speculators . . . 256
7.2.4 Predicting the characteristics of the main future actors . . . 257
7.2.5 Adoption of our approach in the next speculative bubble . . . 263

8 Representation, detection and usage of the content semantics of comments 267
8.1 Methods . . . 267
8.1.1 Comment filtering and text pattern extraction . . . 267
8.1.2 Content Semantics Network definition . . . 269
8.1.3 Evaluation of the semantic similarity of two CS-Nets . . . 271
8.1.4 Semantic similarity degree computation . . . 272
8.1.5 Dataset . . . 275
8.2 Results . . . 277
8.2.1 Analysis of generated Content Semantic Network . . . 277
8.2.2 Investigating β^x . . . 279
8.2.3 Investigating α . . . 281

8.2.4 Extracting knowledge from a real world scenario 283
8.3 Possible Applications 288
8.3.1 Content-based recommender systems 288
8.3.2 Collaborative filtering recommender systems 289
8.3.3 Building new user communities and/or identifying outliers . . 289
8.3.4 Building new subreddits and/or identifying outliers 290

9 Defining user spectra to classify user behaviors in cryptocurrencies . . . 291
9.1 Methods 291
9.1.1 Proposed method 291
9.1.2 Modeling a blockchain as a social network 292
9.1.3 Defining the spectrum of a user or a class of users 294
9.1.4 Defining the new version of the Eros Distance 295
9.1.5 Classifying users based on their spectra 299
9.1.6 Experiments 301
9.1.7 Dataset 301
9.1.8 An example of user spectrum 303
9.1.9 Defining the classes of interest 304
9.1.10 Defining class spectra 305
9.1.11 Weights of the Eros distance 312
9.2 Results 314
9.2.1 Evaluating our approach with the original Eros distance 315
9.2.2 Evaluating our approach with an exhaustive examination of all weight combinations for the Eros distance 317
9.2.3 Evaluating our approach with our version of the Eros distance 319
9.2.4 Computation time analysis 321
9.2.5 Discussion 322

10 Extracting information from posts on COVID-19 325
10.1 Methods 325
10.1.1 Approach to classify posts based on topics 325
10.1.2 Approach to build virtual subreddits with homogenous topics 330
10.1.3 Approach to build virtual communities of users with homogeneous interests 334
10.1.4 Dataset description 337
10.2 Results 339
10.2.1 Exploratory Data Analysis 339
10.2.2 Approach to classify posts based on topics 343
10.2.3 Approach to build virtual subreddits with homogeneous topics 347

10.2.4 Approach to build virtual communities of users with homogeneous interests . . . 351

11 Extracting time patterns from the lifespan of TikTok challenges . . . 357
11.1 Methods . . . 357
11.1.1 Dataset Description . . . 357
11.1.2 A Social Network-based model representing TikTok challenges 363
11.1.3 Analysis of the structure of the social networks associated with the challenges . . . 364
11.1.4 Definition of the lifespan intervals of a challenge . . . 367
11.2 Results . . . 375
11.2.1 Searching for time patterns in the challenge lifespans . . . 375

12 Investigating community evolutions in TikTok . . . 383
12.1 Methods . . . 383
12.1.1 Dataset construction . . . 383
12.1.2 Model definition . . . 384
12.2 Results . . . 386
12.2.1 A preliminary analysis of challenges . . . 386
12.2.2 Analysis of the evolution of user communities for non-dangerous and dangerous challenges . . . 389
12.2.3 Searching for evolutionary patterns in the challenge lifespans . 396

Part II Networking things

13 Networking wearable devices for fall detection in a workplace . . . 405
13.1 Framework description . . . 405
13.1.1 Personal Devices . . . 407
13.1.2 Area Devices . . . 408
13.1.3 Safety Coordination Platform . . . 409
13.2 Specialization of the proposed framework to fall detection . . . 411
13.2.1 Personal Devices for fall detection . . . 411
13.2.2 Area Devices for fall detection . . . 425
13.2.3 Safety Coordination Platform for fall detection . . . 426

14 Anomaly detection and classification in Multiple IoT scenarios . . . 429
14.1 Methods . . . 429
14.1.1 Extending the MIoT paradigm . . . 429

14.1.2 Modeling anomalies in a MIoT . . . 431
14.1.3 Investigating the origins and effects of anomalies in a MIoT . . 436
14.2 Results . . . 440
14.2.1 Testbed . . . 440
14.2.2 Analysis of the forward problem . . . 441
14.2.3 Analysis of the inverse problem . . . 446
14.3 Use case . . . 447

15 Increasing protection and autonomy of smart objects in the IoT . . . 451
15.1 Methods . . . 451
15.1.1 The reference IoT Model . . . 451
15.1.2 Technical description of our approach . . . 452
15.1.3 Security Model . . . 466
15.1.4 Experiments . . . 474
15.2 Results . . . 476
15.2.1 Comparison with other approaches . . . 481

16 Extending saliency maps and gaze prediction in an Industry 4.0 scenario 483
16.1 Methods . . . 483
16.1.1 Improving SalGAN to derive saliency maps for web pages . . . 483
16.1.2 Improving PathGAN to derive gaze path predictions for web pages . . . 488
16.2 Results . . . 495
16.2.1 Saliency map and gaze path prediction tool . . . 495
16.2.2 Dataset description . . . 495
16.2.3 Experiment Results . . . 499

Part III Closing remarks

17 Conclusions . . . 511
17.1 Networking people . . . 511
17.2 Networking things . . . 517

18 Future works . . . 519
18.1 Networking people . . . 519
18.2 Networking things . . . 524
18.3 Networking everything . . . 525

References . . . 527

List of Figures

2.1 Distribution of categories inside the macro-categories of Yelp 66
2.2 Distribution of user reviews in Yelp - Linear scale (on the left) and Logarithmic scale (on the right) 67
2.3 Distribution of the k-bridges against k in Yelp 67
2.4 Distribution of the neighbors of *bridges* in $\mathcal{U}^f$ 70
2.5 Distribution of the neighbors of *non-bridges* in $\mathcal{U}^f$ 71
2.6 Distribution of reviews for users in $\mathcal{U}^{cr}$ - Linear scale (on the left) and Logarithmic scale (on the right) 71
2.7 Distribution of the neighbors of *bridges* in $\mathcal{U}^{cr}$ 72
2.8 Distribution of the neighbors of *non-bridges* in $\mathcal{U}^{cr}$ 73
2.9 Distributions of the number of arcs for non-bridges, bridges, strong and very strong bridges 74
2.10 Distributions of (power) users against the strength of bridges 75
2.11 Distributions of k-bridges against their degree 76
2.12 Distribution of the reviews of Yelp users against the Yelp macro-categories 77
2.13 The network $\mathcal{M}^{1\%}$ 77
2.14 The network $\mathcal{M}^{5\%}$ 78
2.15 The network $\mathcal{M}^{10\%}$ 78
2.16 The network $\mathcal{M}^{15\%}$ 79
2.17 Variation of the density of the macro-category networks $\mathcal{M}^{X\%}$ against the increase of X 79
2.18 Variation of the average clustering coefficient of the macro-category networks $\mathcal{M}^{X\%}$ against the increase of X 80
2.19 Distribution of the k-bridges against k in Yelp after the removal of "Restaurants" 82
2.20 The networks $\mathcal{M}^{1\%}$ - $\mathcal{M}^{15\%}$ after the removal of "Restaurants" 83
2.21 Distribution of the k-bridges against k in Reddit 85

2.22 Distributions of the number of arcs for non-bridges, bridges, strong and very strong bridges in Reddit 86
2.23 Distribution of the k-bridges against k in the network of patent inventors 87
2.24 Distributions of the number of arcs for non-bridges, bridges, strong and very strong bridges in the network of patent inventors 88

3.1 Distribution of subreddits against posts (log-log scale) 94
3.2 Distribution of authors against posts (log-log scale) 95
3.3 Distribution of posts against scores (log-log scale) 95
3.4 Distribution of authors against negative posts (log-log scale) 96
3.5 Distribution of authors against positive posts (log-log scale) 96
3.6 Distribution of subreddits against comments (log-log scale) 98
3.7 Distribution of the average number of comments against the scores of the posts they refer to 98
3.8 Distribution of posts against comments (log-log scale) 99
3.9 Distribution of the average number of comments submitted to the subreddits receiving the 150 most commented posts 99
3.10 Distribution of authors against subreddits (log-log scale) 100
3.11 Distribution of the average number of comments received against the authors submitting the 150 most commented posts 100
3.12 Lifespan of the subreddits created in January 2019 101
3.13 Lifespan of the subreddits created in February 2019 (at left) and March 2019 (at right) 101
3.14 Lifespan of the subreddits born and died in February 2019 (at left) and March 2019 (at right) 102
3.15 Distribution of the subreddits of January 2019 died in the same day they were born against the number of their posts 102
3.16 Distribution of the subreddits of January 2019 died in the same day they were born against the number of their comments 103
3.17 Distribution of the subreddits of January 2019 died one day after they were born against the number of their posts 103
3.18 Distribution of the subreddits of January 2019 died one day after they were born against the number of their comments 104
3.19 Distribution of degree centrality for the nodes of $\mathcal{P}$ 111
3.20 (a) Number of authors of $\mathcal{I}_1$ connected to at least one author of $\mathcal{I}_k$ - (b) Percentage of authors of $\mathcal{I}_k$ connected to at least one author of $\mathcal{I}_1$ 111

3.21 (a) Number of authors of $\mathcal{I}_1$ connected to at least one author of $\mathcal{I}_k$ in the null model - (b) Percentage of authors of $\mathcal{I}_k$ connected to at least one author of $\mathcal{I}_1$ in the null model 112
3.22 (a) Number of authors of $\mathcal{I}_{20}$ connected to at least one author of $\mathcal{I}_k$ - (b) Percentage of authors of $\mathcal{I}_k$ connected to at least one author of $\mathcal{I}_{20}$ 113
3.23 (a) Number of authors of $\mathcal{I}_{20}$ connected to at least one author of $\mathcal{I}_k$ in the null model - (b) Percentage of authors of $\mathcal{I}_k$ connected to at least one author of $\mathcal{I}_{20}$ in the null model 113
3.24 (a) Number of authors of $\mathcal{I}_{39}$ connected to at least one author of $\mathcal{I}_k$ - (b) Percentage of authors of $\mathcal{I}_k$ connected to at least one author of $\mathcal{I}_{39}$ 114
3.25 (a) Number of authors of $\mathcal{I}_{39}$ connected to at least one author of $\mathcal{I}_k$ in the null model - (b) Percentage of authors of $\mathcal{I}_k$ connected to at least one author of $\mathcal{I}_{39}$ in the null model 114
3.26 (a) Number of authors of $\mathcal{I}_1$ connected to at least one author of $\mathcal{I}_k$ - (b) Percentage of authors of $\mathcal{I}_k$ connected to at least one author of $\mathcal{I}_1$ - (c) Number of authors of $\mathcal{I}_1$ connected to at least one author of $\mathcal{I}_k$ in the null model - (d) Percentage of authors of $\mathcal{I}_k$ connected to at least one author of $\mathcal{I}_1$ in the null model 115
3.27 (a) Number of authors of $\mathcal{I}_{20}$ connected to at least one author of $\mathcal{I}_k$ - (b) Percentage of authors of $\mathcal{I}_k$ connected to at least one author of $\mathcal{I}_{20}$ - (c) Number of authors of $\mathcal{I}_{20}$ connected to at least one author of $\mathcal{I}_k$ in the null model - (d) Percentage of authors of $\mathcal{I}_k$ connected to at least one author of $\mathcal{I}_{20}$ in the null model 116
3.28 (a) Number of authors of $\mathcal{I}_{39}$ connected to at least one author of $\mathcal{I}_k$ - (b) Percentage of authors of $\mathcal{I}_k$ connected to at least one author of $\mathcal{I}_{39}$ - (c) Number of authors of $\mathcal{I}_{39}$ connected to at least one author of $\mathcal{I}_k$ in the null model - (d) Percentage of authors of $\mathcal{I}_k$ connected to at least one author of $\mathcal{I}_{39}$ in the null model 116

4.1 Trends of the number of posts and comments over the time interval of our dataset 133
4.2 Trends of the number of posts and comments of r/Coronavirus over the time interval of our dataset 135
4.3 Trends of the number of posts and comments of r/Conspiracy over the time interval of our dataset 135
4.4 Distribution of the users of U against their degree centrality 137
4.5 Distribution of the users of U against their closeness centrality 138
4.6 Distribution of the users of U against their betweenness centrality 138

4.7 Distribution of the users of $D - C$ (on the left) and $C - D$ (on the right) against the posts published by them 141
4.8 Distribution of the users of $D - C$ (on the left) and $C - D$ (on the right) against the comments published by them 141
4.9 Distribution of the users of $D - C$ (on the left) and $C - D$ (on the right) against the number of communities in which they published posts 142
4.10 Distribution of the users of $D - C$ (on the left) and $C - D$ (on the right) against the number of communities in which they published comments . 142
4.11 Distribution of the users of $D - C$ (on the left) and $C - D$ (on the right) with regard the equidistribution of the posts published by them across communities 143
4.12 Distribution of the users of $D-C$ (on the left) and $C-D$ (on the right) with regard the equidistribution of the comments published by them across communities 143

5.1 Log-log plots of the distributions of subreddits against SFW posts (on top) and NSFW posts (on bottom) - Datasets regarding January and February 2019 150
5.2 Log-log plots of the distributions of authors against SFW posts (on top) and NSFW posts (on bottom) - Datasets regarding January and February 2019 152
5.3 Distributions of comments to the top 150 most commented SFW posts (on top) and NSFW posts (on bottom) against subreddits - Datasets regarding January and February 2019 155
5.4 Distribution of comments to SFW posts against scores - Datasets regarding January and February 2019 156
5.5 Distribution of comments to NSFW posts against scores - Datasets regarding January and February 2019 156
5.6 Distribution of the nodes of $\mathcal{P}$ against their degree centrality - linear scale (on top) and log-log scale (on bottom) 161
5.7 Distribution of the nodes of $\overline{\mathcal{P}}$ against degree centrality - linear scale (on top) and log-log scale (on bottom) 162
5.8 Top-ten authors who submitted more posts - authors of SFW posts at left and of NSFW posts at right 163
5.9 Top-ten authors who published on more subreddits - authors of SFW posts at left and of NSFW posts at right 163
5.10 Top-ten authors who received more comments - authors of SFW posts at left and of NSFW posts at right 163

5.11 Degree Assortativity of the authors of NSFW and SFW posts (high degree authors) . . . 165
5.12 Degree Assortativity of the authors of NSFW and SFW posts (medium degree authors) . . . 167
5.13 Degree Assortativity of the authors of SFW posts (low degree authors) . 168
5.14 Eigenvector Assortativity of the authors of NSFW and SFW posts (high degree authors) . . . 169
5.15 Distributions of posts against subreddits (at left, log-log scale) and comments against posts (at right, log-log scale) . . . 172
5.16 Distributions of scores against posts (at left, log-log scale) and comments (at right, log-log scale) . . . 173
5.17 The general workflow of our approach . . . 174
5.18 Number of extracted patterns against values of th_n . . . 181
5.19 Number of extracted patterns against values of th_c^+ . . . 182
5.20 Number of extracted patterns against values of th_c^- . . . 182
5.21 Number of extracted patterns against values of th_p^+ . . . 184
5.22 Number of extracted patterns against values of th_p^- . . . 184
5.23 Distribution of arcs against weights for $\mathcal{N}_{f_n}^{ui}$ . . . 186
5.24 Distribution of arcs against weights for $\mathcal{N}_{f_n}^{p}$ . . . 189
5.25 Distribution of cliques of *coexisting* patterns in our Pattern Networks . . 191
5.26 Distribution of arcs against weights for $\mathcal{N}_{f_n}^{uc}$ . . . 194

6.1 Distribution of the categories inside the Yelp macro-categories . . . 204
6.2 Average number of business reviews made by Yelp *users* for each macro-category . . . 205
6.3 Average number of business reviews made by Yelp *bridges* for each macro-category . . . 205
6.4 Distribution of access-dl-users against k . . . 206
6.5 Average number of stars for each macro-category of Yelp . . . 208
6.6 Distribution of score-dl-users against k . . . 209
6.7 Percentages of *users* such that they, and at least one of their friends, reviewed the same business negatively . . . 212
6.8 Percentages of *bridges* such that they, and at least one of their friends, reviewed the same business negatively . . . 213
6.9 Percentages of *non-bridges* such that they, and at least one of their friends, reviewed the same business negatively . . . 213
6.10 Percentages of *users* in the null model such that they, and at least one of their friends, reviewed the same business negatively . . . 214

6.11 Percentages of friends who, having reviewed the same business as a *user* who reviewed a business negatively, also provided a negative review . . . 214
6.12 Percentages of friends who, having reviewed the same business as a *bridge* who reviewed a business negatively, also provide a negative review 215
6.13 Percentages of friends who, having reviewed the same business as a *non-bridge* who reviewed a business negatively, also provide a negative review 215
6.14 Average percentages of *users* who, having made a negative review in a macro-category, have at least $X\%_{oo}$ of their friends who reviewed a business in the same macro-category negatively 216
6.15 Average percentages of *bridges* who, having made a negative review in a macro-category, have at least $X\%_{oo}$ of their friends who reviewed a business in the same macro-category negatively 216
6.16 Average percentages of *non-bridges* who, having made a negative review in a macro-category, have at least $X\%_{oo}$ of their friends who reviewed a business in the same macro-category negatively 217
6.17 Average percentages of *users* in the null model who, having made a negative review in a macro-category, have at least $X\%_{oo}$ of their friends who reviewed a business in the same macro-category negatively 217
6.18 Average percentages of *bridges* in the null model who, having made a negative review in a macro-category, have at least $X\%_{oo}$ of their friends who reviewed a business in the same macro-category negatively 218
6.19 Average percentages of *non-bridges* in the null model who, having made a negative review in a macro-category, have at least $X\%_{oo}$ of their friends who reviewed a business in the same macro-category negatively 218
6.20 Distribution of users of $\overline{U}$ against k 219
6.21 Distributions of the top $X\%$ of users with the highest degree centrality against k 220
6.22 Distributions of the top $X\%$ of users with the highest eigenvector centrality against k 220
6.23 Distributions of the top $X\%$ of users with the highest PageRank against k 221

7.1 Log-log plots of the distributions of transactions against `from_addresses` (at left) and `to_addresses` (at right) 234
7.2 Number of transactions over time 235
7.3 A graphical abstract representation of our algorithm 245
7.4 A 5-core of $\mathcal{N}^F_{Pre}$ 249
7.5 A 7-core of $\mathcal{N}^F_{Pre}$ 249
7.6 A 5-core of $\mathcal{N}^F_B$ 251

7.7 A 7-core of $\mathcal{N}_B^F$ . . . 252
7.8 A 5-core of $\mathcal{N}_{Post}^F$ . . . 254
7.9 A 7-core of $\mathcal{N}_{Post}^F$ . . . 254
7.10 Distribution of the addresses of $\mathcal{S}^F$ (at left) and $\mathcal{M}^F$ (at right) against the number of transactions of T_{Pre}^F . . . 258
7.11 Distribution of the addresses of $\mathcal{S}^F$ (at left) and $\mathcal{M}^F$ (at right) against the number of contacts of T_{Pre}^F . . . 259
7.12 Distribution of the addresses of $\mathcal{S}^T$ (at left) and $\mathcal{M}^T$ (at right) against the number of transactions of T_{Pre}^T . . . 260
7.13 Distribution of the addresses of $\mathcal{S}^T$ (at left) and $\mathcal{M}^T$ (at right) against the number of contacts of T_{Pre}^T . . . 260
7.14 Distribution of the addresses of $\mathcal{S}^F$ (at left) and $\mathcal{E}^F$ (at right) against the number of transactions of T_B^F . . . 261
7.15 Distribution of the addresses of $\mathcal{S}^F$ (at left) and $\mathcal{E}^F$ (at right) against the number of contacts of T_B^F . . . 261
7.16 Distribution of the addresses of $\mathcal{S}^T$ (at left) and $\mathcal{E}^T$ (at right) against the number of transactions of T_B^T . . . 262
7.17 Distribution of the addresses of $\mathcal{S}^T$ (at left) and $\mathcal{E}^T$ (at right) against the number of contacts of T_B^T . . . 263
7.18 Distribution of the Survivors (`from_addresses`) against the date of the last transaction . . . 264
7.19 Distribution of the Entrants (`from_addresses`) against the date of the last transaction . . . 264
7.20 Distribution of the Others (`from_addresses`) against the date of the last transaction . . . 265
7.21 Distribution of the Survivors (`to_addresses`) against the date of the last transaction . . . 265
7.22 Distribution of the Entrants (`to_addresses`) against the date of the last transaction . . . 266
7.23 Distribution of the Others (`to_addresses`) against the date of the last transaction . . . 266

8.1 Distributions of comments against posts . . . 277
8.2 Distributions of scores against comments . . . 277
8.3 Mean values of β^x against values of $\rho^x = |N_1^x| + |N_2^x|$ . . . 281
8.4 Mean values of α against values of $\phi = |N_1| + |N_2|$ . . . 282
8.5 Distribution of authors against comments on linear scale (top) and log-log scale (bottom) . . . 285

8.6 Hit ratio with different values of h and k. 287

9.1 Number of transactions over time 302
9.2 Distribution of Ethereum training addresses against the main Etherscan classes 304
9.3 Correlation matrix for the spectrum features of all the addresses in the training data set 306
9.4 Spectrum of the class "Token Contract" 308
9.5 Correlation matrix for the spectrum features of the class "Token Contract" 309
9.6 Spectrum of the class "Exchange" 310
9.7 Correlation matrix for the spectrum features of the class "Exchange" . . 310
9.8 Spectrum of the class "Bancor" 311
9.9 Correlation matrix for the spectrum features of the class "Bancor" 312
9.10 Spectrum of the class "Uniswap" 313
9.11 Correlation matrix for the spectrum features of the class "Uniswap" . . . 314
9.12 Distribution of Ethereum testing addresses against the main categories of Etherscan 316

10.1 A flowchart representing our approach to classify posts based on topics . 329
10.2 A flowchart representing our approach to build virtual subreddits with homogeneous topics 332
10.3 A flowchart representing our approach to build virtual communities of users with homogeneous interests 336
10.4 Distribution of the features `created` (normal scale), `subreddit` (normal scale), `num_comments` (log-log scale), `num_crossposts` (log-log scale), `score` (log-log scale) and `upvote_ratio` (semi-log scale) 340
10.5 Correlation matrix of the features of our dataset 341
10.6 Distribution of authors against posts (log-log scale) 342
10.7 Most frequent keywords in post titles and corresponding number of occurrences 343
10.8 Clustering of the keywords derived from post titles 344
10.9 The initial classification for the posts on COVID-19 in Reddit 344
10.10 The final classification for the posts on COVID-19 in Reddit 345
10.11 Distribution of posts against authors for S_1 (on top) and S_2 (on bottom) . 348
10.12 Distribution of posts against comments for S_1 (on top) and S_2 (on bottom) 349
10.13 Trend of the number of posts over time for S_1 (on top) and S_2 (on bottom) 349
10.14 Most frequent sets of at least 3 keywords occurring at least 3 times in the arcs of $\mathcal{S}_1''$ (on top) and $\mathcal{S}_2''$ (on bottom) and corresponding number of occurrences 353

10.15 Four communities of authors with homogeneous interests derived from $\mathcal{S}_1''$ 354
10.16 Four communities of authors with homogeneous interests derived from $\mathcal{S}_2''$ 354

11.1 Structure of non-dangerous networks 365
11.2 Structure of dangerous networks 365
11.3 Trend of the function $\nu_i(\cdot)$ and corresponding intervals for non-dangerous challenges 369
11.4 Trend of the function $\nu_i(\cdot)$ and corresponding intervals for dangerous challenges 370
11.5 Correlation matrix for the 26 features selected for characterizing lifespan intervals 372
11.6 The five clusters of intervals returned by Autoclass 374

12.1 Structure of non-dangerous networks 387
12.2 Structure of dangerous networks 387
12.3 Trends and intervals of $\nu_i(\cdot)$ for non-dangerous challenges 392
12.4 Trends and intervals of $\nu_i(\cdot)$ for dangerous challenges 393
12.5 Correlation matrix for the 20 features representing the behavior of the communities during a challenge 394
12.6 The four clusters of intervals returned by Expectation Maximization . . . 396
12.7 Example of the structure of a user community associated with a challenge for each cluster 397

13.1 The overall architecture of the proposed framework 406
13.2 An overview of Personal Devices available for a worker 407
13.3 Correlation matrix between the features 415
13.4 Activities labeled as *Not Fall* and *Fall* against the mean and the maximum accelerations on the Y axis 418
13.5 SensorTile.box (STEVAL-MKSBOX1V1) 420
13.6 Workflow of the Machine Learning Core of LSM6DSOX 421
13.7 Example of a workplace scenario (on the top) and description of how the verification of a fall and the transmission of the alarms occur (on the bottom) 424
13.8 Modules of the Safety Coordination Platform 427

14.1 Values of δ_{j_k} (corresponding to 0 hops) and average values of the anomaly degrees of all the nodes of $\mathcal{I}_k$ (on the left) and of the MIoT (on the right) being 1, 2 and 3 hops far from n_{j_k} in case of Presence-Hard-Contact anomalies 441

14.2 Values of δ_{j_k} (corresponding to 0 hops) and average values of the anomaly degrees of all the nodes of $\mathcal{I}_k$ (on the left) and of the MIoT (on the right) being 1, 2 and 3 hops far from n_{j_k} in case of Presence-Soft-Contact anomalies 442
14.3 Anomaly degrees and the corresponding standard deviations in different scenarios 443
14.4 Average number of nodes affected by anomalies against the number of IoT which an anomalous object participates to 444
14.5 Average percentage of anomalous nodes against their degree centrality . 445
14.6 Average percentage of anomalous nodes against their closeness centrality 445
14.7 Running time (in seconds) needed to compute δ_j in a MIoT against the number of its nodes 446
14.8 Percentage of times when our approach correctly detects the anomaly source (indicated by the label 0) or terminates in a node being 1, 2 or more than 2 hops far from it 447
14.9 Average running time (in seconds) of our approach for solving the inverse problem 448

15.1 General architecture of our approach 456
15.2 Computation of the trust of a trustor tr_i in a trustee te_j 460
15.3 Transaction aggregation and computation of the reputation of the smart objects of a community 462
15.4 Number of ordinary transactions performed in a month against community size 476
15.5 Average time necessary to execute all the ordinary transactions of a month against community size 476
15.6 Number of transactions in a month and time necessary to execute them against community size and probing probability - Part I 477
15.7 Number of transactions in a month and time necessary to execute them against community size and probing probability - Part II 478
15.8 Reputation decay for a malicious smart object inside a community of 100 components 479
15.9 Number of probing transactions and probing time with dummy actuators ($p = 0.1$) 480
15.10 Number of probing transactions and probing time with dummy actuators ($p = 0.5$) 480
15.11 Comparison of the average delay against the community size between our approach and the ones of [29] ad [470] 482

16.1 The architecture of SALGAN 484
16.2 SalGAN frozen layers during training 486
16.3 Qualitative comparison between the predictions of the original and fine-tuned SalGAN 487
16.4 The architecture of the original PathGAN 489
16.5 Two examples of mode collapse 489
16.6 Generator (blue) and discriminator (orange) loss after that the discriminator overfits the training set 490
16.7 Loss values of the generator (blue) and discriminator (orange) of WGAN 494
16.8 The architecture of WGAN 494
16.9 Ground truth (on the left) and WGAN prediction (on the right) of images that had led the original PathGAN to mode collapse 495
16.10 An example of a saliency map prediction returned by our tool 496
16.11 An example of a gaze path prediction returned by our tool 496
16.12 Comparison of the predictions returned by our fine-tuned variant of SalGAN and TSGAN on a layout rich of images 501
16.13 Comparison of the predictions returned by our fine-tuned variant of SalGAN and TSGAN on a layout dense of images and texts 502
16.14 Comparison of the predictions returned by NormalGAN and WGAN on one of the web pages of our dataset 505

List of Tables

2.1 The main notations used throughout this chapter 65
2.2 The top 20 pairs of macro-categories that appear simultaneously in one business of Yelp 66
2.3 Types of friends for bridges and non-bridges in $\mathcal{U}^f$ 68
2.4 Fractions of users with and without friends in $\mathcal{U}^f$ 69
2.5 Fraction of strong and very strong bridges present in the neighborhoods of bridges and non-bridges in $\mathcal{U}^f$ 69
2.6 Types of co-reviewers for bridges and non-bridges in $\mathcal{U}^{cr}$ 72
2.7 Fraction of strong and very strong bridges present in the neighborhoods of bridges and non-bridges in $\mathcal{U}^{cr}$ 73
2.8 Coefficients α and δ for the power law distributions of Figure 2.9 75
2.9 Values of the density and the average clustering coefficient for the networks $\mathcal{M}^{1\%}$ - $\mathcal{M}^{15\%}$ 79
2.10 Maximum and sub-maximum values of degree centrality and the corresponding macro-categories in the networks $\mathcal{M}^{1\%}$ - $\mathcal{M}^{15\%}$ 80
2.11 Maximum and sub-maximum values of closeness centrality and the corresponding macro-categories in the networks $\mathcal{M}^{1\%}$ - $\mathcal{M}^{15\%}$ 81
2.12 Maximum and sub-maximum values of betweenness centrality and the corresponding macro-categories in the networks $\mathcal{M}^{1\%}$ - $\mathcal{M}^{15\%}$ 81
2.13 Maximum and sub-maximum values of eigenvector centrality and the corresponding macro-categories in the networks $\mathcal{M}^{1\%}$ - $\mathcal{M}^{15\%}$ 82
2.14 Values of the density and the average clustering coefficient for the networks $\mathcal{M}^{1\%}$ - $\mathcal{M}^{15\%}$ after the removal of "Restaurants" 83
2.15 Maximum and sub-maximum values of the various centrality measures and the corresponding macro-categories in the networks $\mathcal{M}^{1\%}$ - $\mathcal{M}^{15\%}$ after the removal of "Restaurants" 84
2.16 Types of co-posters for bridges and non-bridges in $\mathcal{U}^{cp}$ 85
2.17 Types of co-inventors for bridges and non-bridges in $\mathcal{U}^{ci}$ 88

3.1 Parameters of the distributions of authors against negative posts 97
3.2 Parameters of the distributions of authors against positive posts 97
3.3 Classification of stereotypes concerning the subreddits “dead in crib” - Few posts case 105
3.4 Classification of stereotypes concerning the subreddits “dead in crib” - Many posts case 105
3.5 Classification of stereotypes concerning the subreddits “survivors” - Few posts case 106
3.6 Classification of stereotypes concerning the subreddits “survivors” - Many posts case 106
3.7 Classification of stereotypes concerning the subreddits “undelivered promises” - Few posts case 107
3.8 Classification of stereotypes concerning the subreddits “undelivered promises” - Many posts case 107
3.9 Classification of the stereotypes concerning “very positive” authors . . . 109
3.10 Classification of the stereotypes concerning “very negative” authors . . . 109
3.11 Classification of the stereotypes concerning “neutral” authors 110

4.1 List of the subreddits on COVID-19 composing our dataset 132
4.2 Features of a post in the dataset 133
4.3 Features of a comment in the dataset 133
4.4 Total number of posts and comments and average number of comments per post for each subreddit 134
4.5 Main characteristics of $\mathcal{G}_p$, $\mathcal{G}_c$ and $\mathcal{N}$ 136
4.6 Percentage of top users belonging to the intersection of some sets of interest 139
4.7 Percentage of top users belonging to the intersection of $Top(L^{DC}, Y)$ with the other sets of interest 140
4.8 Computation time of the average degree, closeness, betweenness, combined and disseminator centralities for the nodes of the network $\mathcal{N}$ associated with our dataset 144
4.9 Some characteristics of IN and its main connected components 145

5.1 Parameters about the authors and the subreddits of SFW and NSFW posts - $\mathcal{D}$ (resp., $\overline{\mathcal{D}}$) stores SFW (resp., NSFW) posts of January and February 2019, while $\mathcal{D}'$ (resp., $\overline{\mathcal{D}}'$) stores the same kind of post but for March and April 2019 149
5.2 Parameters of the distributions of subreddits against posts 150
5.3 Parameters of the distributions of authors against posts 151
5.4 Parameters of the distributions of posts against scores 153

5.5 Parameters of the distributions of subreddits against authors 153
5.6 Parameters of the distributions of comments against posts 154
5.7 Parameters of the distributions of subreddits against comments 155
5.8 Parameters of the distributions of comments to posts against scores . . . 157
5.9 Monthly trend of some parameters related to SFW posts 158
5.10 Monthly trend of some parameters related to NSFW posts 159
5.11 Basic parameters of the co-posting networks $\mathcal{P}$ and $\overline{\mathcal{P}}$ 160
5.12 Some parameters regarding authors in the dataset 172
5.13 Values of α and δ, minimum and maximum values of the distributions of interests for the dataset - *The values of α and δ for the left part of the distribution of scores against comments were computed considering the absolute values of scores 173
5.14 Some input texts from the dataset (swear words are partially masked) . . 175
5.15 Results obtained by applying the Data Cleaning and Annotation tasks on the texts of Table 5.14 (swear words are partially masked) 176
5.16 Example of the pattern extraction phase 179
5.17 Values of some basic parameters for User Interaction Networks 185
5.18 Values of the parameters α and δ for the power law distributions of arcs against weights in User Interaction Networks 186
5.19 Comparison between interacting users and the overall set of users in the User Interaction Networks 187
5.20 Fraction of interacting users who comment on each other's posts 187
5.21 Number of proactive users belonging to more networks 188
5.22 Values of some basic parameters for Pattern Networks 189
5.23 Values of some basic parameters for Pattern Networks 190
5.24 Cardinality of the sets of patterns exploited very often jointly by users . 190
5.25 Number of *coexisting* patterns simultaneously belonging to more Pattern Networks 192
5.26 Examples of *coexisting* patterns simultaneously belonging to $\mathcal{N}^p_{f_n}$, $\mathcal{N}^p_{f_c^+}$ and $\mathcal{N}^p_{f_p^-}$ 192
5.27 Values of some basic parameters for User Content Networks 193
5.28 Values of the parameters α and δ for the power law distributions of the arcs against the weights in User Content Networks 194
5.29 Comparison between common content users and the overall set of users in the User Content Networks 195
5.30 Number of common content proactive users belonging to more networks 195
5.31 Fraction of real opinion leaders who are also virtual, and vice versa . . . 196

6.1 Numbers and percentages of 2-bridges, access-dl-users and power users in Yelp . . . 206
6.2 Numbers and percentages of 3-bridges, access-dl-users and power users in Yelp . . . 207
6.3 Numbers and percentages of 4-bridges, access-dl-users and power users in Yelp . . . 207
6.4 Numbers and percentages of 5-bridges, access-dl-users and power users in Yelp . . . 207
6.5 Numbers and percentages of 6-bridges, access-dl-users and power users in Yelp . . . 207
6.6 Values of mean, standard deviation and mode of the number of stars assigned by bridges and non-bridges to all businesses . . . 208
6.7 Percentages of k-bridges and score-dl-users k-bridges who negatively reviewed the macro-category they mostly attended . . . 210
6.8 Comparison between the review score based on stars and the review polarity obtained by applying TextBlob . . . 211
6.9 Comparison between the review score based on stars and the review polarity obtained by applying Vader . . . 211
6.10 Characteristics of $\mathcal{U}$ and $\overline{\mathcal{U}}$ . . . 219

7.1 Some preliminary statistics performed on our dataset . . . 234
7.2 Values of the parameters of transaction distributions against addresses . 235
7.3 Percentage of the addresses and transactions covered by each set of power addresses . . . 236
7.4 Number of power addresses simultaneously belonging to the set of the top 1000 `from_addresses` and to the set of the top 1000 `to_addresses` in the three periods of interest . . . 237
7.5 Cardinalities of the possible intersubsections of the top 1000 addresses during the pre-bubble, bubble and post-bubble periods . . . 237
7.6 Number of power addresses belonging to the Survivors, Entrants and Missings categories . . . 238
7.7 Average number of nodes, average number of arcs and average density of the ego networks of the nodes belonging to the address categories of interest - Pre-bubble period . . . 240
7.8 Average number of nodes, average number of arcs and average density of the ego networks of the nodes belonging to the address categories of interest - Bubble period . . . 240

7.9 Average number of nodes, average number of arcs and average density of the ego networks of the nodes belonging to the address categories of interest - Post-bubble period 241
7.10 Analysis of the presence of backbones linking the Survivors during the pre-bubble period 246
7.11 Analysis of the presence of backbones linking the Missings during the pre-bubble period 247
7.12 Analysis of the presence of backbones linking the Entrants during the pre-bubble period 248
7.13 Analysis of the presence of backbones linking the Survivors during the bubble period 250
7.14 Analysis of the presence of backbones linking the Missings during the bubble period 250
7.15 Analysis of the presence of backbones linking the Entrants during the bubble period 250
7.16 Analysis of the presence of backbones linking the Survivors during the post-bubble period 251
7.17 Analysis of the presence of backbones linking the Missings during the post-bubble period 252
7.18 Analysis of the presence of backbones linking the Entrants during the post-bubble period 253
7.19 Average number of transactions, average number of contacts and average values of transactions for T^F_{Pre}, $\mathcal{S}^F$, $\mathcal{M}^F$ and $\mathcal{E}^F_{Pre}$ 258
7.20 Average number of transactions, average number of contacts and average value of transactions for T^T_{Pre}, $\mathcal{S}^T$, $\mathcal{M}^T$ and $\mathcal{E}^T_{Pre}$ 259
7.21 Average number of transactions, average number of contacts and average value of transactions for T^F_B, $\mathcal{S}^F$ and $\mathcal{E}^F$ 260
7.22 Average number of transactions, average number of contacts and average value of transactions for T^T_B, $\mathcal{S}^T$ and $\mathcal{E}^T$ 262

8.1 Some parameters regarding authors in the dataset 276
8.2 Values of α and δ, minimum and maximum values of the distributions of interests for the dataset - *The values of α and δ for the left part of the distribution of scores against comments were computed considering the absolute values of scores 276
8.3 Average number of arcs of the CS-Nets generated by applying our approach, with two different utility functions, and the random one 279

8.4 p-values obtained by performing the t-test between the outputs of our approach and those returned by the null model 279

9.1 Some preliminary statistics performed on our dataset 301
9.2 An example of a user spectrum 303
9.3 Number of addresses belonging to each class of interest for our investigation 305
9.4 Minimum, maximum, mean and standard deviation of the values of the spectrum features for the class “Token Contract” 307
9.5 Minimum, maximum, mean and standard deviation of the values of the spectrum features for the class “Exchange” 308
9.6 Minimum, maximum, mean and standard deviation of the values of the spectrum features for the class “Bancor” 309
9.7 Minimum, maximum, mean and standard deviation of the values of the spectrum features for the class “Uniswap” 312
9.8 Weights combination for the Eros distance relative to each class of interest 315
9.9 Number of addresses belonging to each class of interest 316
9.10 Confusion matrix of our classification algorithm with the classical version of the Eros distance 317
9.11 Values of some quality metrics obtained by applying our classification algorithm with the original Eros distance on testing data 317
9.12 The best weight combination for the Eros distance obtained after an exhaustive examination of all the possible combinations on testing data . . 318
9.13 Confusion matrix of our classification algorithm with an exhaustive examination of all the possible weight combinations for the Eros distance . 319
9.14 Values of some quality metrics obtained by applying our classification algorithm with an exhaustive examination of all the possible weight combinations for the Eros distance 319
9.15 The best weight combination for the Eros distance obtained by applying our heuristics on testing data 320
9.16 Confusion matrix of our classification algorithm with our version of the Eros distance 320
9.17 Values of some quality metrics obtained by applying our classification algorithm with our version of the Eros distance 321

10.1 Precision, Recall and F-Measure for the four samples under consideration 346
10.2 Main characteristics of the two samples S_1 and S_2 348
10.3 Some basic parameters of the networks $\mathcal{S}_1$ and $\mathcal{S}_2$ 348
10.4 The 10 keywords identified for S_1 and S_2 350

10.5 The virtual subreddits constructed for S_1 351
10.6 The virtual subreddits constructed for S_2 351
10.7 Some basic parameters of the networks $\mathcal{S}'_1, \mathcal{S}'_2, \mathcal{S}''_1$ and $\mathcal{S}''_2$ 352
10.8 Distribution of the arcs of $\mathcal{S}''_1$ and $\mathcal{S}''_2$ against the number of associated keywords 352
10.9 Some fundamental parameters of the sets of at least 3 keywords occurring at least 3 times in the arcs of $\mathcal{S}''_1$ and $\mathcal{S}''_2$ 354
10.10 Values of (average) density and (average) clustering coefficients for $\mathcal{S}''_1$ and $\mathcal{S}''_2$ and the networks associated with the communities obtained by applying our approach 355

11.1 Number of videos, date of the first and last one for each challenge 363
11.2 Basic structural characteristics of the networks associated with non-dangerous challenges 366
11.3 Basic structural characteristics of the networks associated with dangerous challenges 366
11.4 Differences between the main basic characteristics of the videos for non-dangerous and dangerous challenges 366
11.5 Differences between the main basic characteristics of the authors of videos for non-dangerous and dangerous challenges 367
11.6 Differences between the main basic characteristics of the lifespan for non-dangerous and dangerous challenges 368
11.7 Normalized MAE between the continuous function returned by the univariate spline interpolation and the real values for non-dangerous challenges (at left) and dangerous ones (at right) 368
11.8 Average values assumed in each cluster by the features representing lifespan intervals 375
11.9 Aggregate values of some fields that refer to non-dangerous and dangerous challenges 380

12.1 The seven non-dangerous challenges of our dataset 384
12.2 The seven dangerous challenges of our dataset 385
12.3 The record associated with each challenge video 386
12.4 Number of videos we collected for non-dangerous challenges (at left) and dangerous ones (at right) 386
12.5 Basic structural characteristics of non-dangerous networks 388
12.6 Basic structural characteristics of dangerous networks 388
12.7 Differences between the main basic characteristics of videos for non-dangerous and dangerous challenges 389

12.8 Differences between the main basic characteristics of the authors of videos for non-dangerous and dangerous challenges 389
12.9 Differences between the growth of user communities associated with non-dangerous and dangerous challenges 390
12.10 Normalized MAE between the continuous function returned by the univariate spline interpolation and the real values for non-dangerous challenges (at left) and dangerous ones (at right) 391
12.11 Sequences of intervals for non-dangerous and dangerous challenges . . . 398
12.12 Sequences of intervals for non-dangerous and dangerous challenges after the verification of the hypothesis that A and B are equivalent 399

13.1 Structure and some example tuples of the merged dataset 413
13.2 Feature definition 414
13.3 Feature relevance in identifying the correct class of activities 417
13.4 Accuracy, Sensitivity, Specificity values achieved by several classification algorithms when applied to our dataset (at left) and Worst Case Time Complexity of Training and Prediction (at right) 419
13.5 Adopted configuration of the MLC component 423
13.6 A taxonomy for *Not Fall Activities* (on the left) and *Fall Activities* (on the right) 425
13.7 Confusion matrix for the output provided by our device 425

14.1 Parameter values for our simulator 441

15.1 The main abbreviations used throughout this paper 453
15.2 An example of our dataset 475

16.1 Overview of the parameters of our PathGAN versions 493
16.2 Differences between the original PathGAN and our proposed variants (i.e., NormalGAN and WGAN) 494
16.3 Comparison of several characteristics of FiWI and our dataset 498
16.4 Values of the adopted evaluation metrics obtained for the original SalGAN, the three variants of this network proposed in this paper and TSGAN 499
16.5 Values of the adopted evaluation metrics obtained for our fine-tuned variant of SalGAN and some other saliency map prediction approaches proposed in the past literature 500
16.6 Performance of the original PathGAN, NormalGAN and WGAN when no threshold was set on the duration of fixations 503
16.7 Performance of the original PathGAN, NormalGAN and WGAN when a threshold equal to 0.0027 has been set on the duration of fixations 504

16.8 Comparison between One human baseline and WGAN 506

1

Introduction

This chapter highlights the motivations, the characteristics, the state-of-the-art and the contributions of this thesis, focusing on both networking people and things. In particular, the plan of the chapter is as follows: the first section points out the motivations that prompted us to define the approach presented. Then, the second section depicts the general characteristics of our approach. The third section presents the state-of-the-art behind this work, while the fourth one highlights the contributions of this thesis. Finally, in the fifth section, the outline of the thesis is presented.

1.1 Motivations

1.1.1 Networking people

The amount of data with which we are daily overwhelmed is increasing exponentially. Everything, from the weather forecast to the route our sat navs recommend to us, as well as news and social media (a person had an average of 8.4 social media accounts in 2020) rely on data. We are also constantly harvesting new data in order to optimize any activity we undertake, whether that is through smart watches, fitness bands or smart homes.

In the first part of this thesis, "Networking people", we address this issue in the social network domain and, specifically, look at three well-known platforms, i.e., Yelp, Reddit, and TikTok. Yelp is a particularly interesting platform to study, as it is a business directory service and a crowd-sourced platform designed to help users find businesses, like restaurants, hotels, pet stores, spas, and many more. It is one of the most widely used review platforms on the Web. It ranks 10^{th} on the SimilarWeb list of the top 100 leading websites by traffic[1], with approximately 800 million visits per month. Reddit is a social media platform that provides a space for users to share links, text posts, and engage in discussions with a community of like-minded

[1] `https://www.similarweb.com/top-websites/`

individuals. It allows users to find and join communities, known as "subreddits", dedicated to specific topics, such as news, technology, politics, and many more. With approximately 1.2 billion monthly active users, Reddit ranks as the 19^{th} most visited website in the world, according to SimilarWeb. TikTok is a social media platform that is centered around short-form videos, with a focus on music, dance, and comedy. It is a platform that allows users to create, share, and discover 15-second videos, often set to music and various sound effects. TikTok has become extremely popular, especially among younger generations, with approximately 689 million monthly active users. According to SimilarWeb, it ranks as the 40^{th} most visited website globally.

In all these cases, and generally speaking for all social networks, the best way to model them is through the construction and analysis of complex networks. For example, by using complex network analysis techniques, we were able to investigate the interactions and connections between users on Yelp. We were also able to identify friendship and review relationships between users, as well as to study negative user behavior and to introduce k-bridges, which are people interested in different business categories who can strongly influence others. Also, by creating a co-posting network, representing the interactions between users and their post publishing activity in Reddit, we were able to gain insight into how users tend to be connected with other ones having similar characteristics. This is known as the homophily property, and it is one of the key properties that researchers analyze in social networks. In TikTok, we were able to identify the lifespan of dangerous and non dangerous challenges. We used Social Network Analysis techniques to build the networks of the evolution of these challenges, in order to identify patterns that characterize and differentiate dangerous and non dangerous challenges.

In this first part of the thesis, we address these issues to better understand and analyze the behavior of users on social networks, by studying the interactions and connections between users, and the spread of information on these platforms. By using complex network analysis techniques, we were able to gain valuable insights on the three social networks mentioned above that can be employed to improve the user experience, inform business decisions, and even influence the future of technology.

Our analyses on networking people are not only focused on online social media platforms, but blockchains as well.

In the next subsections, we take a more detailed look at the motivations for undertaking the researches illustrated in this thesis in each of the cases mentioned so far.

Defining and detecting k-bridges. Bridges, i.e., entities connecting different subnetworks of the same network or different networks of a multi-network scenario,

attracted the interest of many researchers in several disciplines, ranging from sociology to telecommunication networks and transports. They also attracted the interests of researchers studying Online Social Networks, who considered them as users linking sub-networks of a single network [249, 567, 388, 71, 73, 648] or linking different networks in a multi-network context [103, 110, 106, 478].

In the past, all researchers focused on the bridge capability of connecting two communities. However, with the proliferation of social media, bridges currently tend to connect a higher number of sub-networks in a network or a higher number of networks in a multi-network scenario.

Their behavior and properties could vary against the number k of communities they connect. As a consequence, it appears interesting to introduce a new notion, called *k-bridge*. A k-bridge is a user who connects k sub-networks of a network or k networks of a multi-network scenario. k-bridges are particular users capable of playing an important role in opinion transmission, user influence, etc. Indeed, they allow a person or a business in a community to be known in another one. This may have important applications in the dissemination of information, in the search for influencers, and in marketing, for example when a business, leader in one category, wants to expand in another related category.

Our research on bridges was carried out on Yelp's dataset. Yelp (`https://www.yelp.com`) is a platform that helps people find local businesses, like dentists, restaurants, hair stylists, and many more. The motivations underlying the choice to adopt Yelp as a main study platform are related to its pure crowd-sourced nature. As a matter of fact, researchers have found in Yelp one of the main resources for studying user behavior in open-review platforms. Therefore, many works on Yelp have been focused on review and rate analysis, sentiment analysis, fake review and fake rate discovery, as well as recommendation analysis [114, 614, 455, 412, 631].

Detecting user stereotypes and their assortativity. The term "stereotype" comes from the combination of two Greek words, namely "stereos" (i.e., solid) and "typos" (i.e., impression). It is adopted to indicate a popular belief about specific groups of individuals. This term first appeared in the press at the end of the 18^{th} century. Later, it was introduced into modern psychology at the beginning of the 20^{th} century by Walter Lippman [404]. The tendency to classify people into groups and to associate each group with a "general idea", a "label" (and, ultimately, a stereotype) is intrinsic to the human mind. As a result, many (both positive and negative) stereotypes have been defined in the history of humanity, in the most disparate areas. Think, for instance, of the stereotypes coined in sport, art, literature, and so on. With the capillary spread of the Web, the practice of coining and using stereotypes has extended

from real life to cyberspace [239, 369]. As the Web became increasingly interactive, with the transition to the Web 2.0 and, above all, with the appearance of social networks, the adoption of stereotypes in the cyberspace become more and more evident [670, 497, 189, 593, 515, 108]. For example, in Facebook, one can encounter stereotypes like "Lime-Lighters", "Emo's", "Philosophy Majors", "Hopeless Romantics", "Ghosts", "Stalkers", "Addicts", and so forth [2]. Similarly, Instagram also presents a wide range of stereotypes [1]. Stereotypes do not necessarily have a negative meaning, as it often happens in real life. On the contrary, they can be extremely useful in everyday communications and interactions in social networks. Going one step further, it could be possible to define "scientific" stereotypes that could be used in scientific applications. In this, Reddit fits well and, in this context, besides defining stereotypes for the authors of Reddit, it is possible to also introduce stereotypes for subreddits.

On the other hand, the concept of "assortativity" or "assortative mixing" in a social network was introduced in a famous paper of Newman [462]. It is strictly related to the concept of homophily [435] and indicates a network node's predilection to relate to other nodes that are somewhat similar. Several possible similarities could be considered in assortativity, but the most investigated one is node degree. Newman focused on degree assortativity and defined a network as assortative if its nodes having many connections tend to be connected to other nodes with many connections. He showed that social networks are often assortatively mixed, whereas technological and biological networks tend to be disassortative. After Newman, some authors investigated assortativity in several social networks, such as Facebook [109], Twitter [107], Cyworld, Orkut and MySpace [17]. Assortativity in Reddit was only marginally considered in the past studies. Different kinds of assortativity can be analyzed, such as degree assortativity, the most studied one in the past, and eigenvector assortativity.

Detecting backbones of information diffusers among different communities of a social platform. Information diffusion has always been one of the core topics of social network research [391, 407, 236, 544, 669]. This topic is, at the same time, classic and current. In fact, information diffusers are always looking for new techniques to disseminate information of their interest. Unfortunately, these techniques are not exploited only to spread true or useful information. Indeed, they are often adopted by organized groups to spread fake news, political propaganda, etc. [165, 122]. Therefore, identifying the possible existence of backbones of information diffusers, besides being a challenging issue for social network researchers, can become essential to fight negative phenomena, like those mentioned above [141, 282].

A specific, but increasingly common and intriguing, scenario concerns information dissemination among different communities of the same social platform. Consider, for example, the diffusion of information among different subreddits in Reddit or among different groups in Facebook.

In this scenario, it is interesting to verify the possible existence of backbones of users operating in different communities (possibly with different roles in them), who support each other in spreading information of their interest [112, 479].

The aim is to identify the so-called disseminator bridges, i.e., users particularly active and organized in disseminating information on several communities. To identify such users, the concept of centrality is exploited. It has always been considered a key concept in Social Network Analysis [613].

Investigating NSFW contents and their authors. The term "NSFW" (Not Safe For Work) was proposed in 1998 to denote user-submitted content not suitable to be viewed in public or professional contexts. Since its first appearance, many social media have adopted it to indicate certain contents present in them. For this reason, some authors have been interested in studying this phenomenon in several social media. For instance, the authors of [609] investigated the role of images and selfies in NSFW content in `tumblr.com`, while the authors of [181] analyzed the level of anonymity of NSFW content in Twitter and Whisper.

Reddit is one of the social media that has adopted the concept of NSFW in an explicit and well-structured way. In fact, it allows users to explicitly tag certain posts and comments as NSFW. Despite this important role assigned by Reddit to the NSFW concept, only a few researchers have investigated the phenomenon of NSFW content in this social platform. Specifically, the authors of [433] studied the behavior of NSFW moderators, compared to SFW ones, during a specific Reddit moderator protest. Instead, the authors of [457] focused on the protection of NSFW content for minors accessing Reddit. Finally, the authors of [180] proposed a Social Network-based approach for studying NSFW posts on Reddit. However, they considered only the structural features of the posts, without taking their content into account. In addition, several authors have focused on the analysis of this phenomenon in other social networks. The study about the role of images and selfies in NSFW content of `tumblr.com`, presented in [609], and the analysis of the anonymity level of NSFW content in both Twitter and Whisper, described in [181], are two examples.

NSFW content can be analyzed through Social Network Analysis, in order to analyze NSFW posts and comments on Reddit. The study can be focus both on the structure of the posts and comments and on their content. First of all, a possible goal is analyzing how these contents differ from the SFW ones in Reddit. From this

knowledge it is also possible to understand how the authors of NSFW contents are different from the ones of SFW contents. Then, it is possible to outline a study based on the extraction and analysis of text patterns present in NSFW adult content on Reddit.

Investigating negative reviews and negative influencers. A phenomenon that represents a hot topic for review platforms is the analysis of negative reviews [70]. This is extremely important not only for the consequences it has in practice, but also from a more theoretical point of view. In fact, it is well known that the Likert scale, which reviews and the corresponding scores are based on, is positively biased [26, 496, 76]. As a consequence, the presence of negative reviews is a really important problem indicator for a business and, consequently, a valuable piece of information [367, 392]. Indeed, negative reviews can provide much more information, knowledge and improvement possibilities than positive ones [139]. For this reason, many researchers have already investigated the role of ratings and reviews on businesses, along with their social implications [610, 411].

In this scenario, a review platform like Yelp is particularly suitable for analysis by interested researchers. Despite the numerous studies on Yelp that have been presented in the past literature, to the best of our knowledge, no paper has proposed a multi-dimensional model capable of best capturing the specificity of Yelp to be at the same time a review platform, a social network and a business directory. Moreover, no paper has proposed a study focused entirely on negative reviews on Yelp that, starting from a representative model of them, could define several stereotypes of users and, hence, build the profile of negative influencers.

The practical implications of negative reviews and influencers have a large variety of real-world applications. First of all, it was proved that negative reviews have a stronger effect on businesses than positive ones [12]. Furthermore, influencers play a crucial role for the successful placement of products in a social network. So, it is important to know who are the negative influencers that could damage a business, in order to strive to turn them into neutral, or even positive, influencers [660, 673]. Finally, gaining trust through online reviews can help a business gather venture capitals for its growth [243, 367]. As a matter of fact, reviews are consumer opinions, unfiltered by traditional media, more sincere and imperfect [12, 162]. For this reason, a proper coverage of positive reviews can attract more financiers [12, 163, 358]. On the other hand, negative reviews and influencers can drive potential investors away from investing in a company [414].

Investigating user behavior in a blockchain during a cryptocurrency speculative bubble. Cryptocurrencies were the subject of a speculative bubble, similar to the tulipans' and stock market ones [632]. Indeed, the popularity of blockchains has been growing continuously from 2008, and the interest on cryptocurrencies followed the same growth. For instance, the price of Bitcoin surged almost 2,800% in four years and has fallen by 80% in just few weeks, between the end of 2017 and the beginning of 2018, leading to a huge gain for a few people and a big loss for the majority of the investors. These events are interesting to investigate from a data science perspective, because they allow the extraction of knowledge patterns to prevent other similar cases. As a matter of fact, several studies investigate the whole speculative cryptocurrency bubble and its consequences for economy and technology [658, 270].

However, a very limited number of studies take the intrinsic nature of blockchain as a social network into account. Actually, the relationships between blockchain users are extremely relevant in the extraction of unknown patterns and in the disclosure of new viewpoints for analyzing this speculative bubble. For this reason, Social Network Analysis notions [231, 359] can provide a big help to study the relationships in the blockchain network. In this activity, it is reasonable to think of a social network in which each node indicates a user, represented through her/his blockchain address, whereas each arc denotes a transaction between two users. This social network, and the investigation perspective it makes possible, can be extremely useful to support the extraction of knowledge on the speculative bubble of the years 2017 and 2018. The Ethereum blockchain is extremely useful to examine the behavior of its users [127] in these two years, which include the pre-bubble, bubble and post-bubble phases.

Representation, detection and usage of the content semantics of comments. In recent years, content analysis of people's comments on social media has received an increasing boost [44, 68, 561, 117, 118]. In fact, comments on social media represent one of the places where a person expresses her opinion on certain topics most spontaneously [86, 216, 618, 616]. As a consequence, they are an extremely powerful tool to know the true feelings and thoughts of a person and, ultimately, to reconstruct her profile [210, 445, 571]. It is important to deal with comments written by people to whom correspond well-defined accounts. Anonymous comments can be dangerous to analyze because they are both less reliable and they would be useless for this kind of research. While spontaneity is the main strength of comments, it can also become their main weakness. Indeed, just because comments are written on the spot, their content is often unstructured, sometimes apparently confused, other times ap-

parently contradictory. Nevertheless, there is no doubt that an in-depth analysis of a large set of comments, written for example by a single user, could allow the extraction of a "fil rouge", a common thread representing a thought, a content profile beyond the apparent inconsistencies of single comments. However, identifying this "fil rouge" requires a very thorough and holistic analysis of the content semantics.

Defining user spectra to classify user behaviors in cryptocurrencies. In recent years, we have witnessed an impressive development of the blockchain technology [684]. Smart contracts were introduced in Ethereum and this technology has spread to a variety of applications.

Anyone can participate to a blockchain network; therefore, different actors can be identified in this ecosystem [88]. For example, if we consider a blockchain like Ethereum, some actors (called miners) verify transactions, while others allow wallets to trade different cryptocurrencies and/or make banking transactions. Some others deal with auctions, others offer games or services, and so on. In some cases, there are online systems that provide a classification of the wallets of a blockchain network, even if the fraction of classified wallets is very small. The most known of such systems is Etherscan[2], which provides this service for Ethereum. Etherscan keeps tracks of wallets and transactions happening in Ethereum. In addition, it also provides a categorization of addresses in this blockchain[3].

Knowing a wallet's category can be extremely relevant in the context of blockchain networks [666, 612]. For example, such a knowledge allows us to find a set of competitors of a wallet performing a certain activity (exchange, bancor, etc.). In addition, through appropriate analyses, it is possible to identify whether, within a category, there are backbones of wallets connected to each other to avoid competing with one another or to gain dominant positions over others. Again, thanks to even more complex analysis, it is possible to understand the different strategies carried out by actors of the same category and which of them is the winning one.

Despite the importance of this knowledge, in the past literature, there exist very few approaches that, given a user of a blockchain network, can automatically derive her category [328, 612, 402, 691, 312, 606]. Furthermore, the few categorization approaches currently existing are usually tailored to the Bitcoin blockchain, while general ones have been tested on small specific blockchains. As for Ethereum, several approaches to identify actors belonging to a certain category of interest have been proposed in the past. Instead, to the best of our knowledge, no tailored classification approaches, like the ones presented for Bitcoin, have been proposed for Ethereum.

[2] https://etherscan.io/

[3] https://etherscan.io/labelcloud

As a consequence, the only current way to classify wallets in this blockchain is based on the activity of providers of this service, like Etherscan. However, they can classify at most those addresses reported by the users of the service. Unfortunately, such addresses are only a small minority of those present in Ethereum.

Extracting information from posts on COVID-19. COVID-19 is a severe disease that is upsetting the world. It has affected nearly every aspect of human life, from healthcare to economy, from education to tourism, and so on. That is why it has provoked, and continues to provoke, an enormous debate among experts and ordinary people. In this context, it is inevitable that COVID-19 is also one of the most user-focused topics in social networks. This fact has aroused the interest of Social Network Analysts, who have already proposed several studies on how COVID-19 has been treated in the main social networks (see, for example, the studies reported in [146, 578, 215, 172, 63, 453], just to mention a few).

The variety of issues related to COVID-19, along with the variety of social networks and, more generally, social media and journals discussing them, opens up interesting challenges. In fact, it is worth observing how the same issue arouses debates very heterogeneous in content and modalities, depending on the medium in which they take place and the people participating in them. On one hand, we have the major generalist networks, such as Facebook and Twitter, which are very widespread. Because of the intrinsic characteristics of these networks, users are led to write their posts very frequently and "on the fly". Therefore, these networks have the merit of immediately revealing the feelings of their users about the issue they are discussing. However, such feelings could be very fickle, as a user often writes on these networks without careful meditation [325]. As a consequence, it may happen that she takes completely different positions on the same issue during the same day as she reflects better on the subject she is debating. Since these networks are generalist, both common users and specialists in various fields (e.g., virologists, epidemiologists, economists, politicians, etc.) write on them. On the other hand, we have scientific networks and social media. In this case, users are specialists in their fields. Therefore, they are physicians in medical social media and journals, economists in business social media and journals, and so on. Content written in this context is very thoughtful and, in the case of research journals, is also peer-reviewed. Between these two extremes there are several intermediate cases. A very interesting one is the case of generalist social networks that are very popular but not as widespread as Facebook and Twitter. In them, writers do not usually publish their content "out of the blue", as people do on Facebook or Twitter, but periodically, for example at the end of a day [446]. As a result, what is written in these social networks is more medi-

tated than the content published in Facebook or Twitter. However, differently from specialized media, anyone (and not only specialists) can publish on them.

This last category of networks surely deserves great attention because of the intermediate nature between the two extremes highlighted above and because of their considerable diffusion. In fact, it could be possible to extract from the networks belonging to it information different from both the one retrievable from Facebook and Twitter and the one retrievable from specialized social media. One of these networks is Reddit.

Extracting time patterns from the lifespans on TikTok challenges. TikTok, also known as Douyin in China, is a social network that allows its users to make funny and creative videos of short duration, typically 15 to 60 seconds. It has quickly become the social network of choice for several categories of users, especially for the so-called Generation Z [577]. There are several features characterizing TikTok with respect to other social networks. They include: *(i)* the HD resolution and full screen display; *(ii)* the presence of advanced video editing features; *(iii)* the possibility of adding a music clip to a posted video; *(iv)* FYP and the associated recommendation algorithm; *(v)* a much higher prevalence of challenge-related posts than in the other social networks. In particular, the last two features are very characteristic for TikTok.

Actually, despite its young age, TikTok has already been the subject of many studies in the past literature. It has certainly attracted a lot of marketing researchers [159, 635, 292], who studied the characteristics of TikTok influencers [343, 622], as well as the role and the weight they play in increasing sales. Other researchers have studied the role of TikTok in politics [553, 413, 582], health [688, 396, 151], and so forth. The privacy and security issues arising with the use of this social medium [466, 349, 438, 678], the content spread in it [55, 637], the dynamics underlying it [652], and its recommendation algorithm [194, 647, 575, 681, 357, 55] have also been highly investigated. Instead, although challenges are one of the most important aspects of TikTok, only very few authors have still analyzed them [692, 356, 155, 592]. Furthermore, these authors investigated aspects very different from the challenge lifespan and the ability to distinguish non-dangerous challenges from dangerous ones.

A challenge is a viral showdown/competition. It is identified by a hashtag and starts with a user who posts a video with that hashtag and invites other ones to replicate the same video in their own way. Most challenges are fun and harmless; however, there are also other ones related to harmful or dangerous behaviors [381, 250, 641, 485, 403, 355, 355, 387, 281, 689, 492, 525]. TikTok removes challenges reported as dangerous and has increased safety controls . However, considering the

huge number of users and challenges created every day on this social platform, as well as the usage of some tricks exploited by the authors of dangerous challenges to bypass controls, the risk that there are unlocked dangerous challenges is real.

Investigating community evolutions in TikTok. Another interesting research issue regarding this social medium concerns the study of the communities participating in a challenge and their evolution over time. As pointed before, the study of the lifespan of challenges can bring to a better understanding of the dynamics of this social medium and a better identification of dangerous behaviors. In addition we can investigate the communities of users participating to such challenges. The possible research questions are several. For example, it could be interesting to study the differences in the evolution and dynamics of communities related to dangerous and non-dangerous challenges. After this, it is possible to investigate the different connection levels of users. Employing Social Network Analysis, it is possible to study how the relationships between users evolve even in this Online Social Network.

As we said above, TikTok has been studied in the past in different contexts [343, 622, 159, 635, 292, 559, 553, 413, 582, 688, 396, 151, 321]. However, to the best of our knowledge, no paper specifically investigated the differences in the evolutionary dynamics of communities in dangerous and non-dangerous challenges, as well as the possibility of using these differences to search for evolutionary patterns capable of distinguishing one kind of challenge from the other.

1.1.2 Networking things

The increasing pervasiveness of the Internet of Things (IoT) in everyday life is made possible by new research approaches that enable objects to be smart, autonomous and reliable. However, as the IoT grows, various challenges arise. The IoT is characterized by a large number of (often smart) objects, with limited storage and computing capability, great dynamism (due to the high number of nodes joining or leaving the IoT at any time), and criticality and sensitiveness of used services and applications.

In the second part of the thesis, "Networking things", we apply Social Network Analysis to context different from the interactions of real people, shifting to the analysis of smart objects.

Being a set of smart devices connected to each other, IoT can be studied through Social Network Analysis. Sensors and actuators, embedded in objects, are the basis for the Internet of Things. These smart devices can interact with each other and with humans, making our lives easier and more efficient. For example, they can help to monitor the workplace for safety hazards, reducing the risk of accidents.

In the past, anomalies have been investigated in social networks for a variety of reasons, including detecting fraudulent individuals, spammers, and malicious behaviors. More recently, anomaly detection has been analyzed in contexts where more than one social network interacts with each other. It is interesting to verify the possible expansion of Social Network Analysis based approaches to IoT.

In addition, the protection of objects is a crucial issue that needs to be addressed. This is especially important if we want to guarantee them a great autonomy.

Social Network Analysis (SNA) can be also employed in the context of Industry 4.0 to help identify and better understand relationships between people, organizations, and other entities. With so much information available, it is crucial for companies to capture the attention of users when they see digital images concerning them and their products. An effective digital image design that conveys the desired message can lead to increased popularity and potential returns for a company.

In the next, we provide more detailed overview to the motivations for undertaking the researches described in this thesis in each of the fields mentioned above.

Networking wearable devices for fall detection in a workplace. During the last decades, we are experiencing a continuous increase of the attention to the safety and health of workers in their daily activities. This can be explained by the fact that statistics on injuries and deaths at work are far from reassuring. For example, Eurostat statistics reported 3,344,474 injuries and 3,552 deaths at work during 2017[4]. In this worrying scenario, many efforts to develop solutions and devices to protect workers during their activities have been made. Indeed, there are many objectives to achieve, such as predicting what is going to happen, warning in case of emergencies to respond promptly, controlling access to special areas, and so on. In such a scenario, Information and Communication Technology (ICT) has provided, and continues to provide, a great contribution. The Internet of Things (IoT) [535] and Machine Learning [510] are certainly two of the ICT sectors that are playing an increasing key role.

Smart objects, which are the protagonists of the IoT, extend the benefits of the Internet from humans to things, allowing them to interact with each other in ever smarter ways [22, 541]. They play a key role in increasing safety at work [454]. In fact, they can help to continuously and thoroughly monitor the situation in the workplace, immediately issuing alarms, which could lead to a reduction of accidents (think, for example, of gas leaks). Furthermore, smart objects can be included in the

[4] https://ec.europa.eu/eurostat/statistics-explained/index.php/Accidents_at_work_statistics

equipment of workers to monitor particular events and, in case of an accident, send the alarm and speed up rescue operations.

The fact that these objects are becoming increasingly smart and capable of making decisions and communicate with each other in increasingly complex ways has led researchers to propose sophisticated architectures capable of exploiting all this potential. For example, two architectures that take into account social relationships between objects are the Social Internet of Things (SIoT) [45] and the Multiple Internet of Things (MIoT) [53, 617, 471]. In both these cases, objects are becoming more and more social, and their behavior is becoming increasingly similar to human behavior. Another highly evolved IoT architecture is the Sentient Multimedia System [115, 8]. It is a distributed system capable of actively interacting with the environment by gathering, processing, interpreting, storing and retrieving multimedia information originated from sensors, robots, actuators and other information sources. As with SIoT and MIoT, end users are involved in the whole process, since they are called to communicate and express their feelings, evolving needs and requests to the devices.

The development of such complex IoT architectures has led to an enormous growth of data that must be processed in a very short time in order to obtain useful information. The most advanced solution to this problem is Machine Learning, which, not surprisingly, has had a spectacular growth in recent years [82, 504]. Machine Learning provides supervised and unsupervised learning algorithms that aim at extracting knowledge patterns from data. The knowledge extracted can be descriptive, diagnostic, but, above all, predictive and prescriptive [213]. For example, a Machine Learning based approach could analyze data from sensors installed in a working environment (e.g., data about a gas leak in a certain area) to predict a possible imminent accident (e.g., a fire) and to prescribe certain actions (e.g., alerting all personnel in that area to proceed with an immediate evacuation).

Anomaly detection and classification in Multiple IoT scenarios. In the Concise Oxford Dictionary[5], *anomaly* is defined as "*something that deviates from what is standard, normal, or expected*". If regularities allow investigating the general characteristics of a complex system, anomalies allow the uncover and analysis of unexpected features that might not be otherwise discovered. For this reason, the detection of anomalies has become very important in data analytics, and is widely investigated both in statistics and Machine Learning [14, 13, 16]. The relevance of anomaly detection is universally acknowledged, since data anomalies are at the basis of significant events and patterns. Example application domains include: privacy and cybersecu-

[5] Concise Oxford Dictionary - https://en.oxforddictionaries.com

rity [665, 633], fault detection [315], ecological disturbances [145], communication networks [629], social media life [147, 556, 595, 663], and gene regulation [351, 350].

In recent years, anomalies have been widely investigated in social networks to detect fraudulent individuals [545, 19], spammers [568, 235], malicious behavior, and so forth. Even more recently, anomaly detection has been analyzed in contexts where more social networks interact with each other [103], thus going from social networking into social internetworking.

Social internetworking is certainly one of the frontiers of Social Network Analysis, since people tend to have multiple social network accounts and can, thus, become "social bridges". Furthermore, all sorts of networked objects are getting increasingly smart and social, giving rise to the so-called Smart Objects (SOs) and revolutionizing both the Internet of Things (IoT) and the Social Internet of Things (SIoT) [45]. Also, several SIoTs and IoTs cooperate with each other through "bridge" objects, thus generating new architectures, referred to in the literature as Multiple IoT (MIoT) [53].

The detection of anomalies in a single-IoT environment has been widely investigated [67, 667, 52, 393, 132], and many results involving privacy, security and fault detection have been found. However, to the best of our knowledge, no investigation on anomalies and their possible detection in a MIoT has been performed so far.

Increasing protection and autonomy of smart objects in the IoT. In the context of IoT, the protection of objects, on the one hand, and the possibility/need to guarantee them a great autonomy, on the other hand, represent two crucial issues to be addressed. As for *protection*, in [470] a first approach to address this issue was presented when it comes to privacy. Nevertheless, the problem of providing a scalable, reliable and protected framework for IoT devices remains open. As for *autonomy*, making objects independent from each other during their interactions requires the capability of adding/removing contacts recognizing what features/services are provided by other objects [512, 636]. At the same time, in this context, the possibility of assessing the ability of an object to *concretely* and *correctly* provide the needed feature/service is fundamental.

This reasoning highlights that autonomy and protection are two strongly interrelated aspects. In this scenario, the definition of trust and reputation mechanisms appears crucial [569, 148, 337, 149, 499, 198, 262]. However, most of the approaches proposed in recent literature describe strategies leveraging centralized services (such as whatchdogs) or particularly empowered smart objects, dedicated to data gathering from other objects and to the computation of trust and reputation values. Although these solutions may achieve pretty satisfactory results in some

cases, they somehow force the fully distributed and autonomous nature of IoT to include "global" monitoring points.

To achieve a fully distributed solution in this setting, each smart object should be able to build a pretty complete representation of other objects' behavior in the IoT. However, as a prerequisite, it should also be able to unequivocally link a sequence of actions (defining a behavior) to each object. This would require the definition of an authentication mechanism to map each action (e.g., a transaction) to the object making it. To address this issue, in the past literature, many authors have started to propose the use of the Blockchain technology in the IoT as a means to have a shared and reliable environment among all objects [170, 84, 206, 502, 400, 314, 539, 540, 565].

The application of Blockchain-based strategies to add trust and reputation facilities in the IoT without requiring any special actor (e.g., sophisticate smart objects) involved, poses a lot of interesting research challenges that must be faced to build a complete solution. One of the main problems is related to the high computational power required for deploying a Blockchain-based solution in the IoT context. Smart objects are intrinsically very heterogeneous and, therefore, provide a wide range of computation capacity spanning from fully equipped powerful devices (such as smart cars, new generation smartphones, etc.) to very simple, with minimal computational capacity, smart sensors (e.g., smart meters, medical sensors, fitness trackers, etc.). In such a scenario, including the Blockchain technology can be very tricky because solutions must include the possibility of both exploiting fully equipped and powerful devices and supporting very simple and computationally limited ones. Moreover, if we observe this problem from the Blockchain perspective, handling the big volume of transactions generated by smart objects introduces important flaws in terms of both scalability and environmental costs [140, 548]. To partially face these issues several researchers focused on the definition of lightweight Blockchains for the IoT. Typically, these approaches work on the reduction of the information necessary to mine and validate transactions published in the ledger by proposing alternative consensus algorithms [205].

For this reason, some authors proposed to reduce the transaction volume to consider in the public ledger by adopting approaches based on the adoption of validity windows [555]. In this way, smart objects must only work with the transactions available inside the chosen window. Depending on the analyzed application scenario, reducing the size of transaction history may introduce important drawbacks; indeed, for instance, if such a ledger should be used to store trust and reputation information of smart objects at the end of a validity window, each object can have a fresh start as its reputation will be restored. To avoid this issue, historic data can be aggregated

and made available inside each validity window; however, also this aggregation task can be very expensive and unfeasible for IoT objects if the volume of transactions is big [314].

Extending saliency maps and gaze prediction in an Industry 4.0 scenario. Year by year, more and more contents are available for people on the Web. For instance, during 2018, it is estimated that 1,500,000,000 websites, along with their related information, products, services, etc. were online[6]. In this "jungle", the capability of capturing the attention of a user when she visits a website is crucial. Indeed, the design of a website capable of effectively conveying the desired message could lead to an increase of popularity and, possibly, of returns for the corresponding company.

However, the evaluation of the attention paid by a person while watching a picture is not trivial and depends on several factors. Thankfully, it is possible to rely on two powerful tools to reach this goal. They are saliency maps [91] and visual scanpaths [274], which represent a formal definition of the areas where a user poses her eyes and the path made by her gaze, respectively. In the past, the first application scenario of these concepts was the one of natural images. However, with the increase of the number of websites available on the Internet, the interest on evaluating saliency maps and visual scanpaths also on websites has enormously increased. In fact, this last scenario is very valuable for a company, because it could increase its earnings if the website is able to capture user attention, in particular on the products/services it offers.

Scientific literature provides many approaches, belonging to different categories, to achieve this goal. In particular, in recent years, we have witnessed an important development of deep learning, which has impacted many research issues, including the prediction of saliency maps and visual scanpaths. One of the first proofs of the effectiveness of deep learning-based techniques in this setting is reported in [574]. After this attempt, several approaches involving neural networks have been proposed, and most of them achieved important results. In particular, an architecture that has recently gained a lot of attention and has several sophisticated applications is Generative Adversarial Networks (hereafter, GANs) [581, 299, 477, 211]. It is well-known that this architecture can be employed to address different issues and, thanks to it, satisfactory results have been obtained in many fields. Even in the prediction of saliency maps and visual scanpaths, GANs provided satisfying outcomes in the evaluation of user attention [488, 397, 43]. As a matter of fact, they achieved the state-of-the-art results in this field.

[6] https://www.internetlivestats.com/total-number-of-websites/

Actually, the vast majority of approaches involving GAN-based architectures for the prediction of saliency maps and visual scanpaths has been developed only for operating on natural images and not on websites. Indeed, to the best of our knowledge, as far as the web domain is concerned, few GAN-based approaches are able to evaluate saliency maps [397], and none of them can compute visual scanpaths.

1.2 General characteristics of the approach

1.2.1 Networking people

In this section, we present the general characteristics of our approach for Networking people. In particular, this section is divided in subsections, one for each topic.

Defining and detecting k-bridges. As for this issue, we use Yelp as the main reference network. The definition of k-bridges in Yelp starts from the hypothesis of seeing this social platform as a set of sub-nets or communities, one for each of its macro-categories. Actually, the importance of studying Yelp categories has already been highlighted in recent scientific literature [154]. Here, we want to go one step further and we consider that the communities associated with the macro-categories of Yelp are not independent from each other, because a user who reviews businesses of different macro-categories belongs to several communities.

Given the new concept of k-bridge, we show that k-bridges enjoy the anti-monotone property. Starting from this property, we propose a new algorithm for the extraction of k-bridges from social networks. Then, we provide a model for representing k-bridges in the social network they belong to and present three possible specializations of the concept of k-bridges for Yelp, Reddit and the network of patent inventors. We also present several important characteristics of k-bridges and shows that they are valid independently of the social network they refer to.

Finally, we show two use cases highly benefiting from bridges; the former regards the identification of the best targets of a market campaign, whereas the latter concerns the identification of new products/services to propose.

Detecting user stereotypes and their assortativity. As for this issue, we investigate subreddit and author stereotypes by evaluating author assortativity in this social platform. For this purpose, we build a dataset with all the posts published from January 1^{st}, 2019 to September 1^{st}, 2019, which we use for our analyses. We start with some preliminary investigations on Reddit data. They focus on three aspects, namely posts submitted to subreddits, comments under these posts and, finally, users who created a subreddit, posted or commented. The aim of this preliminary descriptive

analysis is not to discover new specific knowledge about Reddit. Instead, it allows us to better understand the dataset, and to check if some theoretical trends, which should have characterized these aspects on Reddit, are verified on it. Furthermore, the results found, which are partially expected, represent the starting point of the next knowledge detection activities. They are also useful to explain the knowledge patterns extracted.

After this preliminary analysis, we discuss our investigation on how to stereotype subreddits. For this purpose, we first investigate the lifecycle of a subreddit, depicting its typical characteristics. Then, starting from this, we identify several subreddit stereotypes and, then, we define and apply three orthogonal taxonomies in order to characterize them. After the analysis of subreddit stereotypes, we proceed similarly for Reddit authors. In particular, we extract several author stereotypes and, then, we classify them according to some orthogonal taxonomies that we defined for this purpose.

The last activity is devoted to verify the possible existence of a degree assortativity in Reddit. We recall that the assortativity in a social network expresses the inclination of a node to associate with other ones that are somewhat similar. Assortativity has been largely investigated by social media analysts [462, 109]. We aim at performing this analysis for Reddit authors and degree assortativity to verify if authors very active in Reddit tend to form a backbone or not.

Detecting backbones of information diffusers among different communities of a social platform. As for this issue, our approach exploits a support social network that records the users of interest, along with their activities performed in different communities. Starting from this social network, it aims at identifying the so-called disseminator bridges, i.e., users particularly active and organized in disseminating information on several communities. To identify such users, it exploits the concept of centrality, which has always been considered a key concept in Social Network Analysis [613]. More specifically, it first uses a combination of three classic centrality measures well known in Social Network Analysis, namely degree centrality, closeness centrality and betweenness centrality. Then, it introduces a new centrality measure specifically designed to better identify the backbone of users of interest. We call this new centrality measure *disseminator centrality*, and its definition represents another main contribution of this paper. Finally, starting from the new disseminator centrality, our approach allows the identification of possible backbones of disseminator bridges.

To validate all the technical aspects introduced, we propose an experimental campaign through which we show that disseminator centrality is more effective in

identifying disseminator bridges than classical centralities or a combination of them. The same campaign allows us to better understand the characteristics of backbones of disseminator bridges.

Investigating NSFW contents and their authors. As for this issue, we conduct in two different ways. The first one is a semantic analysis in three phases. During the first one, called "Data Cleaning and Annotation", we perform the classical ETL operations on NSFW posts and comments taken from Reddit. In addition, we appropriately enrich the cleaned and reorganized posts and comments by associating lexical and sentiment annotations to them.

During the second phase, called "Pattern Extraction and Enrichment", we extract a set of text patterns from the previously annotated NSFW posts and comments. In our context, a pattern is a set of words in posts and comments that satisfy certain properties. In this phase, we first extract the most frequent patterns. Then, we associate each of them with a rich set of features.

During the third phase, called "Network-based Pattern Analysis", we use the patterns selected in the previous phase to construct three social networks. The first allows us to identify, and possibly study, communities of users who exchange NSFW adult posts and comments. The second allows us to analyze groups of text patterns frequently appearing together in NSFW content. These groups represent a starting point for a study of the language adopted by users in this kind of posts and comments. The third can be seen as a step beyond the first because it helps us to extract virtual communities of users adopting the same text patterns.

The second analysis is a structural one. Similarly to the first analysis, it is divided in three phases. The first one is a "Descriptive Analysis" and aims to study the distributions of the entities involved in the phenomenon (e.g., the distribution of NSFW posts against subreddits, authors, score and comments).

The second phase is a "Social Network Analysis" and aims to study the co-posting phenomenon, and therefore the interactions between authors of NSFW posts.

The third and last phase is called "Assortativity Analysis"; it aims to extend and deepen the previous analyses to discover and study whether possible forms of assortativity [462] exist among the authors of NSFW posts. Recall that assortativity is a particular case of homophily in social networks [435], which indicates the tendency of a node to cooperate with nodes having similar characteristics.

Investigating negative reviews and negative influencers. As for this issue, we first define a multi-dimensional social network-based model for Yelp and then use it to study negative reviews and build a profile of negative influencers in this social

medium. We decided to adopt this model because it perfectly fits the specificities of Yelp mentioned above. In fact, our model represents Yelp as a set of 22 communities, one for each macro-category of this social platform (modeling Yelp as a business directory). At the same time, it represents Yelp as a social network, whose nodes indicate users and whose arcs denote the relationships between them. These can be of different types. For example, they can denote friendships between users (modeling Yelp as a social network), or the action of co-reviewing the same business (modeling Yelp as a review platform). Through the concepts and techniques of Social Network Analysis applied to our multi-dimensional model, our approach defines three stereotypes of Yelp users, namely the bridges, the double-life users and the power users. These stereotypes can help the detection of the negative influencers in Yelp and the definition of a profile for them. These last are completed by a Negative Reviewer Network, which allows us to investigate the main characteristics of the negative influencers in Yelp.

Among the possible questions that can be answered thanks to our approach, we focus on the following ones: *(i)* What about the dynamics leading a Yelp user to publish a negative review? *(ii)* How can the interaction of these dynamics increase the "power" of negative reviews and people making them? *(iii)* Who are the negative influencers in Yelp?

Investigating user behavior in a blockchain during a cryptocurrency speculative bubble. As for this issue, our approach starts with the definition of four categories of users. These stereotypes derive from a descriptive analysis of a dataset of transactions in Ethereum. The categories are the following:

- *The power addresses*, i.e., the most active users on Ethereum, who were responsible for most of the transactions of this network. More specifically, we consider the power addresses for each of the periods of interest (i.e., the pre-bubble, bubble and post-bubble).
- *The Survivors*, i.e., those users who were power addresses in all the three periods of interest.
- *The Missings*, i.e., those users who were power addresses in the pre-bubble period and stopped being power addresses in the bubble and post-bubble periods.
- *The Entrants*, i.e., those users who were not power addresses in the pre-bubble period and became power addresses in the bubble and post-bubble periods.

Then, for each user category, we employ Social Network Analysis based techniques to identify the main characteristics that distinguish the corresponding users from the others. In this activity, the concept of ego network [197] plays an important role.

Afterwards, we check if and when there are backbones linking the users of a certain category. The presence of such backbones can be hypothesized on the basis of the principle of homophily [435], characterizing many social networks. However, only a set of experimental analyses can indicate whether this hypothesis is true or not. Also in this case, ego networks play a key role to support analytical investigations. They are flanked by k-cores [204], which help in giving a graphical idea of the analytical results. Finally, we aim at predicting, given a certain period (i.e., pre-bubble, bubble), who will be the main actors in the next ones (i.e., bubble, post-bubble), based on some parameters. This part ends with an analysis aimed at understanding how the users of the various categories have behaved in the months following the ones considered in our investigation, i.e., from the beginning of 2019 until today.

Representation, detection and usage of the content semantics of comments. As for this issue, we propose a data structure and a related approach to extract content semantics from a set of comments. In our experiments, we focus on Reddit comments and posts. However, our approach is general and can also be employed in other social platforms. The activities that our approach performs on comment content are many, but they can be grouped into two phases, which we can call "pre-processing" and "knowledge extraction".

The pre-processing phase aims at cleaning and annotating available comments and, then, selecting the most significant ones. Cleaning is necessary to remove bot-generated content, errors, inconsistencies, etc., as well as to perform tokenization and lemmatization of comments. Annotation allows important information to be added to each lemmatized comment automatically. Examples of this information are the sentiment value associated with the comment, the post which it refers to, the author who wrote it, etc.

Filtering is based on text pattern mining tasks and is used to identify the most significant lemmatized and annotated comments. In order to carry out this activity, our approach takes into account not only the frequency of patterns, as most of the approaches proposed in the past literature do [10, 246, 248], but also, and above all, their utility [248, 11, 448, 259, 354], measured on the basis of a utility function. Interestingly, our approach is orthogonal to the utility function used and, therefore, choosing different utility functions allows it to give priority to certain properties of comments instead of other ones. A first utility function could be the sentiment of the comments in order to select, for instance, patterns involving only positive comments or only negative ones. A second utility function could be the comment rate, which would allow our approach to select, for example, patterns involving only high rate comments or only low rate ones. A third utility function could concern the Pearson's

correlation [495] between sentiment and rate, which would allow it to select, for instance, patterns involving only comments with discordant sentiment and rate or only comments whose sentiment and rate are in agreement with each other.

Once the comments and patterns of interest have been selected, our approach defines a data structure for their representation, which we call CS-Net (Content Semantics Network). The nodes of a CS-Net represent comments' lemmas. Its arcs can be of two types, reflecting two different perspectives of viewing content semantics. The first is based on the concept of co-occurrence and considers that two semantically related lemmas tend to appear together very often in sentences. It summarizes the results of many researches carried out in the field of Information Retrieval [202]. The second concerns the concept of relationships and semantically related terms. It summarizes many researches carried out in the field of Natural Language Processing [95]. The CS-Net model is extensible so that, if we want to consider further content semantics perspectives in the future, it will be sufficient to add another type of arcs for each new perspective.

Defining user spectra to classify user behaviors in cryptocurrencies. As for this issue, the starting point of our approach is that each Ethereum user has a wallet in order to carry out her activities. A wallet is identified by an address, an alphanumeric code allowing it to be recognized in the blockchain network and to carry out transactions with other wallets or to interact with smart contracts. All the transactions made by a user in a certain time period allow us to reconstruct, at least partially, her behavior in that period.

More specifically, in order to define user behaviors in a certain time interval, our approach first builds a social network representing the users involved in Ethereum and their transactions. Then, starting from this social network, it defines and computes a set of features for each user. They are the number of incoming and outgoing arcs of the node corresponding to the user, the number of incoming and outgoing transactions, the amount of incoming and outgoing money (expressed in Ether, the Ethereum cryptocurrency), the clustering coefficient and the PageRank. The values of these features can change over time. Given a time period T and a user u_j, we call the spectrum of u_j in T the set of time series expressing the values of the features for u_j in T. The spectrum of u_j provides a concise, but accurate, picture of the behavior of u_j during T.

Having a spectrum for each user might lead to think that categorizing users is a simple task. In fact, in principle, one could build a spectrum for each class starting from the spectra of the users belonging to it, identified from the training data. At this point, given a new user, whose spectrum is known, she could be assigned to the

class with the spectrum most similar to her own. Although this procedure seems simple at an abstract level, it is much more complex in reality. In fact, we have seen that the spectrum of a user (and, consequently, the spectrum of a class) consists of a set of time series, one for each feature. As a consequence, it is necessary to define a similarity measure between two sets of time series. Furthermore, the various features are not totally independent of each other. In fact, a correlation study on them showed us that some features are totally or partially correlated. Therefore, the spectrum of a user must be managed as a multivariate time series.

As a consequence, we must face a classification problem in which each element to classify and each available class are represented by multivariate time series. To the best of our knowledge, there is no out-of-the-box classification algorithm with these characteristics. Thus, it is necessary to define a new one. The core of such an algorithm consists of a metric capable of measuring the similarity degree between two multivariate time series (which, in our case, are the spectrum of the user to be classified and the spectrum of each class). Several metrics proposed for this purpose exist in the literature. Among them, we mention the Dynamic Time Warping [75], the Weighted Sum SVD [554], and the Eros distance, also known as Extended Frobenius Norm [653]. The latter has been shown to outperform the other more traditional metrics [653]. Hence, it would represent the natural choice in our case. Unfortunately, the results obtained by applying the Eros distance to our reference scenario were not satisfactory. However, we managed to define a variant of it. Even if more expensive in terms of computation time (albeit, as we shall see, these costs are largely acceptable), this variant achieves a very high classification accuracy.

Extracting information from posts on COVID-19. As for this issue, our approach aims to extract information from posts on COVID-19 published on Reddit. In particular, we propose three approaches. The first is a hierarchical classification algorithm for the posts on COVID-19 published in Reddit. The second is an algorithm capable of identifying a set of homogeneous themes regarding the COVID-19 disease discussed by users. The third is an algorithm capable of identifying a number of user communities showing homogeneous interests. We applied these three approaches to all the posts related to COVID-19 published in Reddit from January 9^{th}, 2020 to April 30^{th}, 2020. The number of posts considered is almost two and half million.

The three approaches proposed have been conceived with reference to COVID-19. However, we point out that they are general and can be used to extract information about any other issue that may cause an intense posting activity on Reddit.

Extracting time patterns from the lifespans on TikTok challenges. As for this issue, given a TikTok challenges dataset we crawled, we analyze their lifespans to extract time patterns that allow the classification of challenges into dangerous and non-dangerous ones. By the term "lifespan" we do not mean the time interval between the moment a challenge is launched and the one it disappears permanently. In fact, there are challenges that never disappear even though they have not been active for a long time. From our point of view, the lifespan of a challenge is the period that elapses from the time it is launched to the time it is no longer capable of eliciting at least limited interactions with users. The classification approach we propose in this paper is currently able to support the detection of dangerous challenges only near the end of their lifespan, or at least after a presumably long time period. On the other hand, the early detection of dangerous challenges is not our objective here. In fact, we want to propose a *challenge classification* approach that, once has its validity verified, represents *a first step* in the direction of early detection of dangerous challenges. To reach the latter goal, in the future, we can think of greatly reducing the granularity of the time intervals taken into account (which, as we will see, is currently coarse) in such a way as to identify the time patterns allowing the detection of the dangerous challenges at an early stage.

To perform our analysis, we followed the evolution of seven non-dangerous and seven dangerous challenges. For each challenge, we considered the corresponding videos posted, and, for each video, we considered a set of features (e.g., duration, number of likes received, number of followers of its authors, etc.). Next, we defined a social network-based model to represent a TikTok challenge. At this point, we began our analysis to find features capable of distinguishing non-dangerous challenges from dangerous ones. First, we focused on the characteristics of the videos and the basic structural parameters of social networks (for example, number of nodes, average clustering coefficient, density, etc.). Then, we considered the challenge lifespans and could see that the two types of challenges showed very different lifespans.

In order to capture such a difference, we divided each lifespan into suitable intervals. After that, we performed a clustering activity to group intervals into homogeneous clusters. To define the characteristics of each cluster, we used the properties of the videos and social networks corresponding to the challenges, which the cluster's intervals referred to. Then, we defined the sequence of intervals that characterized the lifespan of each challenge. From the examination of such sequences, after a further study aimed at demonstrating that some clusters were substantially equivalent to each other, we were able to determine a time pattern that characterized all the non-dangerous challenges and three time patterns that could be found in the dangerous ones. Finally, we verified if what we had found with the 14 initial challenges

was valid in general. To do this, we performed two further tests with a much higher number of challenges and were able to verify that our results were very accurate also in this case.

Investigating community evolutions in TikTok. As for this issue, we focus again on TikTok and we study the characteristics of the communities participating in dangerous and non-dangerous challenges, the behavior of the corresponding users and their dynamics and evolution over time. The final goal is the possible detection of evolutionary patterns allowing the distinction of non-dangerous challenges from dangerous ones.

To perform our analysis, we selected seven non-dangerous challenges and seven dangerous ones. For each of them, we considered the corresponding posted videos and a set of features characterizing the associated user communities (e.g., number of connected components, size of the maximum connected component, average clustering coefficient, average path length, etc.). Next, we defined a social network-based model to represent the user community associated with each TikTok challenge. Using this model, we began our analysis to investigate the evolutionary dynamics of the communities associated with non-dangerous and dangerous challenges. First, we focused on the characteristics of their videos and the parameters of the social networks associated with their communities. From a first analysis, taking into account the evolution of the community size during the challenge lifespans, we could observe that non-dangerous and dangerous challenges seemed to show different dynamics.

To capture these differences, we divided challenge lifespans into suitable intervals. Then, we grouped these intervals into homogeneous clusters. At this point, for each cluster, we used the values of the Social Network Analysis parameters characterizing the communities corresponding to the intervals belonging to it for drawing the cluster's profile. After this, for each challenge, we identified the sequence of intervals, along with the corresponding clusters, which formed its lifespan. From examining these sequences and the characteristics of the corresponding clusters, we hypothesized that some clusters were substantially equivalent and verified the correctness of this hypothesis by means of a t-test [100].

After verifying this correctness, we could simplify the sequences related to challenges, and this allowed us to identify a main evolutionary pattern characterizing non-dangerous challenges, and two main evolutionary patterns, different from the previous ones, characterizing dangerous challenges. This result provides a new way to distinguish non-dangerous challenges from dangerous ones. After obtaining this result, we tested whether it was accurate and generalizable to the other challenges of TikTok. To this end, we considered 300 challenges and were able to verify that

our model was very accurate also for this sample, much larger than the one initially used.

1.2.2 Networking things

In this section, we present the general characteristics of our approach for Networking things. In particular, this section is divided in subsections, one for each topic.

Networking wearable devices for fall detection in a workplace. As for this issue, we propose a framework for safety in the workplace whose foundations consist of Sentient Multimedia Systems and Machine Learning. This framework consists of three distinct levels, namely: *(i) Personal Devices*, which are smart objects worn by workers (e.g., safety glasses, protective gloves, etc.); *(ii) Area Devices*, which are fixed smart objects associated with a specific area (e.g., access control gates, devices for controlling environmental parameters); *(iii) Safety Coordination Platform*, which monitors the safety of the working environment and, if necessary, activates the appropriate alarms and provides the related advices.

The design of the framework proposed in this paper is done at an abstraction level that allows it to be used in any working context and to address any safety issue. However, in order to give a very concrete idea of how it could operate in a real context, we also illustrate its specialization to a particular scenario, very studied in past literature, which is fall detection.

In fact, some of the main causes of injuries and deaths in the workplace are slips, trips and falls. Our framework adopts a new, very advanced wearable device, based on Machine Learning, which we designed, built and tested and that we describe in detail in this paper. Furthermore, it employs existing smart objects for Area Devices. Finally, it adopts an appropriate chain of Machine Learning based modules for the management of the Safety Coordination Platform.

Anomaly detection and classification in Multiple IoT scenarios. As for this issue, we propose a framework that models anomalies and the corresponding features in a MIoT by providing a multi-dimensional view, based on three orthogonal taxonomies: *(i)* presence anomalies vs success anomalies; *(ii)* hard anomalies vs soft anomalies; and *(iii)* contact anomalies vs content anomalies. Each combination of the possible values of these dimensions gives rise to a specific type of anomaly to investigate, for instance the *Presence-Hard-Contact* anomalies. Furthermore, anomaly definitions are orthogonal to specific anomaly detection approaches, past or future, which may be applied (and will be combined) in the context of our framework.

Together with the multi-dimensional taxonomy, another main component of our framework is the extension of conventional methodological frameworks to the MIoT case. Our framework has been conceived to address two problems, known as the "forward problem" and the "inverse problem", respectively. In the forward problem, we aim to analyze the effects that multiple anomalies have onto the MIoT. On the other hand, in the inverse problem, which is traditionally more complex, we aim at detecting the source of the anomalies (i.e., the objects that have generated them) based on the effects that these have on the objects or their connections.

In order to show the possible usage of our framework, we present a case study centered around a smart city. Furthermore, in order to evaluate our framework and extract knowledge, we have conducted a series of tests. These allowed us to find several important knowledge patterns about anomalies and their effects in a MIoT. Our most important findings may be summarized as follows: *(i)* the effects of the anomalies of a node rapidly decrease as the distance from the node itself increases; *(ii)* anomalies are less evident in a MIoT than in a single IoT; *(iii)* the number of anomalous nodes increases as the number of IoTs increases, in a roughly linear way; *(iv)* the outdegree of anomalous nodes has a great impact on the spread of the anomaly over the MIoT; *(v)* closeness centrality is even more important than degree centrality in the spread of anomalies; *(vi)* the computation time necessary for the detection of anomalous nodes is polynomial against the number of MIoT nodes; *(vii)* the time necessary for evaluating the effects of anomalies in a MIoT is quadratic against the number of its nodes.

Increasing protection and autonomy of smart objects in the IoT. As for this issue, we propose a two-tier Blockchain framework to increase the protection and autonomy of smart objects in the IoT. Following the intuition proposed in [470], we consider smart objects as organized in communities. Hence, the first, local, tier is used to manage the trust measures of each smart object inside the community it belongs to and exploit a solution leveraging both a lightweight Blockchain and a validity window to control transaction volume. By organizing objects into communities, we can control the size of the local Blockchain in order to avoid excessive loads for smart objects. The second, global, tier is used to record aggregated data related to the individual communities, as well as the trust value that each community assigns to the other ones.

By definition, communities are built by looking at both the heterogeneity and the redundancy of provided features/services (so that multiple objects in the same community can offer the same feature/service). In a community, a smart object may require information to another smart object of the same community about the fea-

tures/services offered by it. In order to estimate the latter's reliability, and ultimately its reputation in the community, our approach adopts a solution based on a probing mechanism. In particular, nodes are tested using probing queries about features/services they can provide. Their answers are then compared with those received by other nodes capable of offering the same features/services. This comparison allows the computation of the reliability of the tested object in providing the features/services declared. All transactions made to assess the reliability of smart objects in a community are stored in a Blockchain with a dedicated smart contract.

After a certain time window, our framework computes the reputation of each object inside its community. At the end of this process, smart objects that do not meet the minimum reputation level are removed from the community. Then, for each community, a transaction with the list of its smart objects, along with their reputation, is stored in the Global Blockchain. In this way, the Local Blockchain is reset, following the approach described in [555], and all transactions occurring in the time window just passed are no longer considered.

Our approach also ensures protection when smart objects from different communities interact with each other. The procedure used in this case is similar to the one seen above. The results of a test performed by a smart object on another are stored in the Local Blockchain of the community the trustor object belongs to. Also in this scenario, after a certain time window, these transactions are aggregated and used to compute the trust of a community in another one. The trust values of each community in the other ones are stored in the Global Blockchain. Therefore, this last contains the reputation of each smart object in its community, as well as the trust of each community in the other ones it interacted with in the past. If there has never been an interaction between two communities, our approach assumes that each of them assigns a default trust value to the other one.

To perform the tasks described above, we use smart contract technology in the Blockchain. Indeed, Blockchain smart contracts are already being used to manage, control and secure IoT devices [346]. In particular, they can provide decentralized authentication rules and logic to implement single and multi-party authentication for an IoT device. They have been adopted to guarantee trustworthy and authorized identity registration, ownership tracking and monitoring of products, goods, and assets [482]. Their applications in IoT are discussed in [170], where the authors describe how Blockchain smart contracts can facilitate and support autonomous workflow and service sharing among IoT devices.

Moreover, through a deep experimental campaign, carried out leveraging real-life smart object data and Ethereum transactions, we prove that our approach is fea-

sible and allows for the detection of compromised nodes in a relative small amount of time strictly related to the chosen probing frequency.

Extending saliency maps and gaze prediction in an Industry 4.0 scenario. As for this issue, the approaches we propose here are variants of GAN-based approaches presented in the past literature and are specifically designed to work on websites. As will be clear below, the starting approaches (i.e., SalGAN [488], for saliency map prediction, and PathGAN [43], for gaze path prediction) were originally designed to operate in the context of natural images. Actually, the context of web pages is much more complex because more natural images, along with texts, logos and animations, can be simultaneously present in a single web page. The peculiarities of web pages make traditional computer vision saliency detection methods, such as the one described in [671], much less effective when applied to them than to natural images. The reason is that a web page presents several salient stimuli and competitions, which make it hard to accurately predict eye fixation [563]. We defined three variants of SalGAN, for saliency map prediction, and two variants of PathGAN, for gaze path prediction. As we will see below, the best variant of SalGAN and the two variants of PathGAN are fine-tuned. In addition, they present several other refinements taking into account various observations we made during some experiments conducted "on the field". At the end of all these activities, we managed to achieve: *(i)* a SalGAN-based approach for website saliency map prediction that has a better performance than existing approaches carrying out the same task; *(ii)* two PathGAN-based approaches that, to the best of our knowledge, are the first ones proposed in the literature for gaze path prediction on websites.

In order to provide deep and accurate training, testing and evaluation of our approaches, we preliminary strived to create a new complete dataset, which could represent all the interface heterogeneities currently found in the web. In fact, existing datasets have some limitations in this aspect (see below). In order to construct such a dataset, we started from a popular existing one called FiWI (Fixations in Webpage Images)[563], and enriched it with new web pages, more in line with the current graphical standards, and new people involved. As we will see below, this much more complete dataset increased the quality of training, testing and evaluation of our approaches significantly.

Using our dataset, we tested: *(i)* SalGAN and all our variants, in order to verify if one of them has a better performance than the others and several related approaches proposed in the past; *(ii)* PathGAN and its two variants, in order to verify if one of them has a better performance than the original PathGAN. In both cases, we obtained a positive result.

We implemented our approach in a user-friendly web application. In this way, a designer can easily upload her web page and, then, know in advance the behavior of future visitors when accessing it. Indeed, our web application returns both the saliency map and the gaze path of visitors accessing the uploaded web page. A designer can leverage the information returned to improve the user interface by moving its objects accordingly and verifying again the visitors reaction. The adoption of our tools allows web designers to reduce the number of meetings with the final users for evaluation purposes, which leads to save a huge amount of time and money.

1.3 Related works

In this section, we illustrate the state-of-the-art behind the methods and approaches presented. It is organized on the basis of the two strands and the different subcontexts we have seen in Section 1.1.

1.3.1 Networking people

Studying the behavior of users in social platforms is a fundamental aspect to understand the dynamics underlying the diffusion and the growth of these systems [334, 682].

A lot of research has been devoted to understanding how users interact with each other in social media and how information diffusion takes place inside the latter [42, 627, 649, 74].

The interaction among users has been studied by leveraging several information available in these social systems, ranging from existing public friendship relationships to posting the same piece of information [546, 77, 7].

The study of social networks has rapidly become a core research field, thanks to its interdisciplinary aspects [415, 179, 203, 24, 179, 104, 127]. Indeed, many researchers of different disciplines, such as computer scientists, sociologists and anthropologists, exhibited a huge interest in Social Network Analysis [434, 111, 156].

Defining and detecting k-bridges. Many studies have proved that, in a social platform, there exist different categories of users, each participating to it with different levels of activity and heterogeneous contents [69, 422]. Of course, when dealing with user interactions, it is important to consider also those that cannot be examined homogeneously [116].

This rises the necessity of analyzing data of each social medium by decomposing it in different networks of relations. Multi-relational networks have been largely

investigated in the past [600, 193, 656, 675]. An interesting approach in the field of multi-relational networks is the one proposed in [677]. Here, the authors combine the analysis of the friendship network with a study of the author-topic network, both built from the information available in an Online Social Network.

Considering each social medium as a set of overlapping relational networks also opens important consequences in the role of each user inside these platforms [600, 339, 276].

The interest towards users serving as bridges among communities has increased over the years so that several studies have been performed to analyze the behavior and peculiarities of such users in complex networks [249, 567, 388, 33].

Studying nodes bridging communities together has been also a crucial research direction in the context of multi-relational networks [71, 73]. Here, the heterogeneity of the scenario is more evident because of the different nature of the relations considered. In particular, the authors of [71] report a complete analysis of bridge users among multi-relational networks.

Several studies also investigated the behavior of users serving as bridges among different social networks [103, 110, 106].

Detecting user stereotypes and their assortativity. In the past literature, approaches for the characterization and identification of specific traits of users have been presented in different papers. Some of the considered traits are: users presenting multi-community engagement [603], anti-social behaviors [191], community opposers [370], "answer-persons" [112], and "explorers" [302]. For example, social and anti-social behaviors are analyzed in [191], where the authors apply a definition that extends Brunton's construct of spam in order to separate norm-compliant behaviors from norm-violating ones. This approach also investigates inter-community conflicts by associating social and anti-social homes to users. Conflicts between users are also studied in [370]. The authors of [112] explore the presence of users showing the trait of "answer-person". The authors of [302] present a study regarding highly related communities; in this analysis, they define the characteristics of explorers and non explorers by adopting a specific taxonomy.

The studies and approaches outlined above have been developed considering several communities. In [364], a specific community about online User Experience is studied. Here, members socialize and learn together. The authors of [364] identify five distinct online social roles, namely the "knowledge broker" (i.e., a member that introduces knowledge to the community by sharing links), the "translator" (i.e., a member that offers her academic knowledge into the community), the "conversation facilitator", the "experienced practitioner", and the "learner".

Assortativity of users has also been analyzed in the past [109, 17, 107]. In particular, in [293], the authors focus on studying loyal communities, finding that they tend to be less assortative as long as their interaction level increases. In [241], the authors discuss the rise of new trends in complex networks by looking at vertices that "shine" (i.e., high-degree vertices), also called network stars. They study the evolution of some complex networks, with Reddit among them.

In this context, Reddit is an invaluable source of information, insights and research possibilities. Indeed, it is a prosperous environment, where users share contents and interact with each other. The heterogeneous nature of Reddit, together with the openness and the richness of its data, encouraged scientific community to explore the twists and turns of this platform. The swift increase of scientific literature related to Reddit has produced a high number of papers with several goals and methodologies [436, 576].

Detecting backbones of information diffusers among different communities of a social platform. The investigation of how information is disseminated is a core problem in Social Network Analysis. As an evidence of this, there are many studies to identify users involved in information dissemination [421, 54, 615, 81, 42, 649, 352, 80, 401]. Not all users contribute to this activity in the same way. Think, for instance, of bridges, who are necessary "gateways" when information must be disseminated from one community to another. There are users, often referred to as power users, who have many links and are, therefore, able to disseminate information easily within a community. Traditionally, the problem of information dissemination is closely related to the concept of centrality [613]. As stated above, identifying bridges can help to extract different information patterns about a given phenomenon [179, 263, 625]. As for central users, several measures were proposed in the past literature. Some of them are classical and extremely general (think, for instance, of degree, closeness, betweenness and eigenvector centralities) [517] while others were introduced to address specific problems.

In the analysis of this topic, an increasing number of researchers are studying the role not only of classic and direct relationships, such as friendship, but also several other ones, such as co-posting or homophily of interests (i.e., having interest in the same topics) [546, 77].

A backbone is a set of nodes that are central, i.e., important in the studied scenario. Network backbones are highly investigated in the literature, because, as the social network sizes grow, support structures are increasingly needed to store essential information and leave out the other [51, 424]. Research in this field focuses on how to identify backbones and their users [141, 282]. The authors of [141] present

an approach to discover backbones on a traffic network combining its structural and functional information. The latter takes into account the activities performed by users. The approach returns a backbone with a small number of arcs but capable of supporting a large number of different activities. Many approaches to find backbones have been proposed in the past literature [420, 371, 516, 543, 685, 643, 542, 654, 586].

Investigating NSFW phenomenon. The term "NSFW" was first proposed in 1998 and is one of the oldest acronyms of the Internet. It refers to content that is not suitable to be viewed in a working environment. Since then, different online systems, like Twitter, WhatsApp, many forums, and Reddit, have adopted this term to label sections with posted content not adequate for everybody and, in general, not suitable for public and professional contexts. Specifically, Reddit has introduced a dedicated group of contents called NSFW to separate posts suitable to be enjoyed in any context from those that should be watched in private environments.

Even if the contents of NSFW posts are considered side-contents to be kept separated from front-end ones, several researchers have started to study the characteristics of these contents, as well as the communities underneath them [433, 123, 609, 683, 181, 277].

A high-level analysis of the research efforts in the context of NSFW content allows us to distinguish two main directions. The former focuses on understanding the main characteristics of people publishing or viewing such materials, as well as the features of the NSFW content itself. The latter, instead, uses features of NSFW content to build content detection and filtering solutions, often with the objective of enabling/disabling the visualization of this material for users.

In particular, the work described in [609] is an example of the first research direction. Here, the author investigates the role of images and selfies in NSFW contents of `tumblr.com`. Another contribution in the first research direction is the one reported in [181]. In this paper, the authors try to understand both the nature of the content posted in anonymous social media and the difference between NSFW content posted in these media and in non-anonymous ones (like, e.g., Twitter).

As for the second research direction mentioned above, several works have been published in the recent scientific literature [457, 195, 79, 683, 176]. For instance, the work described in [457] focuses on the protection of minors accessing the Internet from the exposure to unwanted and harmful contents.

The issue of classifying video content as NSFW is addressed in [195]. In this paper, the authors exploit Convolutional Neural Networks (CNNs) for extracting audio-video patterns from NSFW videos. Similarly, the approach of [79] makes use

of a deep neural network-based solution to identify content belonging to the NSFW category. Finally, the approach described in [176] aims at building a classifier for detecting NSFW content by looking at images and visual material in the post. The authors prove that their solution outperforms the state-of-the-art solutions based on single CNN models. For this purpose, they present a deep comparison on a manual labeled dataset.

Actually, the analysis of NSFW and adult content has mainly focused on finding potentially offensive and dangerous content [160, 175, 345, 579]. Only few authors have studied this phenomenon on Reddit [433, 457, 180], despite it is one of the few social media that have adopted the concept of NSFW in an explicit and well-structured way.

NSFW posts on Reddit are studied in [180]. Here, the authors focus on the characteristics of NSFW posts and highlight the differences from the ones of SFW posts. They employ several descriptive analyses for extracting useful information to understand the dynamics behind the authors and the readers of such posts, as well as the subreddits in which they are published. They show that the NSFW communities of Reddit are characterized by very cohesive authors who, at the same time, are very open to new ones. In [433], the authors study the different behavior between moderators of NSFW and SFW subreddits. In particular, they focus on a protest carried out by moderators of several Reddit communities. In [160], the authors focus on finding "adult accounts" on Twitter. These are defined as accounts disseminating sexually explicit content. To achieve their goal, they construct a graph connecting accounts and entities contained in tweets. The authors of [579] aim at characterizing and predicting adult content spammers on Twitter.

Investigating negative reviews and negative influencers. One of the most investigated review platforms is Yelp. In recent years, it has received a lot of attention from the scientific community. The corresponding papers can be classified in the following groups, according to their goal: *(i) Rating Analysis*: It includes the investigations that analyze the dynamics describing how rates are assigned to businesses in Yelp [114, 311, 386, 597, 187, 580]; *(ii) Review Analysis*: It comprises the works focused on the analysis of reviews and of what events drive the users writing them [614, 599, 490, 491, 64, 285, 610, 411]; *(iii) Sentiment Analysis*: It also deals with the analysis of reviews, but with a specific focus on their content from a sentiment point of view [455, 537, 38, 284]; *(iv) Fake review and rate discovery*: It includes the proposals dealing with the detection of fake reviews and rates [412, 451, 425, 385]; *(v) Recommender Systems*: It comprises all the research works devoted to providing Yelp users with recommendations about suitable businesses, other users to interact with,

and even text suggestions for new reviews [631, 366, 212, 154, 626]. Their research efforts have also been supported by the social medium itself, which has made available a complete snapshot of its data to foster comprehensive analyses on it [185].

Several authors have investigated Yelp using Social Network Analysis (SNA, for short) [508, 509]. As for the analysis of social relationships, several studies have been conducted in both Yelp and other social platforms to understand how users perceive their social contacts and how they influence their acquaintances [398, 474, 287, 456, 566, 333, 686, 676].

The authors of [398] investigate the effects of the review rate, the reviewer profile, and the receiver familiarity with the platform, on the credibility of a review on Yelp. Moreover, the authors of [474] find a strong correlation between the moral attitude of a community of users and their tendency to express low rates and negative reviews in case some moral foundation is violated.

In multi-relationship networks, the classical definition of influencer is extended because the role of such users is not bound to communities derived from a single category of relationships. Instead, it also includes the capability of providing information diffusion channels among different networks, one for each type of relationships. To refer to this extended definition of influencer, the term "bridge" is often adopted. In the past literature, several studies have been devoted to investigating the role of bridges in the formation of social communities. For instance, the authors of [339] show that users with a weak connection bridging heterogeneous groups have higher levels of community commitment, civic interest, and collective attention than the other ones.

Investigating user behavior in a blockchain during a cryptocurrency speculative bubble. Since the introduction of Bitcoin in 2008 [532], thousands of cryptocurrencies have been created [177], and the interest about them has increased significantly. At the same time, the scientific literature about Blockchain and digital currencies has progressively grown [394, 225, 39, 596, 607]. The spread of this new technology has also created a lively discussion in the economic field on the possibility of speculations around these assets [134, 49, 142, 96].

Indeed, at the end of 2017, the price of Bitcoin (as well as the ones of the other cryptocurrencies, like Ethereum or Litecoin) increased by almost 600% (reaching an all-time high value of $19,475.80) before falling by 80% in few weeks, until January 2018 [658, 409, 85, 227]. This is the biggest speculative bubble in the cryptocurrencies history so far. Researchers have strived to analyze every detail of this particular event to understand the corresponding dynamics in order to prevent other speculations in the future. For instance, in [659], the authors investigate market ef-

ficiency and volatility persistence in 12 highly priced and capitalized cryptocurrencies, based on daily data from August 7^{th}, 2015 to November 28^{th}, 2018. In [178], the authors examine the existence and the time intervals of pricing bubbles in Bitcoin and Ethereum.

The prediction of speculative bubbles in the cryptocurrency scenario is investigated in [161, 252, 584]. In [270], the authors introduce an automatic peak detection method that classifies price time series into periods of uninterrupted market growth (i.e., drawups) and periods of uninterrupted market decrease (i.e., drawdowns). In [501], the authors investigate a new approach to predict speculative bubbles involving four cryptocurrencies (Bitcoin, Litecoin, Ethereum, and Monero) based on the behavior of new online social media indicators. In [144], the authors propose another possible way to detect speculative bubbles in cryptocurrencies, i.e., through an approach based on a social microblogging platform for investors and traders.

A further approach to investigate the cryptocurrencies market is based on the analysis of the corresponding blockchain. It starts from the consideration that a blockchain represents a public ledger in which all committed transactions are stored in a chain of blocks [684, 661]. This chain can be represented and analyzed like a graph with nodes and edges [331, 585, 136, 323].

Representation, detection and usage of the content semantics of comments. In the context of social media, the analysis of content semantics has become a hot topic, because it allows researchers to investigate phenomena more deeply than they could do with the structural analysis of networks alone.

One of the lines most closely related to this research field is semantic network analysis [251, 374, 662]. It examines the way in which two words are associated with each other within a set of texts. To this end, it constructs suitable networks whose nodes represent words and whose arcs denote ties between words. The networks thus constructed are investigated by means of the concepts and theories of classic network analysis. An approach using semantic analysis, in combination with Social Network Analysis, is presented in [251]. Here, the authors investigate online travel forums to predict tourism demand. In [374], the authors present an approach that uses semantic network analysis to study social media rumors in Twitter discourses during a specific event. A component of the approach of [374] uses semantic network analysis. In [662], the authors adopt semantic network analysis to investigate user experiences on mental disorders shared on Reddit. Specifically, they consider two subreddits, namely `/r/Bipolar` and `/r/Depression`. In the context of community detection approaches, the content and semantics of the underlying network are often analyzed. In [506], the authors propose a community detection approach us-

ing topological and content information. It exploits a network in which each node is associated with one or more attributes. In [521], the authors propose a community detection framework in ranking-based social networks. They aim to find overlapping communities in which members are interested in the same topic, with their relationships measured based on the rate of their viewpoints.

Topic oriented community detection is also investigated in [679]. Here, the authors combine social objects clustering and link analysis to define the semantics within a network. Their methodology is very similar to the one proposed in [521]. Frequent pattern mining is also applied in [9] to perform community detection.

Defining user spectra to classify user behaviors in cryptocurrencies. In the context of cryptocurrencies, malicious users have found new opportunities for profit by deceiving newcomers [620] thanks also to the fact that blockchains guarantee a certain degree of anonymity [520, 564]. Many researchers have proposed approaches to detect frauds, scams and, generally, illegal transactions on several cryptocurrencies, such as Bitcoin and Ethereum [119, 152, 62, 383].

Other researchers have focused on tracking accounts and people, or groups of people, who performed these illegal acts [395, 368, 61]. This last challenging issue has paved the way to the more general problem of classifying and characterizing accounts, addresses and smart contracts in a blockchain [384, 375].

As for this topic, the authors of [328] propose to characterize an entity in the Bitcoin blockchain by analyzing information revealed by the patterns of the transactions made by its neighbors. The approach of [328] models the Bitcoin blockchain as a directed weighted bipartite graph. The authors of [612, 402] propose a multi-class service identification of Bitcoin addresses based on a summarization of transaction history. The authors of [402] start from the approach proposed in [612] but add two more parameters to support classification. The authors of [691] present a new approach to decrease the anonymity of Bitcoin through entity characterization based on a cascade of Machine Learning models. The authors of [518] propose an approach focused on the detection of entities belonging to a single class, i.e., Exchange. First, they model the Bitcoin blockchain as a directed hypergraph. Then, they use this hypergraph to build classification models capable of detecting a set of discriminating features.

Finally, in [312, 606, 645], the authors propose two different methods that perform classification and clustering of addresses in a blockchain starting from the behavior of the corresponding users. In particular, the authors of [606] propose a Deep Learning based classification method called PeerClassifier. Instead, those of [312]

propose a clustering method that uses the Dynamic Time Warping similarity measure applied to two sequences represented as two univariate time series.

Extracting information from posts on COVID-19. Social media have already played a key role during emergencies [353]. Indeed, a social medium can easily spread a message to a huge number of users; therefore, it could be useful for emergency response managers to know what is happening in real time. This has led researchers to analyze the content shared by users during past pandemic outbreaks [164, 530, 557, 443, 591]. As for post analysis, in [598], the authors analyze different characteristics of the content posted on Reddit in order to determine whether this platform benefits from a freedom from the press or not.

Since December 2019, when the first cases of COVID-19 were reported in Wuhan (China), the conversations about it have increased in Twitter and Facebook, as well as in other social network platforms. In [146], the authors started collecting tweets, from January 28^{th}, 2020, continuously monitoring Twitter's trending topics, keywords, and sources associated with COVID-19, to capture conversations related to the outbreak. Social networks are also leveraged to convey misinformation, myths and other low quality news. In [578], the authors analyze 5 types of myths emerged during the crisis: flu comparison, heat kill disease, home remedies, theories about the origin of COVID-19 and vaccine development.

Along with Twitter and Facebook, Reddit is a social platform where users discuss historical phenomena too. Indeed, the past literature provides us with several studies in which researchers analyze the behavior of users through their posts and comments during different events [598, 382]. The ultimate goal is the derivation of interesting patterns that could deeply describe the whole scenario, or that could be leveraged to perform a prediction activity [549, 25, 379, 186, 480, 514, 410, 433]. Researchers analyzed Reddit during these months of COVID-19 pandemic. To the best of our knowledge, at the time of writing this thesis, few studies consider Reddit along with other social networks [172, 63], and even fewer ones focus only on the Reddit perspective [672, 275, 453].

Extracting time patterns from the lifespans on TikTok challenges. Similar to what happened in the past with other social media, such as Instagram, Facebook [498], Twitter [192, 98], Yelp and Reddit [130, 180, 131], TikTok has recently attracted the interest of researchers from different fields [590]. For instance, it has been the subject of investigation by researchers working in the context of marketing [166], Social Network Analysis, Machine Learning and Deep Learning, health and politics [688, 396, 151, 553, 413, 582], just to cite a few of these fields.

Being considerably popular among teenagers [301], TikTok has led to the emergence of new types of influencers [343]. Many people have become influencers in this social medium without even planning to do so.

Another aspect of TikTok that has caught the attention of researchers concerns the recommendation algorithm [194, 647, 575]. In fact, when a user scrolls through her home page, TikTok suggests her some videos to watch.

Some authors have focused on developing and/or applying Machine Learning and Deep Learning algorithms to understand the dynamics of TikTok. For example, the authors of [652] developed an algorithm to predict the effects of influencer advertising on product sales. The authors of [692] analyze how TikTok challenges encourage the principle of imitation.

Several authors in the past have been interested in Internet challenges that have led to harmful behavior, especially among young people [303]. Like all other social platforms, TikTok also has positive and negative effects. For example, in [668] the authors analyze the role of TikTok in stimulating science memes. A report on the role of TikTok as a widely used source of information on popular culture, as well as on other issues, and even news, can be found in [465].

Alongside these examples of positive behavior, researchers have also studied instances of negative behavior. One such examples involves the Blackout challenge [604]. In [529], the authors analyze the participation of adolescents in TikTok challenges, as well as the potential impact that the latter exert on them. In [348], the authors mention the Cinnamon challenge, which requires participants to ingest spoonfuls of ground cinnamon powder without any liquid. Other challenges have prompted adolescents to commit crimes, such as stealing something at school and posting an incriminating video online [427]. The authors of [692] study the role of TikTok challenges in fostering the imitation principle. In this analysis, they use the concept of memes and introduce the notion of "imitation publics". The author of [356] focuses on the strategies that can be adopted to create a video for a challenge; to this end, he analyzes the `#distantdance` challenge in detail. The authors of [155] study the processes through which challenges can influence TikTok users. Finally, the authors of [592] analyze how TikTok challenges can be exploited to spread specific messages in this social medium.

Investigating community evolutions in TikTok. One of the most common methodologies for the detection and investigation of communities in Social Network Analysis involves the adoption of pattern mining algorithms [9, 448, 646]. In [448], frequent pattern mining is used for community detection in social networks. Here, the authors propose a method based on the operations that might be performed by a

user, such as following another user or suggesting a content. The pattern mining process is applied on the database of user actions. Another approach for community detection based on frequent pattern mining is presented in [9]. Here, the authors propose a method to model a dataset of entities as a social network.

In addition to what we have seen in the previous subsection, the dynamics of TikTok has been further analyzed in the past, even if the number of studies is still small. Some authors have focused on using Machine Learning and Deep Learning approaches to understand the dynamics of TikTok. For example, the authors of [652] propose a Machine Learning-based approach to predict the effects of influencer advertising in TikTok sales. Specifically, this approach uses Convolutional Neural Networks to determine where products should be placed in a video. They also introduce the concept of motion-score to quantify the benefit of placing a product in a certain part of a video.

To the best of our knowledge, community evolution in TikTok has not been analyzed in the past. In this thesis, we aim at filling this gap.

1.3.2 Networking things

We have seen that the IoT paradigm is spreading in a massive way in these last years. Therefore, minimizing human intervention for the installation and management of devices in IoT contexts has been one of the core issues in this research scenario. This leads to the necessity of finding smarter and smarter autonomous decision-making processes, so that devices are able to vary their configuration dynamically throughout their working duration, selecting the best protocol to use, the best routes and the best nodes to communicate with [41, 505, 313, 21]. In the context of IoT, the typical security goals of confidentiality, integrity and availability introduce additional problems. Indeed, the classical countermeasures to face privacy and security threats have to be rethought taking into account the many restrictions and limitations, in terms of components and devices, computational and power resources, and even the heterogeneous and distributed nature of the IoT [419, 680, 673, 5, 346, 23, 475, 360, 416].

Even Social Network Analysis has been employed in this context. Indeed, many researchers have started to employ the results already obtained in this research field to address issues concerning IoT, such as anomaly detection and information extraction after disasters [118, 120, 6, 353, 621, 318, 32]. For example, the MIoT environment represents the extension to smart objects and the IoTs of social internetworking scenarios [53]. Indeed, users joining multiple social networks can be assimilated to objects belonging to different IoTs, although the data type and nature, and the kind of issues to be addressed, are rather different.

Networking wearable devices for fall detection in a workplace. Only in the United States, thousands of deaths and disabilities occur every year because of occupational accidents [169]. Given these statistics, researchers have devoted much effort to study safety at work [459].

In addition to theoretical studies of safety at work, some researchers have focused on the practical implementation of the rules and regulations using Sentient Multimedia Models and Systems [121, 40]

As in many other aspects of everyday life, Machine Learning has started to play an important role in safety at work [432, 611]. For instance, in [432], the authors propose a methodology based on Machine Learning techniques, like Classification Trees (CTs), Support Vector Machines (SVMs), Extreme Learning Machines (ELMs) and Bayesian Networks (BNs) for the analysis of the causes and types of workplace accidents. The aim of this research is the construction of an expert system with the ultimate goal of providing a tool that facilitates the elaboration of a workplace accident prevention policy.

An important context is fall detection. There are three different types of techniques developed for fall detection, namely ambient sensor based, vision based, and wearable device based [450].

Ambient sensor based techniques exploit the recordings of audio and video and/or monitor vibrational data from the environment [601, 690, 133, 629, 138, 31, 524]. For instance, the approach described in [601] analyzes and verifies sensor-transmitted events through audio and video streams for object detection and tracking. It detects falls through an event sensing function and a continuous tracking of the approximate location of the user.

The second category of fall detection approaches comprises those based on vision [440, 469, 184, 201]. For instance, the approach described in [440] uses an artificial vision algorithm to detect the person with a camera and to study changes in human actions. A Machine Learning algorithm classifies the current state of the user.

The last category of fall detection approaches relies on wearable devices [489, 534, 322, 335, 674, 97, 35, 378, 373, 431, 602]. These devices are made up of different kinds of sensor. For instance, the approach described in [332] uses different wireless tags placed on some parts of the body to detect the posture of a user. Acceleration thresholds, along with velocity profiles, are applied to detect falls.

Other fall detection approaches are based on Machine Learning [483, 533]. For instance, the approach of [533] monitors tri-axial accelerometer data in three different sliding time windows, each one lasting one second.

Anomaly detection and classification in Multiple IoT scenarios. Anomaly detection has been largely investigated in past literature. Here, anomalies have been defined in very different ways, based on the reference domain and data model. A widely accepted definition of anomaly is the one proposed by Hawkins in [298], where an anomaly is defined as "an observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism". Other definitions have been proposed in the past too [83, 137].

Anomaly detection is an issue largely investigated in past literature. Recently, several surveys have proposed structured and comprehensive overviews of anomalies to cope with the need of providing usable taxonomies (see, for instance, [137]). Some applications are reported in [15, 399, 538, 365].

In [20, 15], the authors investigate anomalies in graph-based environments. Specific analyses of this topic can be found in [34] for social networks, in [266, 273, 330, 289] for intrusion detection, in [560] for traffic modelling, and in [351, 350] for gene regulation. In [19], the authors show that both near-stars and near-cliques are indicators of anomalous behaviors in networks. They focus on anomaly detection in weighted graphs. In [157], the authors propose an approach to anomaly detection in dynamic networks. This exploits the analysis of sub-structures, such as maximal cliques, for detecting community-based anomalies, i.e., unexpected variations of communities. In this work, a community coincides with a maximal clique.

As said before, recently, some authors have started to study scenarios in which several social networks interact with each other to allow their users to achieve certain goals [103]. In past literature, different terms have been used to refer to this context, including multilayer social networks [83], cross platform online social networks [558], multi social networks [428], and Social Internetworking Scenarios [103]. This is a highly investigated field, since the number of users who simultaneously interact with multiple social networks is constantly growing. For instance, in [83], new forms of anomalies emerging in multi-layer social networks are investigated. Several recent approaches on anomaly detection exploit classification through Machine Learning-based and/or neural network-based engines [486, 27, 89, 619, 468, 267, 460].

Increasing protection and autonomy of smart objects in the IoT. In the context of IoT security, trust architecture design and reputation evaluation play a crucial role, enhancing object security and reliable data collection and management. However, as stated above, due to its peculiarities (i.e., large number of entities with limited computation ability, coupled with the highly dynamic nature of the network), existing solutions for sensor or P2P networks strive to be directly applicable to the IoT [569, 148, 337, 105, 149, 499, 198, 262, 608]. In particular, some works leverage

cryptographic primitives or authentication mechanisms, such as TinySec [337], Key Session Scheme [149], SPINS [499], INSENS [198], and SERP [262]. However, they are computational demanding. Moreover, they are not secure against internal malicious nodes having the valid cryptographic keys. On the other hand, some of the nodes may have hardware fault (i.e., radio/sensor), and using only cryptographic mechanisms does not guarantee that these nodes are excluded from the network. Hence, a behavior-based or experience-based trust management framework could be more suitable.

In [148], the authors propose a scheme in which, using cryptographic primitives, each node has a unique and trustworthy identity. In [153], the authors present a trust architecture, called IoTrust, with a cross-layer authorization protocol. An agent-based trust model for a WSN is presented in [150]. It adopts a watchdog scheme to observe the behavior of nodes and broadcast their trust ratings. In [587] a reputation-based scheme called DRBTS is proposed.

In this context, blockchain technology brings a lot of advantages, such as managing device configuration, storing sensor data, enabling micro-payments and, above all, enhancing and securing IoT functionalities [170, 84, 206, 502, 400, 314, 539, 540, 565]. Among the above cited approaches, [502, 400] leverage Blockchain technology to provide forms of trust in an IoT network. In particular, the authors of [502] propose an approach for bridging trust between secure domains by leveraging Blockchain technology.

A focal point is that, although Blockchain technology provides decentralized security and privacy, it involves significant energy, delay, and computational overhead, not suitable for most resource-constrained IoT devices. So a lightweight instantiation of a Blockchain, suited for the IoT, is needed [206, 190].

Extending saliency maps and gaze prediction in an Industry 4.0 scenario. In [91], the authors say that saliency "intuitively characterizes some parts of a scene - which could be objects or regions - that appear to an observer to stand out relative to their neighboring parts".

Saliency map and visual scanpath (also called gaze path) have attracted a lot of interest from researchers in recent years [406, 522, 296, 552, 650, 589, 319, 588, 390, 574, 408, 488, 372, 280, 397, 562].

As for traditional approaches applied on natural images, the authors of [296] propose the generation of saliency maps through a graph-based methodology using Markov chains. The authors of [552] introduce a bottom-up framework for both static and space-time saliency detection. The authors of [408] obtain remarkable results using a recurring Convolutional Neural Network (hereafter, CNN), which

extracts features and takes the spatial Long Short-Term Memory (hereafter, LSTM [306]) into account. The authors of [372] introduce two new models employing a unique architecture but different feature spaces.

For web pages, the authors of [319] introduce a model to predict both the locations of the most attended information and the corresponding attention sequence on a web page. The authors of [588] extend a web page saliency model by including the history of the previous interactions. In [390], the authors propose a framework to predict visual attention on web pages through the extraction of multi-features and a Machine Learning algorithm. Unlike natural images, websites have been rarely investigated in the past. One of the first approaches generating saliency maps for websites is reported in [563]. It proposes the usage of multiple kernel learning to integrate several feature maps. In [562], the authors present a framework that combines low-level features and high-level representations from deep neural networks of images.

During the last years, a class of Deep Learning architectures, i.e., GANs, has achieved outstanding results in the saliency prediction for both natural images and web pages. For instance, the authors of [488] propose a GAN architecture, called SalGAN.

Apart from saliency maps, visual scanpaths have obtained the interest of researchers as well. Indeed, the past literature provides several approaches to perform this task. They are based on traditional techniques [182, 628, 406, 326, 274, 228] or Deep Learning [158, 43, 324, 623, 573]. Also in this case, traditional approaches can be classified according to the context in which they are applied; again, contexts could be natural images [634, 406, 182] or websites [326, 274, 228].

Besides traditional approaches, a lot of Deep Learning-based techniques have been employed to evaluate visual scanpaths [158, 43, 324, 623, 573]. All of them have been developed for natural images. Instead, to the best of our knowledge, no approaches for websites have been proposed in past literature.

1.4 Contributions

1.4.1 Networking people

In summary, the main contributions of Networking people are the following:

- The definition of the notion of *k-bridge*, i.e., users playing an important role in opinion transmission and user influence among different communities.

- An analysis framework to show that Reddit is assortative with respect to common centrality measures in Social Network Analysis and a set of stereotypes for the authors of posts and comments in Reddit.
- A framework for the detection of backbones of information diffusers in a social network, alongside a new centrality measure to capture this phenomenon.
- A semantic and a structural analysis of NSFW contents in Reddit, their authors, and the communities built around this kind of posts and comments.
- A multi-dimensional social network-based model for Yelp, supporting the study of negative reviews and negative influencers, and exploited to define a set of stereotypes of users in Yelp.
- A set of types of users describing different behaviors during a cryptocurrency speculative bubble, and an analysis of backbones performed in the network model defined for this study.
- The definition of a framework to extract content semantics from a set of comments; the framework is based on frequent patters and a data structure called CS-Net.
- A method for user behaviors classification in blockchain based on multivariate time series called "spectra".
- New approaches to extract information from posts on COVID-19, along with a hierarchical classification algorithm, an algorithm capable of identifying sets of homogeneous themes, and an algorithm capable of identifying communities based on shared interests.
- A method to classify TikTok challenges in dangerous or non-dangerous ones based on their lifespan.
- A method to classify TikTok challenges in dangerous or non-dangerous ones based on the community evolution they produce in the social network.

In the last subsections, we will examine these contributions in more detail.

Defining and detecting k-bridges. First of all we introduce a new notion, that we call *k-bridge*. A k-bridge is a user who connects k sub-networks of a network or k networks of a multi-network scenario. k-bridges are particular users capable of playing an important role in opinion transmission, user influence, etc. Indeed, they allow a person or a business in a community to be known in another one. This may have important applications in the dissemination of information, in the search for influencers, and in marketing, for example when a business, leader in one category, wants to expand in another related category. In this thesis, we present and formalize the notion of k-bridge and we show that it has interesting properties, such as the anti-monotone one. Then, we propose a k-bridge detection algorithm that exploits

these properties. Afterwards, we extract several knowledge patterns about k-bridges. We carried out our work on Yelp.

Another contribution is the investigations of k-bridges and their characteristics on two additional networks, i.e., Reddit and the network of patent inventors derived from PATSTAT-ICRIOS, a repository storing metadata of patents submitted in many countries (see below). The ultimate goal was to verify if the results we found in Yelp were generally valid for k-bridges.

As a last contribution, we present two possible use cases that could benefit from the knowledge and the exploitation of k-bridges. The former regards the engagement of k-bridges in Yelp to find the best targets of a market campaign, whereas the latter concerns the analysis of k-bridges' activities to infer new products/services in order to expand and improve the revenues of existing businesses.

Detecting user stereotypes and their assortativity. The main contribution is to demonstrate that Reddit is assortative with respect to centralities, based on the characterization of users through Social Network Analysis. This confirms the hypotheses of Newman regarding the existence of assortative mixing in social networks and provides insights into how users on Reddit form communities and interact with others who share similar characteristics or interests.

The significance and value of our contribution concern both the theoretical and the application viewpoints. From the theoretical point of view, this is the first study on the concept of stereotype in Reddit. Thanks to this, it contributes to this research area by providing new and valuable information about how Reddit users can be characterized and identified based on specific traits. Actually, approaches for the characterization and identification of *specific* traits of users have been independently presented in different scientific works: users showing multi-community engagement [603], anti-social behaviors [191], community opposers [370], "answer-persons" [112], and "explorers" [302] are some examples. However, all of them considered only a specific trait of users. This thesis represents also the first research effort to analyze the concept of assortativity in Reddit. By showing that Reddit is assortative with respect to centralities, it confirms the previous hypotheses about the existence of assortative mixing in social networks, which can provide insights into how people form communities and interact with others who share similar characteristics or interests.

Instead, as far as the application point of view is concerned, we highlight that the knowledge patterns on stereotypes and author assortativity can be employed in a large variety of contexts. Just to cite a few of them, we mention: *(i)* the definition of some guidelines to follow in order to make a subreddit successful; *(ii)* the definition

and realization of different categories of recommender systems for Reddit; *(iii)* the definition of an algorithm that finds subreddits to merge or, at least, to integrate; *(iv)* the detection of possible targets for an advertising campaign; *(v)* the definition and implementation of different categories of recommender systems; *(vi)* the definition of an algorithm that builds blacklists of users based on author stereotypes. These practical applications demonstrate the significance of the research described in this thesis, whose results can be used to support decision makers and strategists in a variety of fields, such as marketing, information diffusion, consensus analysis, and so on.

Detecting backbones of information diffusers among different communities of a social platform. The main contribution to the detection of backbones of information diffusers is the definition of a new centrality measure, which we tested on an analysis framework thought to capture the characteristics of information diffusers. This centrality measure is thought to capture those kind of users easier then the classic centrality measures. More specifically, our framework first uses a combination of three classic centrality measures well known in Social Network Analysis, namely degree, closeness and betweenness centralities. Then, it introduces a new centrality measure specifically designed to better identify the possible existence of backbones of information diffusers. We call this new centrality measure *disseminator centrality*. Finally, starting from the new disseminator centrality, we demonstrate how it is possible to identify the possible backbones of disseminator bridges.

To validate all the technical aspects introduced, we describe our experimental campaign through which we show that disseminator centrality is more effective in identifying disseminator bridges than classical centralities or a combination of them.

Investigating NSFW contents and their authors. We contribute to the research on NSFW contents and authors through two different kinds of analyses. The first one is semantic, while the second is structural.

In the semantic analysis, we show how it is possible to clean a dataset of comments and how it can be annotated in order to enrich, clean and reorganize all the posts and comments. In particular, we present a method that associate lexical and sentiment annotation to them. Then, we show how frequent patterns of words can be extracted from all these posts and comments. All patterns are enriched with a set of features. This leads to one of the main contributions of the semantic analysis, i.e., the definition of several utility measures to capture the characteristics of interest of those patterns. The last contribution is a network-based analysis that allows us to identify and study communities of users exchanging NSFW contents.

In the structural analysis, we provide a contribution by investigating the phenomenon of NSFW posts in Reddit and describing the whole context (authors, subreddits and readers) behind it. For this purpose, we consider a dataset that includes all the posts published in Reddit from January 1^{st}, 2019 to December 31^{st}, 2019. These analyses allowed us to extract three findings regarding NSFW posts, NSFW authors and NSFW subreddits, respectively. In particular, the main contributions of this second kind of analysis are: *(i)* the discovery that traditional approaches to sentiment computation do not work well in the case of NSFW posts and comments; *(ii)* the definition, and next detection, of opinion leaders in real communities sharing NSFW adult content; *(iii)* the discovery of text patterns representing the seeds or building blocks of NSFW posts and comments on Reddit; *(iv)* the possibility of determining new virtual communities of users sharing NSFW adult content; *(v)* the discovery of new virtual opinion leaders, who can guide these new communities.

Investigating negative reviews and negative influencers. The main contribution of this work is the definition of a multi-dimensional social network-based model for Yelp, which is then used to study negative reviews and negative influencers. Then, thanks to the concepts and techniques of Social Network Analysis, we exploit the multi-dimensional model to define three stereotypes of Yelp's users. Another main contribution of this work is the definition of *(i)* bridges, *(ii)* double-life users and *(iii)* power users. These stereotypes can help the detection of the negative influencers in Yelp and the definition of a profile for them. These last are completed by a Negative Reviewer Network, which allows us to investigate the main characteristics of the negative influencers in Yelp.

Investigating user behavior in a blockchain during a cryptocurrency speculative bubble. The analysis of user behavior during a cryptocurrency speculative bubble led us to the definition of three categories of users, namely:

- *The power addresses*, i.e., the most active users on Ethereum, who were responsible for most of the transactions of this network. More specifically, we consider the power addresses for each of the periods of interest (i.e., the pre-bubble, bubble and post-bubble).
- *The Survivors*, i.e., those users who were power addresses in all the three periods of interest.
- *The Missings*, i.e., those users who were power addresses in the pre-bubble period and stopped being power addresses in the bubble and post-bubble periods.
- *The Entrants*, i.e., those users who were not power addresses in the pre-bubble period and became power addresses in the bubble and post-bubble periods.

The first main contribution of this work is the employment of Social Network Analysis based techniques to identify the main characteristics that distinguish the corresponding users from the others. Those users are mapped as a network. Then, an analysis on backbones is performed, in order to determine when similar users tend to link with each other in the network built from the data. The last contribution is a prediction method to understand the main actors that will be protagonists of the next period of the bubble.

Representation, detection and usage of the content semantics of comments. The main contribution of this work is the definition of a framework to extract content semantics from a set of comments. The experiments are carried out on a Reddit dataset, but the method proposed is general and can be employed in other social platforms. We show how the comments can be cleaned from errors, inconsistencies and bot-generated content. Then, we annotate each comment with its sentiment.

A second contribution regards the possibility of filtering the comments based on different utility functions. In this work, we show the results obtained by performing filtering in the sentiments of comments, comment rate and Pearson's correlation. Then, we define CS-Net, a data structure able to represent the comments chosen. The CS-Net model is extensible so that, if we want to consider further content semantics perspectives in the future, it will be sufficient to add another type of arcs for each new perspective.

The last contribution concerns the definition of an approach to evaluate the semantic similarity of two CS-Nets. In particular, our approach privileges the most extended component because it represents a greater portion of the content semantics than the other. Analogously to the CS-Net model, our approach can be easily extended in case we want to add further content semantics perspectives.

We believe that the approach and data structure proposed in this work allow us to extract the "fil rouge" connecting a set of comments. We mentioned above that if these were the comments published by a single user, we could employ the extracted knowledge to reconstruct her profile. However, this is not the only possible application of our approach. In fact, the comments under consideration could also be those written by more users on a single community, or a set of comments on a certain topic (e.g., COVID-19) or a set of comments written during a certain time period (e.g., during the Tokyo Olympics).

Depending on the set of comments, which it operates on, our approach has several applications. These may concern, for example, the construction of content-based or collaborative filtering recommender systems, the construction of new user com-

munities, the identification of outliers or the construction of new thematic forums (e.g., subreddits in Reddit) from the existing ones.

Defining user spectra to classify user behaviors in cryptocurrencies. As for this topic, we aim at filling a gap on user classification in a blockchain by proposing an automatic approach for classifying users in Ethereum. First of all, we define a method to represent the behavior of an address in the blockchain. Starting from a network built from transactions on the blockchain, we define the concept of "spectrum", which represents the variation of the values of some features during a period of time. Each of these features represents the behavior of the user in the network.

Based on this first result, we define a classification method based on the spectrum of an address. To the best of our knowledge, there is no out-of-the-box classification algorithm with these characteristics. Thus, it is necessary to define a new one. The algorithm is based on a modified version of the Eros distance, which is able to capture the difference between multivariate time series.

Finally, we propose an automatic multi-class algorithm (instead of the single-class existing ones) for classifying Ethereum users based on their past behavior.

Extracting information from posts on COVID-19. As for this topic, we propose three new approaches to extract information from posts on COVID-19. These approaches are defined on Reddit data. The main contributions of these three approaches are:

- A hierarchical classification algorithm for posts on COVID-19. This helps to categorize the posts based on their main discussion themes.
- An algorithm capable of identifying a set of homogeneous themes regarding COVID-19. This helps to find all the topics discussed by users in Reddit.
- An algorithm capable of identifying user communities based on shared interests. This helps to find communities of discussion inside the social network.

The three approaches proposed have been conceived with reference to COVID-19. However, we point out that they are general and can be used to extract information about any other issue that may cause an intense posting activity on Reddit.

Extracting time patterns from the lifespans on TikTok challenges. As far as this topic is concerned, we provide a contribution to address the problem of classification of TikTok challenges in dangerous and non-dangerous ones. In particular, we analyze their lifespans to extract time patterns that allow the classification of challenges into dangerous and non-dangerous ones. By the term "lifespan" we do not mean

the time interval between the moment a challenge is launched and the one it disappears permanently. In fact, there are challenges that never disappear even though they have not been active for a long time. From our point of view, the lifespan of a challenge is the period that elapses from the time it is launched to the time it is no longer capable of eliciting at least limited interactions with users. As will be clear in the following, the classification approach we are proposing is currently able to support the detection of dangerous challenges only near the end of their lifespan, or at least after a presumably long time period. On the other hand, the early detection of dangerous challenges is not our objective. In fact, we want to propose a *challenge classification* approach that, once has its validity verified, represents *a first step* in the direction of early detection of dangerous challenges. To reach the latter goal, in the future, we can think of greatly reducing the granularity of the time intervals taken into account (which is currently coarse) in such a way as to identify the time patterns allowing the detection of the dangerous challenges at an early stage.

Investigating community evolutions in TikTok. In this case, we study the characteristics of the communities participating in dangerous and non-dangerous challenges, the behavior of the corresponding users and their dynamics and evolution over time. The final goal is the possible detection of evolutionary patterns allowing the distinction of non-dangerous challenges from dangerous ones.

Regarding this fact, it must be said that TikTok has been intensively studied in the literature from multiple perspectives, especially with regards to influencers [343, 622], and their role in marketing [159, 635, 292, 559], politics [553, 413, 582], health [688, 396, 151, 321], etc. Many other studies have focused on the recommendation algorithm underlying TikTok [194, 647, 575, 681, 357, 55], privacy and security issues [466, 349, 438, 678], types of messages and contents that, directly or indirectly, are spread through this social platform [55, 637]. In the literature, there are also some studies about challenges [692, 592], the principle of imitation at their base [356] and the strategies with which the videos launching them are designed [155]. Our contribution goes exactly in that direction, showing that challenges can be divided in different intervals of their lifespan. We demonstrate that dangerous challenges have different lifespan intervals of non-dangerous challenges.

1.4.2 Networking things

In summary, the main contributions of Networking things are the following:

- A framework for safety in the workplace that consists of three distinct levels: personal devices, area devices, and a safety coordination platform.

- A model that organizes smart objects in communities, which is one of the main parts of the framework. This community organization is used to increase protection and autonomy of smart objects in the IoT.
- An extension of saliency maps and gaze prediction in an Industry 4.0 scenario using GAN-based approaches specifically designed for websites.

In the next subsections, we will examine these contributions in more detail.

Networking wearable devices for fall detection in a workplace. As for this topic, we contribute in the contexts of Sentient Multimedia Systems and Machine Learning by providing a framework for safety in the workplace. This framework consists of three distinct levels, namely: *(i) Personal Devices*, which are smart objects worn by workers (e.g., safety glasses, protective gloves, etc.); *(ii) Area Devices*, which are fixed smart objects associated with a specific area (e.g., access control gates, devices for controlling environmental parameters); *(iii) Safety Coordination Platform*, which monitors the safety of the working environment and, if necessary, activates the appropriate alarms and provides the related advices.

The design of the framework proposed is done at an abstraction level that allows it to be used in any working context and to address any safety issue. However, in order to give a very concrete idea of how it could operate in a real context, we also illustrate its specialization to a particular scenario, very studied in past literature, which is fall detection.

In fact, some of the main causes of injuries and deaths in the workplace are slips, trips and falls. Our framework adopts a new, very advanced wearable device, based on Machine Learning, which we designed, built and tested. Instead, it employs existing smart objects for Area Devices. Finally, it adopts an appropriate chain of Machine Learning based modules for the management of the Safety Coordination Platform.

Anomaly detection and classification in Multiple IoT scenarios. As far as this topic is concerned, we propose a new methodological framework for anomaly detection and classification in MIoTs. Our framework models anomalies and the corresponding issues in a MIoT by providing a multi-dimensional view, based on three orthogonal taxonomies: *(i)* presence anomalies vs success anomalies; *(ii)* hard anomalies vs soft anomalies; and *(iii)* contact anomalies vs content anomalies. Each combination of the possible values of these dimensions gives rise to a specific type of anomaly to investigate, for instance the *Presence-Hard-Contact* anomalies. Furthermore, anomaly definitions are orthogonal to specific anomaly detection approaches, past or future, which may be applied (and will be combined) in the context of our framework.

Together with the multi-dimensional taxonomy, another main contribution of our framework is the extension of conventional methodological frameworks to the MIoT case. In this scenario, our framework has been conceived to address two problems, known as the "forward problem" and the "inverse problem", respectively. In the forward problem, we aim to analyze the effects that multiple anomalies have onto the MIoT. On the other hand, in the inverse problem, which is traditionally more complex, we aim at detecting the source of the anomalies (i.e., the objects that have generated them) based on the effects that these have on the objects or their connections.

In order to show the possible usage of our framework, we present a case study centered around a smart city. Furthermore, in order to evaluate our framework and extract knowledge, we have conducted a series of tests. These allowed us to find several important knowledge patterns about anomalies and their effects in a MIoT. Our most important findings may be summarized as follows: *(i)* the effects of the anomalies of a node rapidly decrease as the distance from the node itself increases; *(ii)* anomalies are less evident in a MIoT than in a single IoT; *(iii)* the number of anomalous nodes increases as the number of IoTs increases, in a roughly linear way; *(iv)* the outdegree of anomalous nodes has a great impact on the spread of the anomaly over the MIoT; *(v)* closeness centrality is even more important than degree centrality in the spread of anomalies; *(vi)* the computation time necessary for the detection of anomalous nodes is polynomial against the number of MIoT nodes; *(vii)* the time necessary for evaluating the effects of anomalies in a MIoT is quadratic against the number of its nodes.

Increasing protection and autonomy of smart objects in the IoT. As for this topic, we provide a contribution in the context of protection and autonomy of smart objects in the IoT. Indeed, we propose a two-tier blockchain framework to increase them. The first contribution is a model that organizes smart objects in communities. This community organization is one of the main parts of our framework, which is made of two tiers, a local one and global one. Thanks to this idea, we propose a lightweight blockchain to manage the protection of smart devices, as they have low computational capacities. The local tier is the one that manages trust measures of each smart object inside the communities, while the global one is used to record aggregated data related to the individual communities, as well as the trust value that each community assigns to the other ones.

Another important contribution is the protection of smart objects from different communities interacting with each other. Indeed, our framework manages the trust of objects between different communities, alongside the one of the same community.

Extending saliency maps and gaze prediction in an Industry 4.0 scenario. As far as this topic is concerned, we propose some GAN-based approaches to predict the saliency map and the gaze path of a user accessing a web page. The approaches we propose here are variants of GAN-based approaches presented in the past literature and are specifically designed to work on websites. We defined three variants of SalGAN, for saliency map prediction, and two variants of PathGAN, for gaze path prediction. As we will see below, the best variant of SalGAN and the two variants of PathGAN are fine-tuned. In addition, they present several other refinements taking into account various observations we made during some experiments conducted "on the field". As we will see below, at the end of all these activities, we managed to achieve: *(i)* a SalGAN-based approach for website saliency map prediction that has a better performance than existing approaches carrying out the same task; *(ii)* two PathGAN-based approaches that, to the best of our knowledge, are the first ones proposed in the literature for gaze path prediction on websites.

Furthermore, we present a new dataset supporting training, testing and evaluation of approaches for predicting saliency maps and gaze paths of users accessing websites. Finally, we illustrate a tool supporting a web page designer to organize the graphical layout of the page in order to increase the visitor interest and curiosity.

1.5 Outline of the thesis

This thesis aims to explore how it is possible to model and study relationships between people, things and both. In order to do this, it is organized in two main parts.

Part I, called *Networking people*, explores all the models and approaches defined to uniformly represent and study connections between people, mainly on Online Social Networks. The studies in this part cover several Social Platforms that people use everyday, from Reddit to Yelp, from Twitter to TikTok. But we also study people connections in blockchains, in order to investigate their behavior in this increasingly widespread world, especially when it comes to cryptocurrencies. In particular, in Chapter 2, we focus on Yelp, where we define a new type of users, namely k-bridges, along with an approach to detect them. In Chapter 3, we investigate user behaviors in Reddit and show that users are assortative in this social platform. Afterwards, in Chapter 4, we study backbones of information diffusers and how to find this kind of users by means of a customized centrality measure. In Chapter 5, we investigate Not Safe For Work contents and their authors; these analyses are made in Reddit. In Chapter 6, we investigate the negative reviews on Yelp and define an approach to identify negative influencers and to evaluate their impact on their neighbors. In Chapter 7, we propose an approach to study and classify user

behaviors in the Ethereum blockchain during a cryptocurrency speculative bubble. We also introduce, in Chapter 8, a new framework to represent, detect and study the usage of content semantics in Online Social Networks. In Chapter 9, we focus on blockchain to study user behavior through a new representation of it, called "spectrum". In Chapter 10, we propose an approach to extract information from posts on COVID-19; we also study the results obtained from this analysis. In Chapters 11 and 12, we focus on TikTok, analyzing the evolution of challenges through the investigation of their lifespan or the communities that arise from them.

Part II, called *Networking things*, delves into the analysis on how to represent connections among (smart) objects in real contexts, mainly IoT ones, through (Social) Networks. In particular, in Chapter 13, we propose a framework to model a workplace characterized by wearable devices and area devices with the aim of detecting slips, trips and falls. In Chapter 14, we focus on the definition of a new model to represent Multiple IoT networks and introduce a framework for the analysis of anomalies in this kind of network. In Chapter 15, we illustrate a new framework to increase protection and autonomy of smart objects in an IoT scenario; for this purpose, it exploits a network-based model and a lightweight blockchain. Finally, in Chapter 16, we propose an extension of two existing GAN models for saliency maps and gaze prediction in an Industry 4.0 scenario.

Part III, called "Closing remarks", presents some conclusions and future works on the methods and approaches presented in this thesis. In particular, in Chapter 17, we draw some conclusions on our approaches in networking people and things. In Chapter 18, we present some possible future works on networking people and things. A last section in this chapter is dedicated to future developments on Internet of Everything (IoE).

Part I

Networking people

In this part, we apply Social Network Analysis concepts, parameters and approaches to people joining several Social Platforms and Blockchains. In particular, we investigate: (i) the concept of k-bridge in Chapter 2; (ii) user stereotypes and their assortativity in Chapter 3; (iii) information diffusers among different communities of a social platform in Chapter 4; (iv) the NSFW phenomenon in Chapter 5; (v) negative reviews and negative influencers in Chapter 6; (vi) user behavior in a blockchain during a cryptocurrency speculative bubble in Chapter 7; (vii) content semantics of comment in Chapter 8; (viii) user behaviors in cryptocurrencies in Chapter 9; (ix) posts on COVID-19 in Chapter 10; (x) the lifespan of TikTok challenges in Chapter 11; (xi) community evolution in TikTok in Chapter 12.

2

Defining and detecting k-bridges

In this chapter, we introduce the concept of k-bridge (i.e., a user who connects k subnetworks of the same network or k networks of a multi-network scenario) and propose an algorithm for extracting k-bridges from a social network. Then, we analyze the specialization of this concept and algorithm in Yelp and extract several knowledge patterns about Yelp k-bridges. In particular, we investigate how some basic characteristics of Yelp k-bridges vary against k (i.e., against the number of macro-categories to which the businesses reviewed by them belong). Then, we verify if there exists an influence exerted by k-bridges on their friends and/or on their co-reviewers. Furthemore, we analyze the relationship between k-bridges and power users. In addition, we investigate the relationship between k-bridges and the main centrality measures in the macro-categories of Yelp. We also propose two further specializations of k-bridges, regarding Reddit and the network of patent inventors, to prove that the knowledge on k-bridges we initially found in Yelp is not limited to this social network. Finally, we present two use cases that can highly benefit from the knowledge on k-bridges detected through our approach.

The material presented in this chapter was derived from [221].

2.1 Methods

2.1.1 A model for k-bridges and an approach to extract them

In this section, firstly we propose a general model for k-bridges, and specialize it to several social networks and, then, we present an algorithm to extract k-bridges.

Defining and modeling k-bridges

Let $\mathcal{N}$ be a social network and let $\mathcal{CS}$ be the set of the communities of $\mathcal{N}$ of our interest:

$$\mathcal{CS} = \{\mathcal{C}_1, \mathcal{C}_2, \cdots, \mathcal{C}_M\}$$

Given the community $\mathcal{C}_i$, $1 \leq i \leq M$, it is possible to define the corresponding user network $\mathcal{U}_i = \langle N_i, A_i \rangle$. N_i is the set of nodes of $\mathcal{U}_i$; there is a node n_{i_p} for each user u_{i_p} belonging to $\mathcal{C}_i$. A_i is the set of arcs of $\mathcal{U}_i$; there is an arc $a_{pq} = (n_{i_p}, n_{i_q}) \in A_i$ if there exists a relationship between the users u_{i_p} and u_{i_q}, corresponding to n_{i_p} and n_{i_q}, respectively.

Finally, it is possible to define the overall user network $\mathcal{U} = \langle N, A \rangle$ corresponding to $\mathcal{N}$. There is a node $n_i \in N$ for each user of $\mathcal{N}$. There is an arc $a_{pq} = (n_p, n_q) \in A$ if there exists a relationship between the users u_p and u_q, corresponding to n_p and n_q, respectively.

Here, and in the previous definition, we do not specify the kind of relationship between users. As we will see in the following, it is possible to define a specialization of $\mathcal{U}$ for each relationship we want to investigate. For instance, $\mathcal{U}^f$ is the specialization of $\mathcal{U}$ when we consider *friendship* as the relationship between users.

After having introduced our model, we can present our definitions of *k-bridge*, *bridge*, *non-bridge*, *strong bridge and very strong bridge*.

Definition 2.1. A ***k-bridge*** is a user of $\mathcal{N}$ belonging to exactly k different communities of this social network, $1 \leq k \leq M$. □

Definition 2.2. A ***non-bridge*** is a k-bridge such that $k = 1$, i.e., a user belonging to exactly one community. □

Definition 2.3. A ***bridge*** is a k-bridge such that $k \geq 2$, i.e., a user who belongs to at least 2 different communities of $\mathcal{N}$. □

Definition 2.4. A ***strong bridge*** is a k-bridge such that $k \geq th_s$. Here, th_s is a threshold such that $2 \leq th_s < M$. □

Definition 2.5. A ***very strong bridge*** is a k-bridge such that $k \geq th_{vs}$. Here, th_{vs} is a threshold such that $th_s < th_{vs} \leq M$. □

Observe that the definition of k-bridge is anti-monotone. This means that if a user is a k-bridge then she is also a h-bridge $1 \leq h \leq k-1$.

Finally, given a k-bridge $u_p^k \in \mathcal{U}$, there are k nodes $n_{1_p}, n_{2_p}, \cdots, n_{k_p}$ associated with her, one for each community of $\mathcal{N}$ it belongs to. Each node represents a sort of "avatar" of u_p^k in the network corresponding to this community.

An algorithm for k-bridge extraction

An important consequence of the anti-monotone property of k-bridges mentioned above is the possibility of designing an optimized algorithm to extract them, borrowing some ideas from the well-known Apriori approach [10]. Indeed, the anti-monotone property allows us to state that the search space to find k-bridges is reduced to the set of identified (k-1)-bridges, which can be obtained, in turn, starting

from the set of identified (k-2)-bridges, and so forth. This observation strongly resembles the reasoning and the properties underlying the Apriori algorithm. In our case, due to the possible huge number of users who could be bridges, it is more convenient to revert the problem and extend our reasoning to communities. Indeed, according to the definition of bridges, we can derive a formal property for communities, as follows:

*Property 2.6 (*Anti-monotonicity of communities). *All the communities involved in the definition of k-bridges must also be involved in the definition of (k-1)-bridges.* □

Therefore, a possible algorithm to identify k-bridges from the communities of a social network consists of the following steps. First, for each community, the set of the corresponding users is retrieved. Intuitively, in order to be consistent with its general definition, a community must have a minimum number of users joining it. We call this measure `support` and we impose that a community must have a `support` greater than a threshold min_sup. The result of this step is a set of communities called L_1.

To obtain 2-bridges, we start from L_1 and compute a set of community pairs, called P_1, joining L_1 with itself. Each pair of communities in P_1 represents a possible case in which at least a user acts as a bridge between them. Therefore, for each pair of communities in P_1, we compute the intersection of their users, and impose, once again, that its cardinality is greater than min_sup. The resulting filtered set of community pairs is called L_2. Observe that, for each community pair in L_2, the intersection among the corresponding users is also an outcome of this iteration as it contains all 2-bridges.

To compute 3-bridges, the algorithm proceeds by joining L_2 with itself; in this way, it obtains a set of community triplets, called P_2. Each triplet in P_2 contains the communities candidate to be simultaneously joined by 3-bridges. Once again, for each triplet in P_2, we compute the intersection of users among the three communities and impose that its cardinality is greater than min_sup. The resulting set is called L_3. Also in this case, the set of 3-bridges, which is the outcome of this iteration, is implicitly obtained in the intersection computed above for each element of L_3.

In general, this procedure can be extended to compute k-bridges starting from the set L_{k-1} used to computed (k-1)-bridges. Algorithm 1 reports a pseudo-code of our approach for extracting k-bridges from a social network.

As a final remark, we observe that our solution can be easily extended to a big data strategy (which is a realistic requirement in the social network context) by leveraging the advances available for Apriori in the scientific literature, because our algorithm follows a strategy very near to the one adopted by Apriori. For instance, it is

Input

- D, a dataset of a Social Network
- $\mathcal{CS}$, the set of communities of D
- min_sup, a suitable threshold for minimum support

Output

- L_k, the set of k-communities linked by k-bridges
- B_k, the set of k-bridges

Require: L_t, a temporary set; $\texttt{getN}(\mathcal{C}_i)$ a function returning the set of users of the community $\mathcal{C}_i$

$L_1 = \{\mathcal{C}_i \mid \mathcal{C}_i \in \mathcal{CS} \wedge |\texttt{getN}(\mathcal{C}_i)| > th_s\}$ *//the set of communities in the dataset having support greater than min_sup*
$P = L_1 \bowtie L_1$ *// $\bowtie$ is the join operator*
$j = 2$ *//start with 2-bridges*
while $j \leq k$ **do**
  if $P \neq \emptyset$ **then**
    //for each tuple of the communities in P
    for $< (\mathcal{C}_1), (\mathcal{C}_2), \cdots, (\mathcal{C}_j) > \in P$ **do**
      $I = \texttt{getN}(\mathcal{C}_1) \cap \texttt{getN}(\mathcal{C}_2) \cap \cdots \cap \texttt{getN}(\mathcal{C}_j)$
      //if the minimum support is satisfied for this intersection
      if $|I| > min\_sup$ **then**
        Add $< \mathcal{C}_1, \mathcal{C}_2, \cdots, \mathcal{C}_j >$ to L_t
        //in the last iteration, store the found bridges and the involved
        //communities into the output parameters B_k and L_k, resp.
        if $j == k$ **then**
          Add I to B_k
          $L_k = L_t$
        end if
      end if
    end for
    $P = L_t \bowtie L_t$ *//re-compute P for the next iteration*
    $j++$, $L_t = \emptyset$
  end if
end while
return L_k, B_k

Algorithm 1: k-bridges Extraction Algorithm

possible to adapt our solution to work in a Map-Reduce based architecture following the studies described in [389, **?**].

Specializing our k-bridge model to Yelp

In Yelp, businesses are organized according to a taxonomy consisting of four levels. Level 0 comprises 22 macro-categories. Each macro-category has one or more child categories, so that level 1 comprises 1002 categories. A category may have zero, one or more sub-categories, so that level 2 consists of 532 sub-categories. Proceeding with this reasoning, the final level, i.e., level 3, has only 19 sub-sub-categories; indeed, most sub-categories are not further categorized.

When we specialize our model to Yelp, we have that this social network can be modeled as a set of 22 communities, one for each macro-category:

$$\mathcal{Y} = \{\mathcal{Y}_1, \mathcal{Y}_2, \cdots, \mathcal{Y}_{22}\}$$

Given the macro-category $\mathcal{Y}_i$, $1 \leq i \leq 22$, and the corresponding user network $\mathcal{U}_i = \langle N_i, A_i \rangle$, there is a node n_{i_p} for each user u_{i_p} who reviewed at least one business of $\mathcal{Y}_i$. Based on the relationship that we want to model, $\mathcal{U}$ can be specialized into $\mathcal{U}^f$, obtained when we consider friendship as the relationship between users, and $\mathcal{U}^{cr}$, obtained when co-review (i.e., reviewing the same business) is the relationship between users.

Given a k-bridge $u_p^k \in \mathcal{U}$, the k nodes $n_{1_p}, n_{2_p}, \cdots, n_{k_p}$ associated with her represent u_p in the k macro-categories where she performed at least one review.

Specializing our k-bridge model to Reddit

In Reddit, a user can participate to several subreddits. In this social network, the number of both users and subreddits is huge. So, in specializing our model to it, we consider only a subset of subreddits, for instance those about a certain topic or those published in a certain time interval. We can consider all the users who published at least one post in a subreddit as a community. So, we can model this scenario as:

$$\mathcal{R} = \{\mathcal{S}_1, \mathcal{S}_2, \cdots, \mathcal{S}_M\}$$

Given the subreddit $\mathcal{S}_i$, $1 \leq i \leq M$, and the corresponding user network $\mathcal{U}_i = \langle N_i, A_i \rangle$, there is a node n_{i_p} for each user u_{i_p} who submitted at least one post in $\mathcal{S}_i$. Based on the relationship that we want to model, $\mathcal{U}$ can be specialized into $\mathcal{U}^{cp}$, obtained when co-posting (i.e., contributing to the same subreddit) is the relationship between users.

Given a k-bridge $u_p^k \in \mathcal{U}$, the k nodes associated with her represent u_p in the k subreddits where she submitted at least one post.

Specializing our k-bridge model to the community of patent inventors (and/or applicants)

Patents are largely investigated in scientific literature because they provide a large amount of knowledge patterns on Research & Development sector [238, 203]. Patents can be grouped in several ways, for instance based on the country of their inventors and/or applicants or according to the International Patent Classification (IPC) class they belong to. According to this classification, they have associated a symbol of the form A01B 1/00. Here:

- The first letter denotes the "section" of the patent (for instance, A indicates "Human necessities").
- The following two digits denote its "class" (for instance, A01 indicates "Agriculture; forestry; animal husbandry; trapping; fishing").
- The next letter indicates the "subclass" (for instance, A01B represents "Soil working in agriculture or forestry; parts, details, or accessories of agricultural machines or implements, in general").
- The next one-to-three-digit number represents the "group".
- Finally, the other two digits denote the "main group" or "subgroup".

A patent examiner assigns classification symbols to each patent according to the above rule, at the most detailed level which is applicable to its content.

After having chosen a level of the IPC classification, for instance the "class" level, the set of patent inventors (or, alternatively, the set of patent applicants), taken from a world patent metadata repository, for example PATSTAT-ICRIOS, can be represented as:

$$\mathcal{I} = \{\mathcal{I}_1, \mathcal{I}_2, \cdots, \mathcal{I}_M\}$$

Given the IPC class i, the corresponding set of inventors $\mathcal{I}_i$ (i.e., the set of inventors who filed at least one patent belonging to this class), $1 \leq i \leq M$, and the corresponding user network $\mathcal{U}_i = \langle N_i, A_i \rangle$, there is a node n_{i_p} for each inventor u_{i_p} who filed at least one patent of the class $\mathcal{I}_i$. $\mathcal{U}$ can be specialized into $\mathcal{U}^{ci}$, obtained when co-inventing (i.e., filing the same patent) is the relationship between inventors.

After having defined a model for k-bridges and an approach to extract them, after having specialized it to Yelp, Reddit and the network of patent inventors, in the next section, we will focus on k-bridge properties. To help the reader understand the concepts of this chapter, in Table 2.1, we report the main notations introduced.

2.1.2 Investigating k-bridge properties

In this section, we analyze k-bridge properties. We carried out this task focusing on Yelp, which is our reference network. However, in the next paragraphs, we present

Notation	Semantics
$\mathcal{N}$	a generic social network
$\mathcal{C}_i$	the i^{th} community of $\mathcal{N}$
M	the maximum number of communities of $\mathcal{N}$
$\mathcal{U}_i$	the network representing the users of $\mathcal{C}_i$ and their relationships
N_i	the set of nodes of $\mathcal{U}_i$
A_i	the set of arcs of $\mathcal{U}_i$
u_i^p	the p^{th} user of the community $\mathcal{C}_i$
n_i^p	the node of $\mathcal{U}_i$ corresponding to u_i^p
$\mathcal{U}$	the overall user network corresponding to $\mathcal{N}$
n_i	a node of $\mathcal{U}$
$\mathcal{U}^r$	the specialization of $\mathcal{U}$ to the relationship r
th_s	the threshold for defining strong bridges
th_{vs}	the threshold for defining very strong bridges
$\mathcal{Y}_i$	the i^{th} community of Yelp
$\mathcal{S}_i$	the i^{th} subreddit of Reddit
$\mathcal{I}_i$	the set of inventors who filed at least one patent belonging to the i^{th} IPC class
$\mathcal{U}^f$	the specialization of $\mathcal{U}$ by taking the friendship relationship in Yelp
$\mathcal{U}^{cr}$	the specialization of $\mathcal{U}$ by taking the co-review relationship in Yelp
$\mathcal{U}^{cp}$	the specialization of $\mathcal{U}$ by taking the co-posting relationship in Reddit
$\mathcal{U}^{ci}$	the specialization of $\mathcal{U}$ by taking the co-inventory relationship in PATSTAT-ICRIOS
$\mathcal{M}$	the "macro-category" network of Yelp
$\mathcal{M}^{X\%}$	the subset of $\mathcal{M}$ whose macro-categories have been reviewed by at least X% of users

Table 2.1: The main notations used throughout this chapter

some experiments on Reddit and the network of patent inventors devoted to verifying if the results on k-bridges found in Yelp are general or specific for this social network.

Overview of Yelp dataset

The data required for the investigation activities was downloaded from the Yelp website at the address `https://www.yelp.com/dataset`.

In order to extract information of interest from this data, we needed a preliminary analysis. As a first insight, we found 10,289 businesses that belong to a category not referable to any of the macro-categories, and 482 businesses that belong to no category at all. Since the total number of businesses was 192,609, we considered these data as noise and so we discarded it.

After this task, we analyzed the distribution of the categories in the macro-categories. The result obtained is shown in Figure 2.1. From the analysis of this figure, we can observe that the "Restaurants" macro-category has a much larger number of categories than the other macro-categories.

Note that, in Yelp, a business can belong to more macro-categories. Therefore, as a preliminary step, it seemed us particularly interesting to analyze how many times two macro-categories appeared simultaneously in the same business. The total number of businesses with at least two macro-categories is 59,086. The top 20 pairs

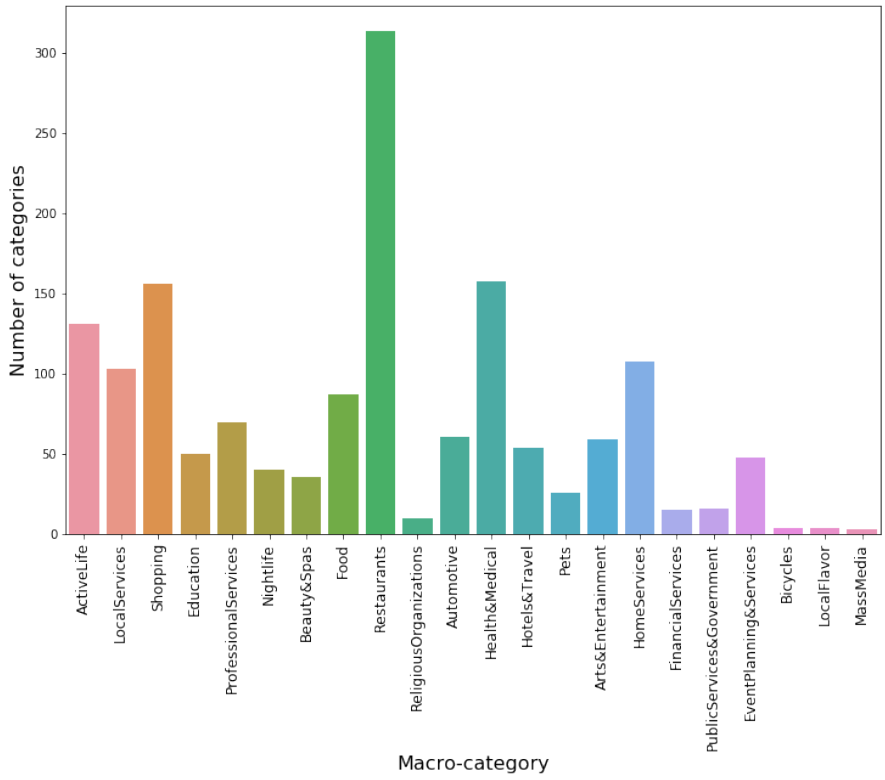


Fig. 2.1: Distribution of categories inside the macro-categories of Yelp

of macro-categories that appear several times together in one business of Yelp are shown in Table 2.2. As we can see from this table, there are two pairs of macro-categories (i.e., ⟨ "Restaurants", "Food" ⟩ and ⟨ "Restaurants", "Nightlife" ⟩) that appear together a much higher number of times than the other pairs.

Pair of macro-categories	**Count**		**Pair of macro-categories**	**Count**
Restaurants, Food	11094		Restaurants, EventPlanning&Services	1051
Restaurants, Nightlife	5566		HomeServices, ProfessionalServices	758
Health&Medical, Beauty&Spas	2544		Automotive, Food	736
Shopping, LocalServices	2315		Shopping, EventPlanning&Services	708
HomeServices, LocalServices	1998		Arts&Entertainment, Nightlife	589
Hotels&Travel, EventPlanning&Services	1964		LocalServices, ProfessionalServices	579
Shopping, HomeServices	1883		ActiveLife, Health&Medical	527
Shopping, Beauty&Spas	1711		ActiveLife, Shopping	484
Shopping, Food	1470		FinancialServices, HomeServices	445
Shopping, Health&Medical	1384		Shopping, Arts&Entertainment	434

Table 2.2: The top 20 pairs of macro-categories that appear simultaneously in one business of Yelp

After that, we considered the total number of Yelp users who made at least one review and we saw that it is equal to 1,637,138. The distribution of their reviews

is shown in Figure 2.2. We can observe that this distribution follows a power law. This result is perfectly in line with the ones of numerous studies about Online Social Networks and communities [444]. These studies highlight that the well-known social theory, according to which human activities usually follow a power law distribution, is still valid also in online communities. As a consequence, also in this kind of community, a few numbers of individuals (typically 10-20% of members) perform the majority of the activities (around 80-90% of the overall activities) [**?**]. Our experiment confirms that this trend also persists in the review tasks in Yelp.

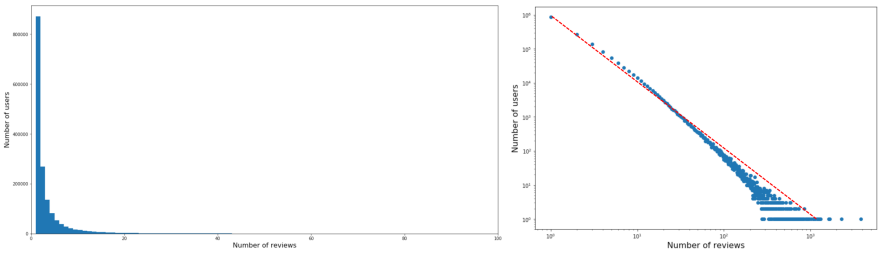


Fig. 2.2: Distribution of user reviews in Yelp - Linear scale (on the left) and Logarithmic scale (on the right)

The non-bridges are 530,411. All the other users are bridges. In order to start a deeper investigation of the k-bridge phenomenon, we computed the distribution of k-bridges against k. This is shown in Figure 2.3. An examination of this figure reveals that also this distribution follows a power law.

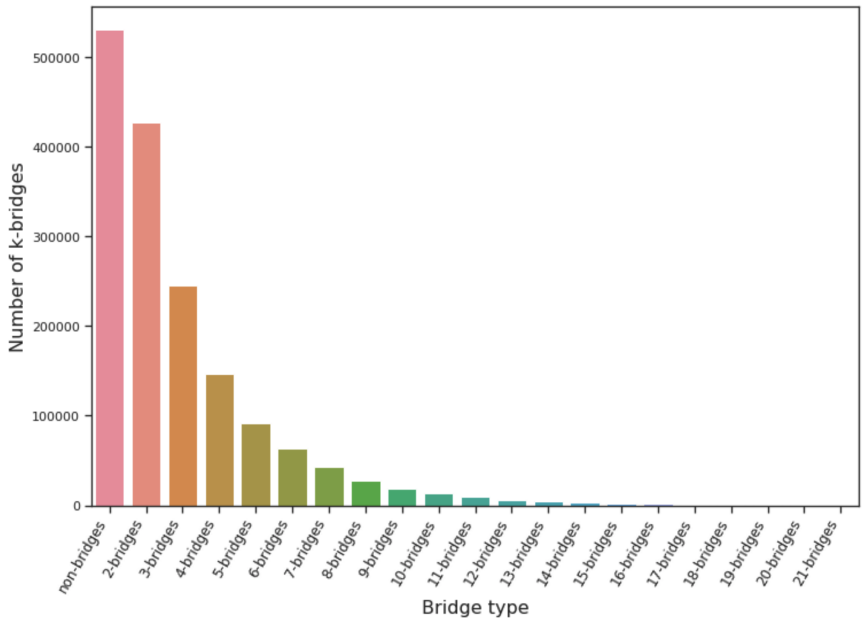


Fig. 2.3: Distribution of the k-bridges against k in Yelp

A last interesting, although partially expected, result that we found concerns the average number of reviews made by users. This is equal to 5.493 for bridges and 1.143 for non-bridges. This result confirms that a bridge tends to carry out more reviews than a non-bridge. It is also interesting to observe the corresponding standard deviations. In fact, the one for bridges is 17.69 whereas the one for non-bridges is 0.486. Such a high standard deviation for bridges confirms that this category of users is very varied, since it includes users who perform a huge number of reviews alongside users who perform few reviews. This is not the case, instead, for non-bridges, who always make few reviews.

k-bridges in the Yelp Friendship network

We began to verify the possible existence of a backbone among the bridges in $\mathcal{U}^f$. In order to have a connected network to study, we performed a pre-processing activity during which we eliminated the unconnected nodes from $\mathcal{U}^f$, corresponding to users who had no friendship relationship. The number of users having at least one friend (and, therefore, the number of network nodes) is 948,076. Specifically, 676,445 of these were bridges, while 271,631 were non-bridges.

After that, for each bridge (non-bridge), we measured the fraction of her friends who were bridges (non-bridges). The results obtained are shown in Table 2.3. From the analysis of this table, we can see that there are no significant differences in the fraction of bridges in the neighborhoods of bridges and non-bridges. The same applies to the fraction of friends of non-bridges. In light of this, we can conclude that there is no backbone among the bridges in $\mathcal{U}^f$.

	Fraction of friends that are bridges	**Fraction of friends that are non-bridges**
Bridges	0.9618	0.0382
Non-bridges	0.9633	0.0367

Table 2.3: Types of friends for bridges and non-bridges in $\mathcal{U}^f$

Then, we analyzed whether there was any form of correlation between being a bridge and having friends. For this purpose, we computed the fraction of bridges (non-bridges) having at least one friend and the fraction of bridges (non-bridges) having no friends. The result obtained is reported in Table 2.4. From the analysis of this table, we can see that bridges have a higher tendency to have friends than non-bridges. However, the extent of this phenomenon is not extremely evident.

At this point, we focused on investigating the possible influence that bridges exert on their neighborhoods. This investigation requires the usage of the strong and the very strong bridges. To detect them, it is necessary to specify the values of th_s

	Fraction of users with friends	Fraction of users without friends
Bridges	0.6113	0.3887
Non-bridges	0.5121	0.4879

Table 2.4: Fractions of users with and without friends in $\mathcal{U}^f$

and th_{vs} (see Section 2.1.1). To perform this task, we considered the distribution of the k-bridges against k in Yelp and we observed that it follows a very steep power law. As a consequence, according to the general trend of power law distributions, in particular of those showing a steep trend [**?**], it appeared us reasonable to choose th_s in such a way that only 10% of bridges are strong. Applying an analogous reasoning, we chose th_{vs} in such a way that only 10% of strong bridges are very strong. This way of proceeding led us to obtain that $th_s = 6$ and $th_{vs} = 12$.

After having determined the values of th_s and th_{vs}, we computed the fraction of strong and very strong bridges in the neighborhoods of bridges and non-bridges, respectively. The result is shown in Table 2.5. Differently from what emerges from Table 2.3, where there is a little difference between the *fraction of bridges* in the neighborhoods of bridges and non-bridges, in Table 2.5 it is evident that there is a big difference on the *strength of bridges* in the neighborhoods of bridges and non-bridges. In fact, the fraction of very strong bridges is more than double in the neighborhoods of bridges compared to the neighborhoods of non-bridges.

	Fraction of strong bridges	Fraction of very strong bridges
Bridge neighborhoods	0.41	0.12
Non-bridge neighborhoods	0.27	0.05

Table 2.5: Fraction of strong and very strong bridges present in the neighborhoods of bridges and non-bridges in $\mathcal{U}^f$

As a further verification of this trend, we computed:

- The ratio of the number of *non-bridges* in a bridge's neighborhood to the number of non-bridges in a non-bridge's neighborhood. This is equal to 2.50.
- The ratio of the number of *bridges* in a bridge's neighborhood to the number of bridges in a non-bridge's neighborhood. This is equal to 5.23.
- The ratio of the number of *strong bridges* in a bridge's neighborhood to the number of strong bridges in a non-bridge's neighborhood. This is equal to 7.27.
- The ratio of the number of *very strong bridges* in a bridge's neighborhood to the number of very strong bridges in a non-bridge's neighborhood. This is equal to 10.97.

This analysis fully confirms the fact that, in the neighborhoods of bridges, it is much more frequent to find strong or very strong bridges than in the neighborhoods of non-bridges.

As a final analysis on neighborhoods, we computed the distribution of bridges and non-bridges present in the neighborhood of a bridge and a non-bridge, respectively. These two distributions are illustrated in Figures 2.4 and 2.5. These figures show that both of them follow a power law distribution. Looking at the values of these distributions, we can observe that the difference between the values of non-bridges and weak bridges is not very evident. Instead, this difference becomes evident for strong and very strong bridges. This is a third confirmation of the trends seen previously.

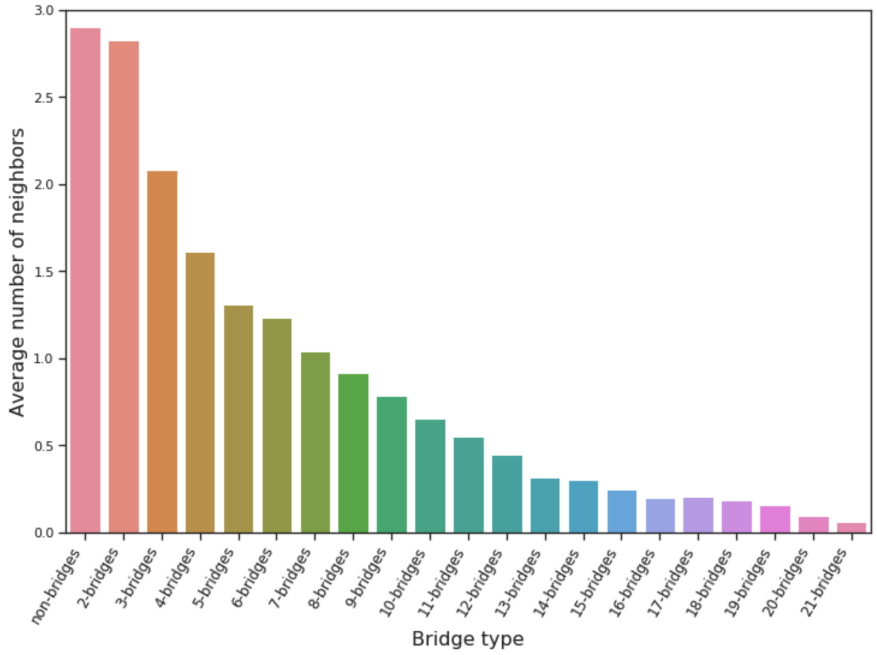


Fig. 2.4: Distribution of the neighbors of *bridges* in $\mathcal{U}^f$

k-bridges in the Yelp Co-review network

After the analysis done on the friendship network $\mathcal{U}^f$, we investigated the co-review network $\mathcal{U}^{cr}$. We started by verifying the existence of a backbone among the bridges in this network. Preliminarily, we removed those nodes corresponding to users who reviewed businesses not belonging to any macro-category of Yelp. As a consequence, the number of users (and, therefore, the number of nodes) who composed this network was equal to 1,634,547. Specifically, 1,037,484 of these were bridges while 597,063 were non-bridges.

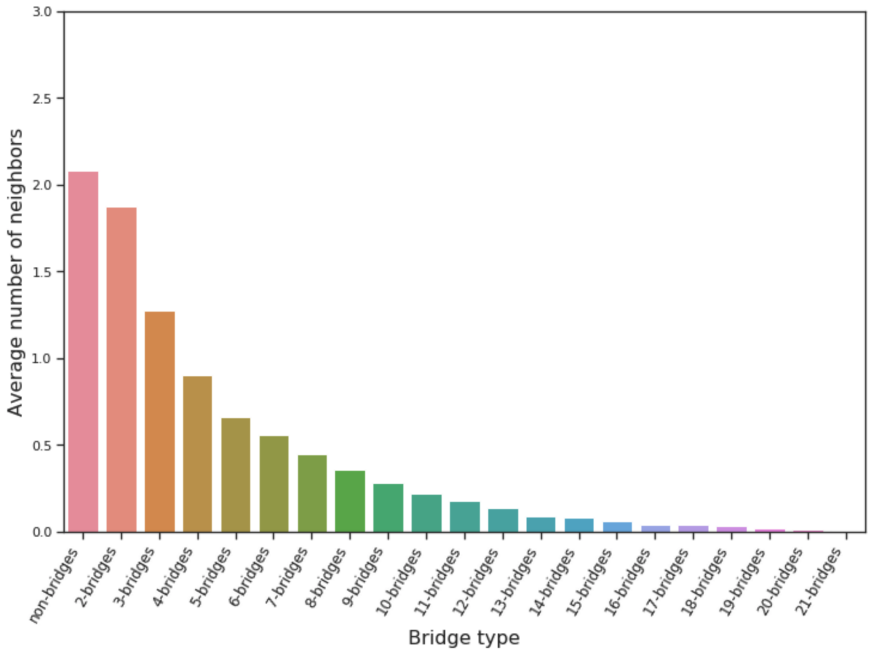


Fig. 2.5: Distribution of the neighbors of *non-bridges* in $\mathcal{U}^f$

The first analysis we made concerned the distribution of reviews with respect to users. The result obtained is shown in Figure 2.6. From the analysis of this figure, we can see that the distribution follows a power law. As a further analysis, we observe that $\mathcal{U}^{cr}$ is much denser than $\mathcal{U}^f$. In fact, the average degree of its nodes is equal to 1426.34, while, in $\mathcal{U}^f$, it is equal to 82.92.

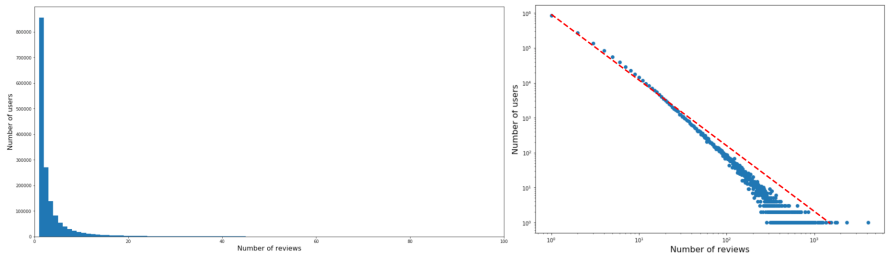


Fig. 2.6: Distribution of reviews for users in $\mathcal{U}^{cr}$ - Linear scale (on the left) and Logarithmic scale (on the right)

As a first analysis, we verified if there is a backbone among the bridges in $\mathcal{U}^{cr}$. Similarly to what we did for $\mathcal{U}^f$, for each bridge (non-bridge) we considered the fraction of co-reviewers that were bridges (non-bridges). The results obtained are shown in Table 2.6. From the analysis of this table we can see that there are significant differences in the percentage of co-reviewers that are bridges between a bridge and a non-bridge. The same applies to the percentage of co-reviewers that are non-bridges. In light of this, we can conclude that there is a backbone among the bridges in $\mathcal{U}^{cr}$.

	Fraction of co-reviewers that are bridges	**Fraction of co-reviewers that are non-bridges**
Bridges	0.9456	0.0543
Non-bridges	0.7451	0.2548

Table 2.6: Types of co-reviewers for bridges and non-bridges in $\mathcal{U}^{cr}$

As a further analysis of the neighborhoods of bridges and non-bridges in $\mathcal{U}^{cr}$, we computed the distribution of bridges and non-bridges present in the neighborhoods of bridges and non-bridges, respectively. These distributions are shown in Figures 2.7 and 2.8. These figures fully confirm the previous results about $\mathcal{U}^{cr}$. In fact, we can observe how the presence of bridges in the distribution of the neighbors of a bridge is very evident. The same happens for the presence of non-bridges in the distribution of the neighbors of non-bridges. These results represent a confirmation of the presence of a backbone among the bridges in the co-review network.

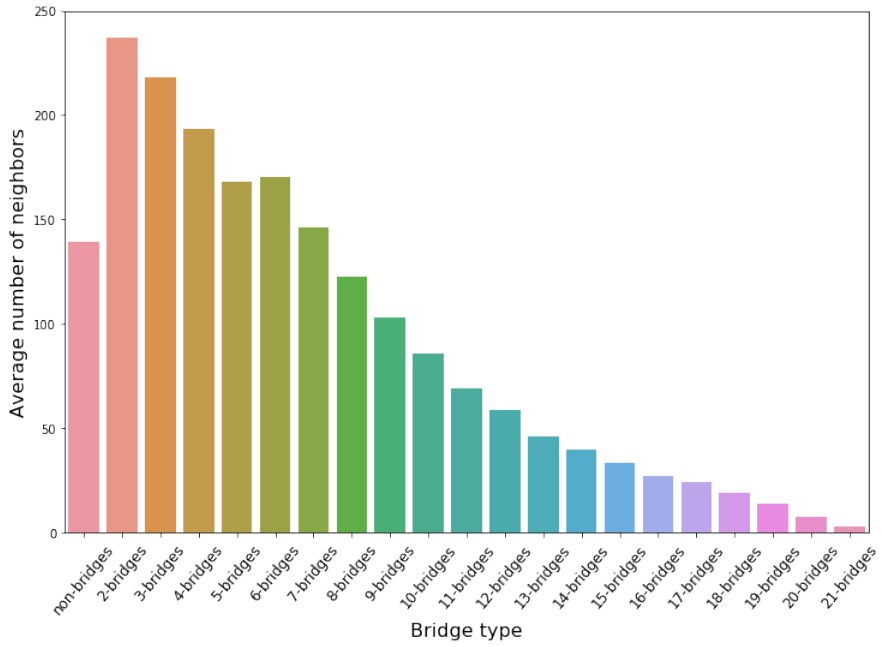


Fig. 2.7: Distribution of the neighbors of *bridges* in $\mathcal{U}^{cr}$

As a next analysis, we focused on the investigation of the possible influence that bridges can exert on their co-reviewers. For this objective, we computed the fraction of strong and very strong bridges present in the neighborhoods of bridges and non-bridges, respectively. The result is shown in Table 2.7. From the analysis of this table we can see that, differently from what happens in $\mathcal{U}^{f}$, in $\mathcal{U}^{cr}$ the fraction of strong and very strong bridges present in the neighborhoods of bridges is almost identical to the corresponding fraction relative to the neighborhoods of non-bridges. This

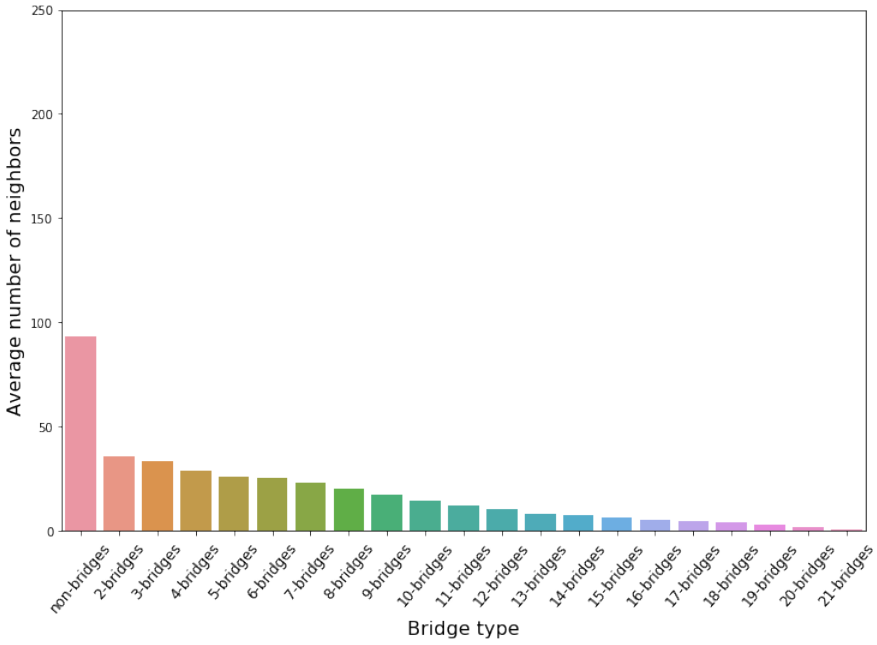


Fig. 2.8: Distribution of the neighbors of *non-bridges* in $\mathcal{U}^{cr}$

means that, while there exists a backbone linking bridges together, their evolution towards strong and very strong bridges does not depend on the support received by their neighbors.

	Fraction of strong bridges	**Fraction of very strong bridges**
Bridge neighborhoods	0.54	0.15
Non-bridge neighborhoods	0.57	0.18

Table 2.7: Fraction of strong and very strong bridges present in the neighborhoods of bridges and non-bridges in $\mathcal{U}^{cr}$

As a further verification of this trend we computed:

- The ratio of the number of *bridges* in the neighborhood of a bridge to the number of bridges in the neighborhood of a non-bridge. This is equal to 12.83.
- The ratio of the number of *strong bridges* in the neighborhood of a bridge to the number of strong bridges in the neighborhood of a non-bridge. This is equal to 12.19.
- The ratio of the number of *very strong bridges* in the neighborhood of a bridge to the number of very strong bridges in the neighborhood of a non-bridge. This is equal to 10.73.

This analysis fully confirms the previous one, i.e., the fact that there is no strong correlation between the strength of a bridge and being or not neighbor to another bridge in $\mathcal{U}^{cr}$.

The presence of a backbone among the bridges in $\mathcal{U}^{cr}$ and the absence of an analogous backbone among the bridges in $\mathcal{U}^{f}$ led us to consider $\mathcal{U}^{cr}$ more interesting than $\mathcal{U}^{f}$ for further analyses on k-bridges. Therefore, we decided to perform all the next investigations only on $\mathcal{U}^{cr}$.

Analysis of the possible correlation between k-bridges and power users in the co-review network

Firstly, we verified if there is a correlation between k-bridges and power users or, in other words, between k-bridges and degree centrality. To this end, we computed the distribution of the number of arcs for non-bridges, bridges, strong and very strong bridges. The results obtained are shown in Figure 2.9. As we can see from this figure, all distributions follow power laws; their corresponding coefficients α and δ are reported in Table 2.8. However, we observe that as k grows, the power law distributions move to the right and flatten out. It implies that, as k grows, the degree centrality of the corresponding k-bridges grows. This allows us to conclude that there is a correlation between the strength of k-bridges and degree centrality.

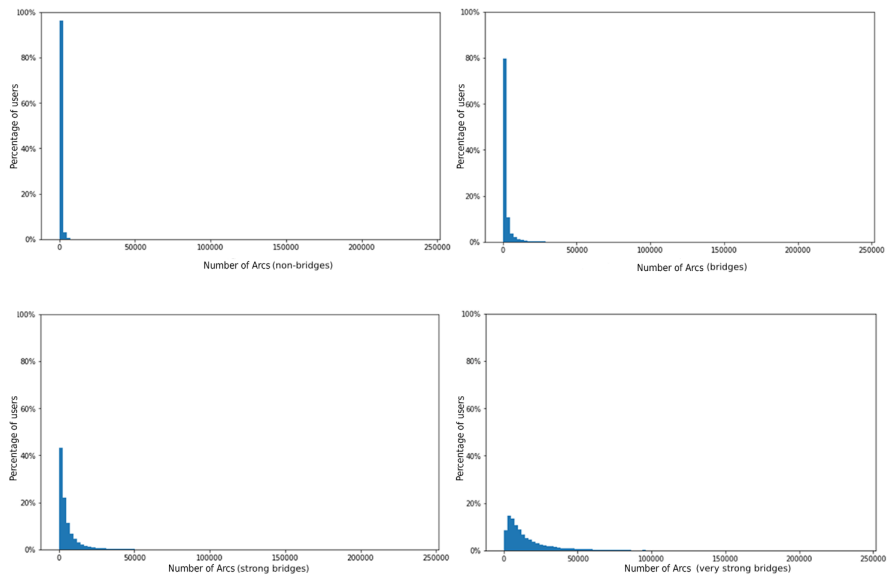


Fig. 2.9: Distributions of the number of arcs for non-bridges, bridges, strong and very strong bridges

As a second analysis, we selected the top 1% of power users (corresponding to the top 1% of the nodes of $\mathcal{U}^{cr}$ with the highest degree) and determined how these

	α	δ
Non-bridges	1.203	0.177
Bridges	1.403	0.066
strong bridges	1.290	0.077
Very strong bridges	1.322	0.113

Table 2.8: Coefficients α and δ for the power law distributions of Figure 2.9

were distributed between k-bridges (with k varying). We also repeated this analysis for the top 5%, the top 10%, the top 15%, the top 20% and, finally, for all users. The results obtained are shown in Figure 2.10. The analysis of this figure reveals that, as we select increasingly strong power users, the fraction of them that are strong bridges also increases, as the distribution moves to the right. This is a confirmation of the previous results regarding the existence of a correlation between k-bridges and power users.

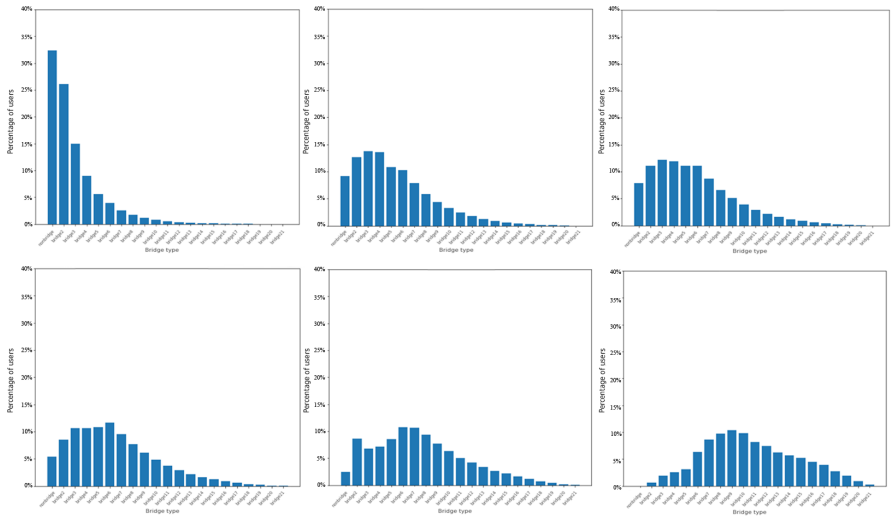


Fig. 2.10: Distributions of (power) users against the strength of bridges

As a final task, we repeated the previous analysis but we inverted k-bridges and power users. In particular, we selected the top 1% of k-bridges and determined the distribution of their degree. We repeated this analysis for the top 5%, the top 10%, the top 15%, the top 20% of k-bridges and, finally, for all users. The results obtained are shown in Figure 2.11. From the analysis of this figure, we can see that the distribution moves to the right. This implies that, as we select stronger and stronger bridges, the fraction of them with higher and higher degree increases too. This represents a third confirmation of the previous results and, ultimately, allows us to say that there is a strong correlation between k-bridges and power users.

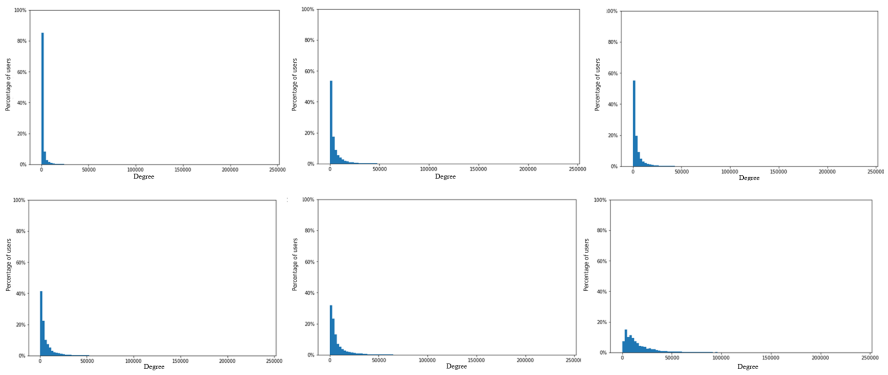


Fig. 2.11: Distributions of k-bridges against their degree

After having investigated the main properties of k-bridges, we focus on Yelp more deeply by analyzing the possible correlations between k-bridges and Yelp macro-categories.

2.2 Results

2.2.1 Analysis of k-bridges and macro-categories in Yelp

In this section, we aim at deepening our study of the correlations between k-bridges and Yelp macro-categories.

First of all, we considered the macro-categories which the reviews made by Yelp users refer to. The corresponding distribution is shown in Figure 2.12. From the analysis of this figure we can see that the "Restaurants" macro-category has a much higher number of reviews than all the other ones.

Once again, we are interested in investigating the co-review mechanism and the role of k-bridges as possible pioneers in this context. In order to carry out this study, we created a new network, which we call "macro-category network" and denote it with $\mathcal{M} = \langle N, E \rangle$. N represents the set of nodes of $\mathcal{M}$. In particular, there is a node $n_j \in N$ for each macro-category $\mathcal{Y}_j$ in Yelp. E is the set of edges of $\mathcal{M}$; in particular, there is an edge $e_{jh} \in E$ if both the macro-categories $\mathcal{Y}_j$ and $\mathcal{Y}_h$ have been reviewed by a fraction of users greater than or equal to a threshold $X\%$. Clearly, as X varies, we have different networks $\mathcal{M}^{X\%}$. Based on these definitions, we constructed the networks $\mathcal{M}^{1\%}$, $\mathcal{M}^{5\%}$, $\mathcal{M}^{10\%}$ and $\mathcal{M}^{15\%}$. These are shown in Figures 2.13 - 2.16.

The corresponding density and average clustering coefficient are reported in Table 2.9. Figures 2.17 and 2.18 present the variation of the values of the density and the average clustering coefficient when X increases. As shown in these figures, it is very likely to find two macro-categories that are co-reviewed by a small number of users. In fact, 98.1% of the possible combinations of categories are co-reviewed by

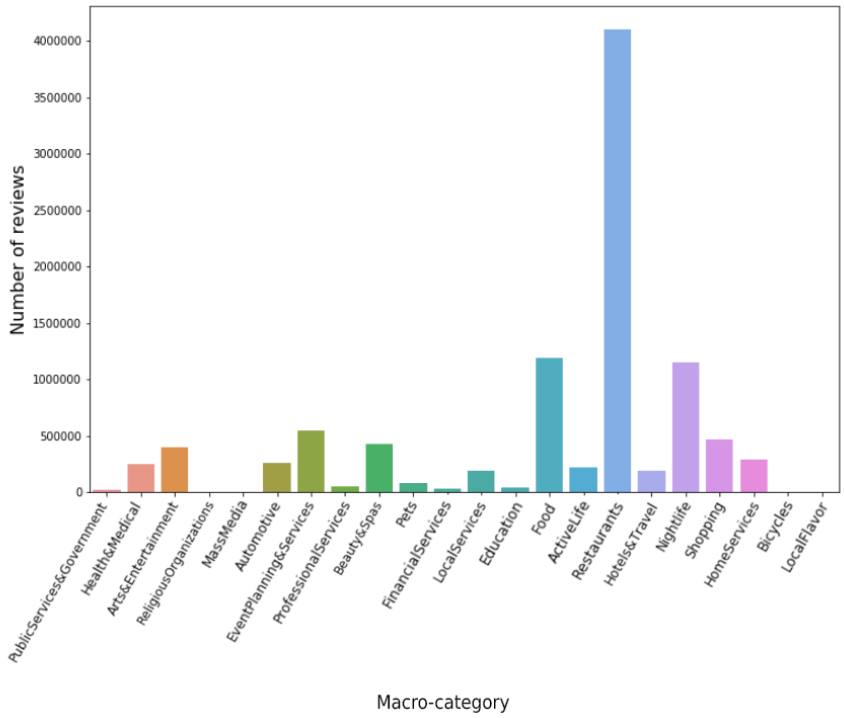


Fig. 2.12: Distribution of the reviews of Yelp users against the Yelp macro-categories

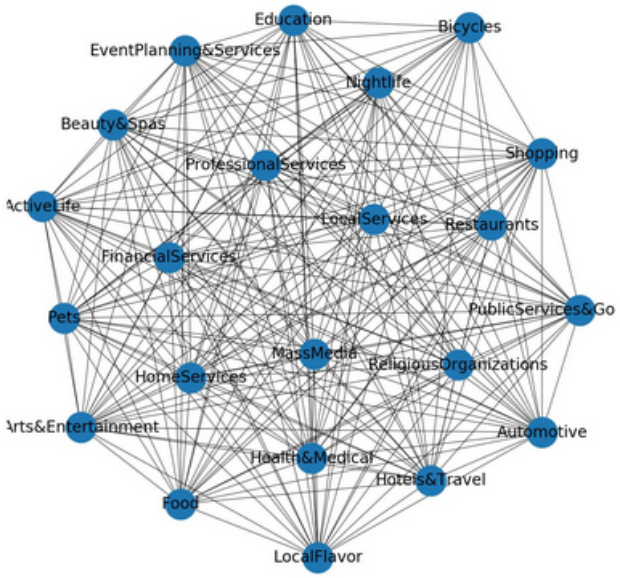


Fig. 2.13: The network $\mathcal{M}^{1\%}$

at least 1% of the users. However, if we are more demanding on the fraction of users that co-review the same macro-category, we can see from the figures that the trend of co-reviews varies rapidly. In fact, even if the possible combinations of co-reviewed

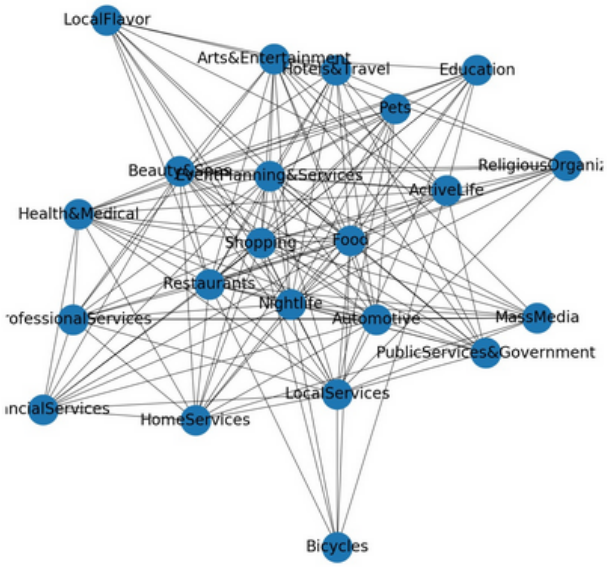


Fig. 2.14: The network $\mathcal{M}^{5\%}$

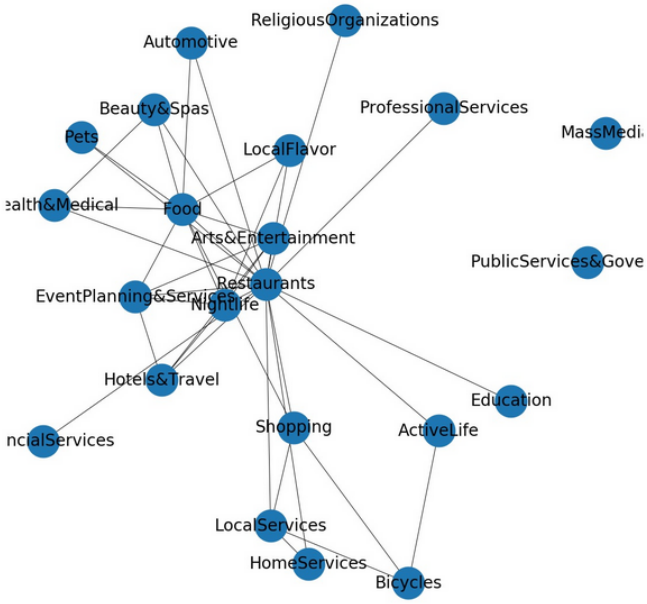


Fig. 2.15: The network $\mathcal{M}^{10\%}$

macro-categories is quite high with at least 5% of co-reviewing users, this number decreases rapidly when we further increase the value of X.

Table 2.10 shows the maximum and sub-maximum values of the degree centrality for the networks of Figures 2.13 - 2.16, along with the macro-categories which they refer to. The objective is to identify which macro-categories tend to

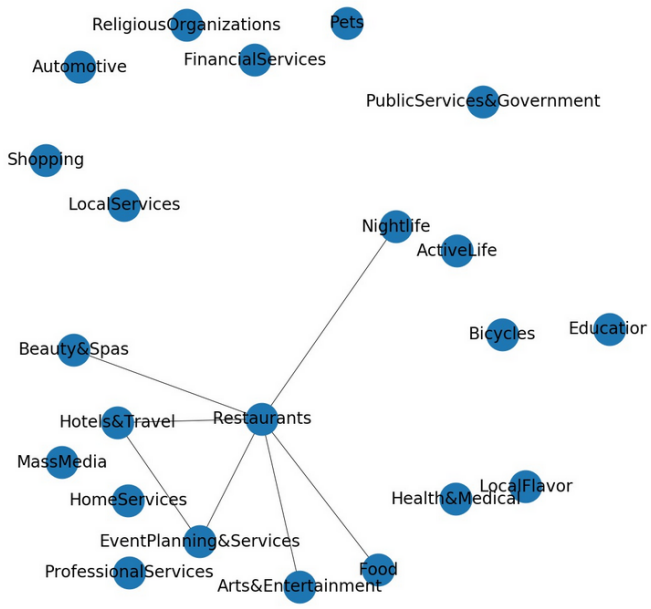


Fig. 2.16: The network $\mathcal{M}^{15\%}$

	$\mathcal{M}^{1\%}$	$\mathcal{M}^{5\%}$	$\mathcal{M}^{10\%}$	$\mathcal{M}^{15\%}$
Density	0.978	0.680	0.173	0.030
Average Clustering Coefficient	0.981	0.833	0.514	0.094

Table 2.9: Values of the density and the average clustering coefficient for the networks $\mathcal{M}^{1\%}$ - $\mathcal{M}^{15\%}$

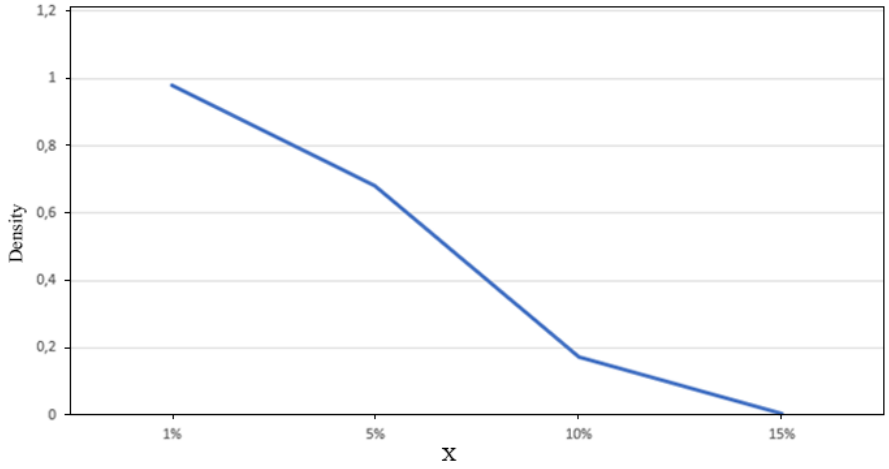


Fig. 2.17: Variation of the density of the macro-category networks $\mathcal{M}^{X\%}$ against the increase of X

have more co-reviews with other ones. From the analysis of this table we can observe that the two macro-categories most present with maximum or sub-maximum values are “Restaurants” and “Food”. Actually, this result was quite obvious, given

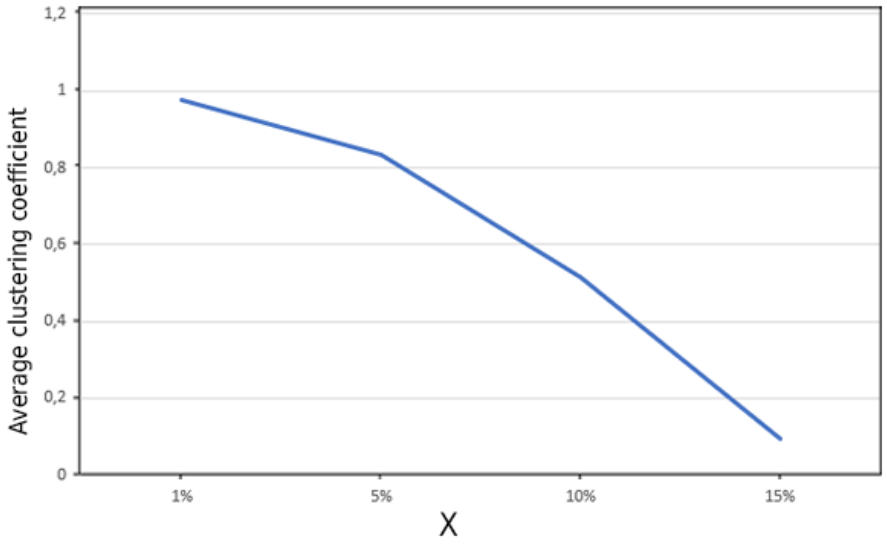


Fig. 2.18: Variation of the average clustering coefficient of the macro-category networks $\mathcal{M}^{X\%}$ against the increase of X

the distribution of the reviews in Yelp (see Figure 2.12). Instead, the fact that the macro-categories "Beauty&Spas" and "Hotels&Travel" are present as maximum or sub-maximum is particularly interesting. In fact, these two macro-categories have a much lower number of reviews not only than "Restaurants" and "Food" but also than several other macro-categories not present in Table 2.10.

	$\mathcal{M}^{1\%}$	$\mathcal{M}^{5\%}$	$\mathcal{M}^{10\%}$	$\mathcal{M}^{15\%}$
Maximum value and associated macro-category	1 (Beauty&Spas)	1 (Food)	0.857 (Restaurants)	0.286 (Restaurants)
Sub-maximum value and associated macro-category	1 (Food)	1 (Nightlife)	0.476 (Food)	0.095 (Hotels&Travel)

Table 2.10: Maximum and sub-maximum values of degree centrality and the corresponding macro-categories in the networks $\mathcal{M}^{1\%}$ - $\mathcal{M}^{15\%}$

Table 2.11 shows the maximum and sub-maximum values of the closeness centrality for the networks of Figures 2.13 - 2.16. We do not present this table for the semantics of closeness centrality in this application context. Instead, we want to highlight that, unlikely what generally happens in Social Network Analysis, where the nodes having the highest degree centrality and the highest closeness centrality are generally different [613], the macro-categories that have the highest values of closeness centrality are exactly the same as the ones having the highest values of degree centrality.

	$\mathcal{M}^{1\%}$	$\mathcal{M}^{5\%}$	$\mathcal{M}^{10\%}$	$\mathcal{M}^{15\%}$
Maximum value and associated macro-category	1 (Beauty&Spas)	1 (Food)	0.86 (Restaurants)	0.286 (Restaurants)
Sub-maximum value and associated macro-category	1 (Food)	1 (Nightlife)	0.614 (Food)	0.171 (Hotels&Travel)

Table 2.11: Maximum and sub-maximum values of closeness centrality and the corresponding macro-categories in the networks $\mathcal{M}^{1\%}$ - $\mathcal{M}^{15\%}$

Table 2.12 shows the maximum and sub-maximum values of the betweenness centrality for the networks of Figures 2.13 - 2.16. As we can notice, in $\mathcal{M}^{1\%}$ all the values of the betweenness centrality are very low. This is not surprising because this network is almost totally connected. The maximum and sub-maximum values of the betweenness centrality grow, albeit slightly, in $\mathcal{M}^{5\%}$. Once again, this is understandable because, if we look at Figure 2.14, we can see that this network is still very connected. The most interesting situation for this kind of centrality happens in $\mathcal{M}^{10\%}$. In fact, in this case, we have that the maximum and sub-maximum values of betweenness centrality are high. These values are associated with "Restaurants" and "Food". Now, looking at Figure 2.14, we can see how "Restaurants" and "Food" are actually two nodes from which we must pass to go from a node located in the top sub-net to a node located in the bottom one. Finally, as far as the betweenness centrality is concerned, the network $\mathcal{M}^{15\%}$ is not very significant, since it is almost completely disconnected.

	$\mathcal{M}^{1\%}$	$\mathcal{M}^{5\%}$	$\mathcal{M}^{10\%}$	$\mathcal{M}^{15\%}$
Maximum value and associated macro-category	0.001 (Arts&Entertainment)	0.049 (Food)	0.627 (Restaurants)	0.067 (Restaurants)
Sub-maximum value and associated macro-category	0.001 (LocalServices)	0.049 (Nightlife)	0.614 (Food)	0 (Beauty&Spas)

Table 2.12: Maximum and sub-maximum values of betweenness centrality and the corresponding macro-categories in the networks $\mathcal{M}^{1\%}$ - $\mathcal{M}^{15\%}$

Table 2.13 shows the maximum and sub-maximum values of the eigenvector centrality for the networks of Figures 2.13 - 2.16. We can observe that the maximum and sub-maximum values correspond to those of the degree centrality and the closeness centrality. Once again the two macro-categories with the highest values are "Restaurants" and "Food".

The analysis of the distributions and the ones of all the different forms of centrality show that "Restaurants" is an extremely dominant macro-category. Therefore,

	$\mathcal{M}^{1\%}$	$\mathcal{M}^{5\%}$	$\mathcal{M}^{10\%}$	$\mathcal{M}^{15\%}$
Maximum value and associated macro-category	0.217 (Arts&Entertainment)	0.279 (Food)	0.525 (Restaurants)	0.665 (Restaurants)
Sub-maximum value and associated macro-category	0.217 (LocalServices)	0.279 (Nightlife)	0.397 (Food)	0.395 (Hotels&Travel)

Table 2.13: Maximum and sub-maximum values of eigenvector centrality and the corresponding macro-categories in the networks $\mathcal{M}^{1\%}$ - $\mathcal{M}^{15\%}$

it is interesting to verify whether or not most of the properties we have previously found depend exclusively on "Restaurants".

To perform this verification, we removed all references to the macro-category "Restaurants" from the reviews. Then, we computed again the number of k-bridges and the distribution of users. In particular, the number of k-bridges decreased from 1,106,727 to 813,146, while the number of non-bridges increased from 530,411 to 823,992.

The distribution of users is shown in Figure 2.19. From the analysis of this figure, we can observe that, in this case, the distribution follows a much steeper power law. This is understandable because those nodes that were previously non-bridges continue to be so now. At the same time, all the nodes that were previously 2-bridges and that referred to "Restaurants" become non-bridges. More in general, all nodes that where k-bridges ($k \geq 2$) and referred to "Restaurants" become $(k-1)$-bridges.

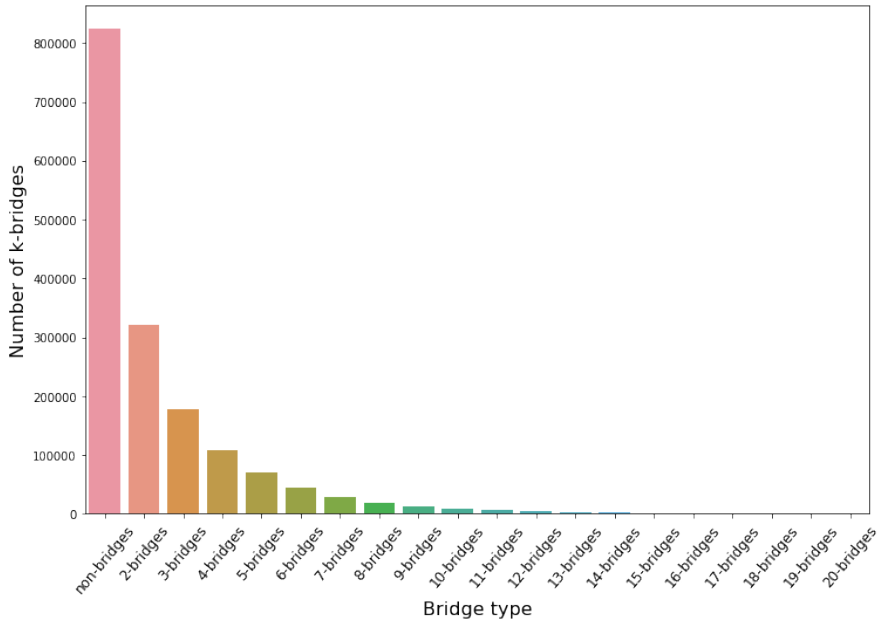


Fig. 2.19: Distribution of the k-bridges against k in Yelp after the removal of "Restaurants"

Then, we computed again the networks $\mathcal{M}^{1\%}$ - $\mathcal{M}^{15\%}$. They are shown in Figure 2.20. From the analysis of this figure, we can observe that the connection level of these networks slightly decreases compared to the corresponding networks with "Restaurants", albeit this trend remains the same from a qualitative viewpoint. This can also be deduced from the values of the density and the average clustering coefficient shown in Table 2.14.

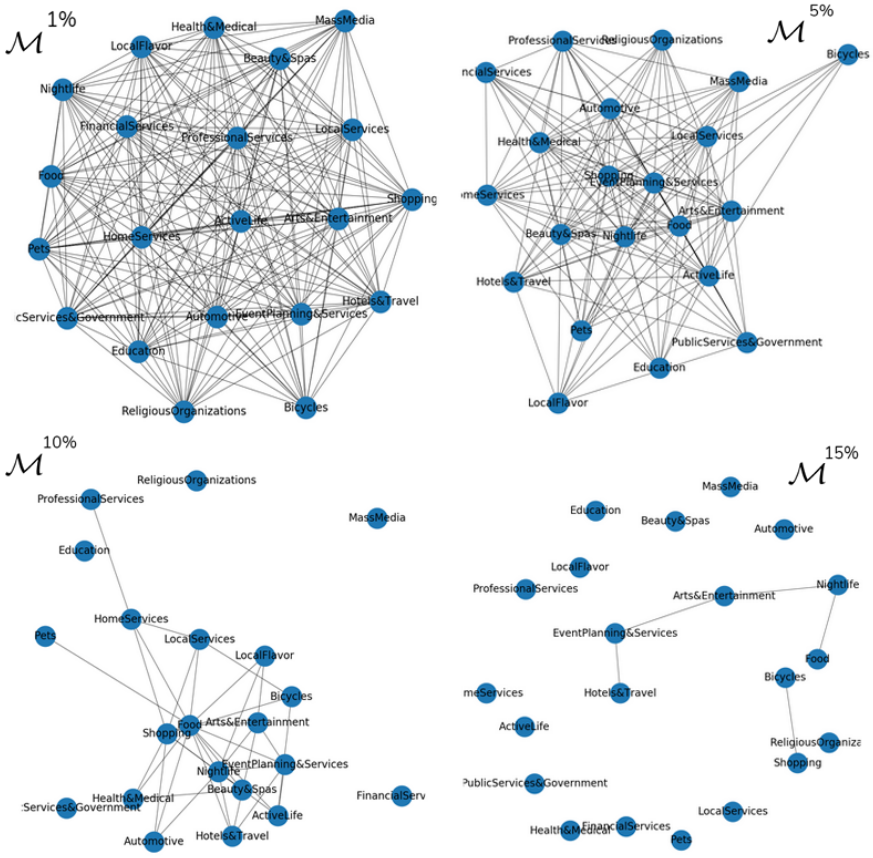


Fig. 2.20: The networks $\mathcal{M}^{1\%}$ - $\mathcal{M}^{15\%}$ after the removal of "Restaurants"

	$\mathcal{M}^{1\%}$	$\mathcal{M}^{5\%}$	$\mathcal{M}^{10\%}$	$\mathcal{M}^{15\%}$
Density	0.976	0.719	0.176	0.024
Average Clustering Coefficient	0.979	0.846	0.452	0

Table 2.14: Values of the density and the average clustering coefficient for the networks $\mathcal{M}^{1\%}$ - $\mathcal{M}^{15\%}$ after the removal of "Restaurants"

Finally, we computed the maximum and sub-maximum values for all centrality measures for the new networks obtained after the removal of "Restaurants". The results are reported in Table 2.15. From the analysis of this table, we can observe that

the values are slightly lower than before, but the trend is confirmed. This allows us to conclude that the trends and features related to co-reviews in Yelp are intrinsic to this social medium and are not biased by the presence of "Restaurants". This macro-category certainly contributes to strengthen these trends but it does not upset them.

Clearly, in absence of "Restaurants", the macro-category that plays the main role in the co-reviews is "Food". Instead, different macro-categories often alternate in the role of sub-maximum for the centrality measures into consideration.

	$\mathcal{M}^{1\%}$	$\mathcal{M}^{5\%}$	$\mathcal{M}^{10\%}$	$\mathcal{M}^{15\%}$
Maximum Degree Centrality	1 (Beauty&Spas)	1 (Food)	0.65 (Food)	0.1 (Nightlife)
Sub-maximum Degree Centrality	1 (Food)	1 (Nightlife)	0.45 (Nightlife)	0.1 (EventPlanning&Services)
Maximum Closeness Centrality	1 (Beauty&Spas)	1 (Food)	0.662 (Food)	0.133 (Arts&Entertainment)
Sub-maximum Closeness Centrality	1 (Food)	1 (Nightlife)	0.511 (Shopping)	0.114 (EventPlanning&Services)
Maximum Betweenness Centrality	0.002 (Beauty&Spas)	0.044 (Food)	0.271 Food)	0.021 (Arts&Entertainment)
Sub-maximum Betweenness Centrality	0.002 (Food)	0.044 (Nightlife)	0.074 (HomeServices)	0.016 (Nightlife)
Maximum Eigenvector Centrality	0.223 (Beauty&Spas)	0.273 (Shopping)	0.49 (Food)	0.577 (Arts&Entertainment)
Sub-maximum Eigenvector Centrality	0.223 (Food)	0.273 (Nightlife)	0.403 (Nightlife)	0.5 (Nightlife)

Table 2.15: Maximum and sub-maximum values of the various centrality measures and the corresponding macro-categories in the networks $\mathcal{M}^{1\%}$ - $\mathcal{M}^{15\%}$ after the removal of "Restaurants"

After having performed a deep analysis on the features of k-bridges in Yelp, in the following section, we verify if some results on k-bridges found in this social network are general or specific to it.

2.2.2 Validation of k-bridge properties in other networks

This section is devoted to validating the k-bridge properties mentioned above in other networks. Actually, due to space constraints, we limit our analysis to only some of the properties found above. We verify their validity first in Reddit and, then, in the network of patent inventors.

Validation of k-bridge properties in Reddit

We downloaded all the data for the investigation activity from the `pushshift.io` website, one of the most known Reddit data sources. Our dataset contains all the posts published on Reddit from January 1^{st}, 2019 to February 1^{st}, 2019. The number of posts available for our investigation was 485,623.

As a first task, we selected the 30 subreddits with the highest number of posts. According to our model, as described in Section 2.1.1, all the authors of a subreddit represented a community in our model, and the authors who submitted one or more posts in at least two subreddits represented bridges. Specifically, a k-bridge is an author who posted in exactly k subreddits.

As a first experiment, we computed the distribution of k-bridges against k in Reddit. It is shown in Figure 2.21. From the analysis of this figure, we can see that it follows a power law. This result is in total agreement with the one obtained for Yelp and reported in Figure 2.3.

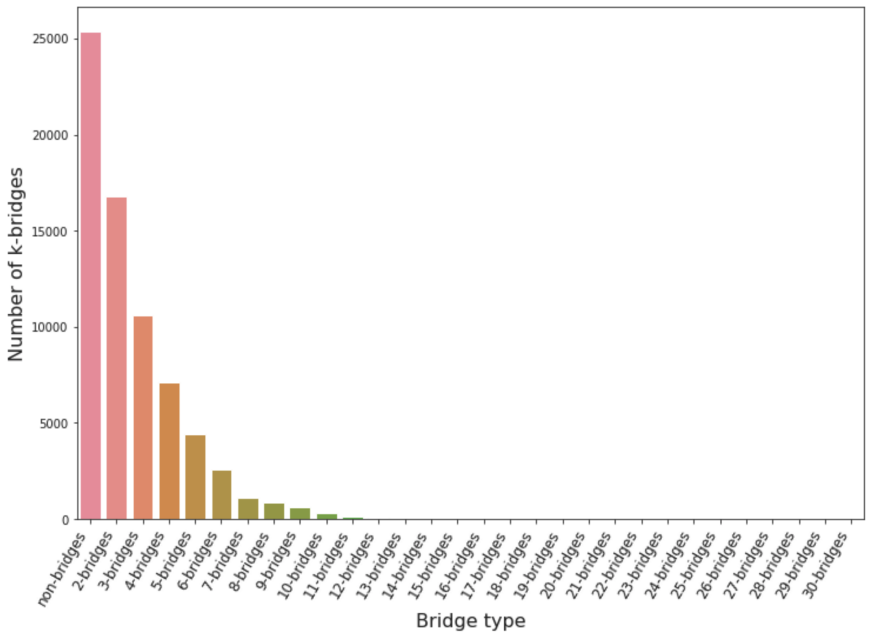


Fig. 2.21: Distribution of the k-bridges against k in Reddit

As a second experiment, we considered the co-posting network $\mathcal{U}^{cp}$, defined in Section 2.1.1. We recall that, in this network, there is a node for each user who submitted at least one post in at least one of the 30 subreddits into consideration, and there is an arc between two users if both of them contributed to the same subreddit. The co-posting network in Reddit corresponds to the co-review network in Yelp. In that case, we had found that there is a backbone among the bridges of this network. Therefore, it appears interesting to verify whether this property exists also in $\mathcal{U}^{cp}$.

For this purpose, for each bridge (non-bridge), we considered the fraction of co-posters that were bridges (non-bridges). The results obtained are shown in Table 2.16. They denote that there is a backbone among bridges in $\mathcal{U}^{cp}$. They also confirm what we had obtained for Yelp in Table 2.6.

	Fraction of co-posters that are bridges	**Fraction of co-posters that are non-bridges**
Bridges	0.9234	0.0585
Non-bridges	0.7531	0.2243

Table 2.16: Types of co-posters for bridges and non-bridges in $\mathcal{U}^{cp}$

Finally, we verified if there is a correlation between k-bridges and power users. For this purpose, we computed the distribution of the number of arcs for non-bridges, bridges, strong and very strong bridges. Preliminarily, by applying the same approach described in Section 2.1.2 for Yelp, we found that, in Reddit, the thresholds for strong bridges and very strong bridges are $th_s = 5$ and $th_{vs} = 9$, respectively.

Afterwards, we computed the distribution of the number of arcs for non-bridges, bridges, strong and very strong bridges. The results obtained are shown in Figure 2.22. This figure reveals that, as k grows, the power law distributions move to the right and flatten out. This result confirms the one in Figure 2.9 obtained for Yelp and tells us that also for Reddit there is a correlation between the strength of k-bridges and their degree centrality.

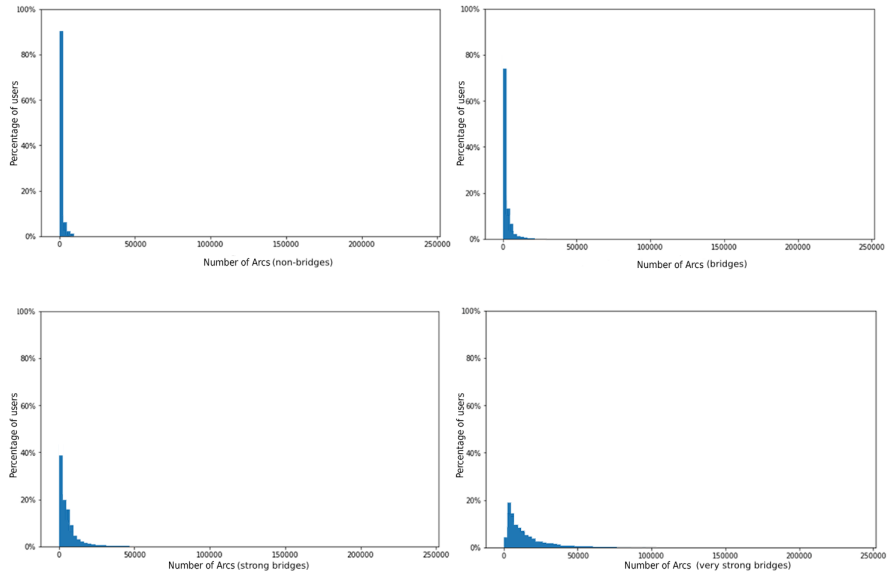


Fig. 2.22: Distributions of the number of arcs for non-bridges, bridges, strong and very strong bridges in Reddit

Validation of k-bridge properties in the network of patent inventors

Data about patents adopted in our analyses has been taken from the PATSTAT-ICRIOS database. It stores data about all patents from 1978 to the current years coming from about 90 patent offices worldwide. The number of patents taken into consideration is 9,605,147 and the number of inventors is, instead, 23,637,883.

According to our model, as described in Section 2.1.1, the set of inventors who filed at least one patent in an IPC class represents a community. Therefore, we have 127 communities. In this setting, the authors who filed patents in at least two IPC

classes represent bridges. A k-bridge is an author who filed patents that, in the whole, cover exactly k IPC classes.

Also in this case, we computed the distribution of k-bridges against k. We report it in Figure 2.23. From the analysis of this figure, we can see that it follows a power law. This result is in line with what we have seen for Yelp and Reddit.

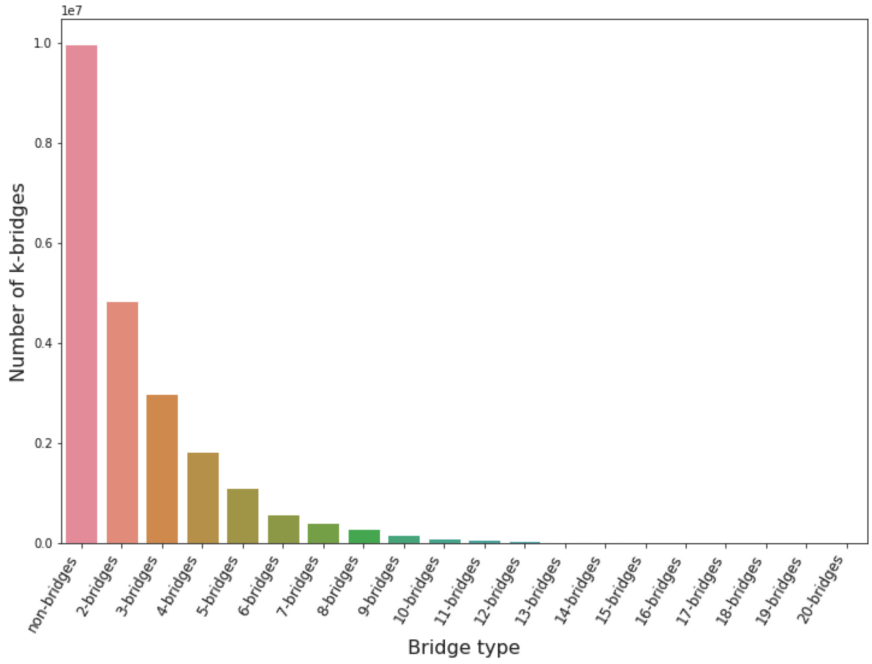


Fig. 2.23: Distribution of the k-bridges against k in the network of patent inventors

After this, we considered the co-inventing network $\mathcal{U}^{ci}$, defined in Section 2.1.1. Here, there is a node for each inventor and there is an arc between two inventors if both of them filed at least one patent together. Clearly, the co-inventing network strictly corresponds to the co-posting network of Reddit and the co-review network of Yelp.

In order to verify if there exists a backbone among the bridges of this network, for each bridge (resp., non-bridge), we considered the fraction of co-inventors that were bridges (resp., non-bridges). The results, reported in Table 2.17, clearly denote the existence of a backbone among the bridges in $\mathcal{U}^{ci}$, analogous to the ones found in $\mathcal{U}^{cr}$ for Yelp and in $\mathcal{U}^{cp}$ for Reddit.

Finally, we verified if there is a correlation between k-bridges and power users also in $\mathcal{U}^{ci}$. In this case, a reasoning analogous to the one described in Section 2.1.2 allowed us to find that, in the network of patent inventors, the threshold th_s for strong bridges is 5 whereas the threshold th_{vs} for very strong bridges is 10.

	Fraction of co-inventors that are bridges	**Fraction of co-inventors that are non-bridges**
Bridges	0.9632	0.0563
Non-bridges	0.7924	0.2356

Table 2.17: Types of co-inventors for bridges and non-bridges in $\mathcal{U}^{ci}$

We computed the distribution of the number of arcs for non-bridges, bridges, strong and very strong bridges. The results are reported in Figure 2.24. They denote that, as k grows, the power law distributions move to the right and flatten out. This result is a further confirmation of the ones reported in Figure 2.9 for Yelp and in Figure 2.22 for Reddit, i.e., that also in the network of patent inventors there is a correlation between the strength of k-bridges and the degree centrality.

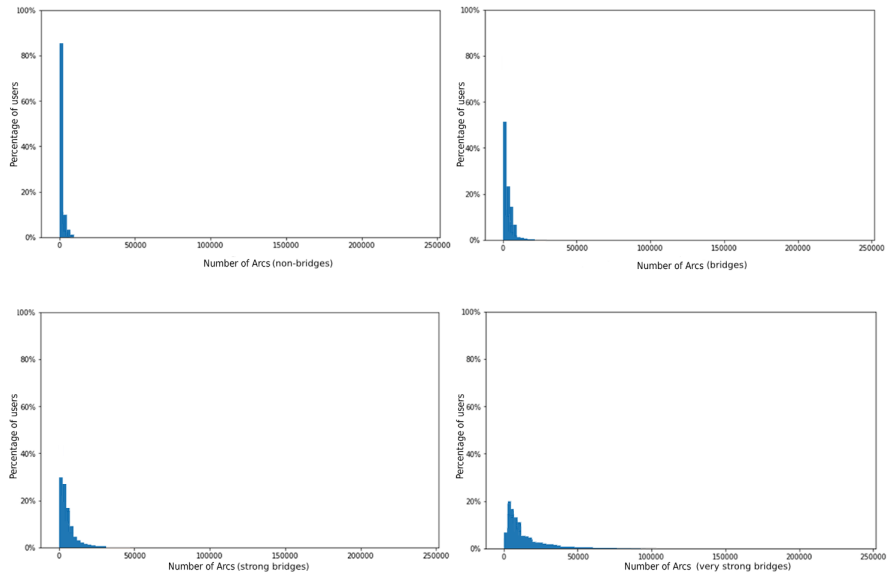


Fig. 2.24: Distributions of the number of arcs for non-bridges, bridges, strong and very strong bridges in the network of patent inventors

After having verified that the main properties of k-bridges are intrinsic to this concept and not specific to only Yelp, in the next section, we present two use cases that could highly benefit from the knowledge of k-bridges.

2.2.3 Applications of k-bridges

The social networking phenomenon has completely changed the way people conceive interaction with each other and consume information. Several studies have investigated the consequences of the massive proliferation of Online Social Networks that we are observing in these years.

From a consumer point of view, social networks bring impressive benefits, such as richer and more participative information, a broader selection of products, more competitive pricing, and cost reduction. Instead, in the industry context, 81% of firms plan to invest in social networking sites, and more than 50% of them consider digital advertising and marketing as a priority area of investment [**?**]. Actually, several online services, like Yelp (but also TripAdvisor[1], and, in a certain sense, Booking[2], Airbnb[3], etc.), have been conceived just to encourage this kind of interaction. Of course, in this scenario, obtaining a very large number of positive reviews is crucial for businesses. Therefore, designing ad-hoc marketing and advertising campaigns is extremely important. In the next paragraphs, we describe in detail two case studies related to this concept, which massively exploit k-bridges to conduct marketing campaigns and support business decisions in Yelp.

Finding the best targets for a marketing campaign

This first case study refers to a scenario in which a business is planning to expand its activities including services that belong to new Yelp categories, along the ones already covered. The business already performed an internal evaluation analysis with the goal of identifying the best services, possibly referring to new categories, to improve its revenues. The next step concerns the design of a goal-oriented marketing campaign to foster the diffusion of the new services among new potential customers. Of course, a naive flooding approach of advertising messages appears not convenient, as it would not be possible to properly target the advertising campaign based on customer features. Moreover, it would lead to an excessive amount of unwanted messages from a user point of view.

For these reasons, the knowledge derived from the identification of k-bridges, who are already customers of both the original categories of interest for the business and the new ones it intends to embrace, plays a crucial role. Indeed, these bridges can be considered as links among the different communities they belong to and, hence, they can be "engaged" as convenient diffusion points to properly target the marketing campaign.

Now, let us consider a simple example scenario where a business, which already provides services belonging to the *Restaurant* category of Yelp, decides to include new services belonging to two new related categories, namely *Nightlife* and *Hotel&Travel*. In this case, according to the reasoning above, the following steps can be performed to obtain a very effective marketing campaign.

[1] `https://www.tripadvisor.com`
[2] `https://www.booking.com`
[3] `https://www.airbnb.com`

First, 3-bridges are identified as the most correct typology of users to involve. Indeed, 3-bridges can potentially link together all and only the three categories of interest. Actually, more powerful bridges (e.g., 4-bridges or higher) could have been also considered; however, this would lead to the inclusion of other categories not interesting for the business, which in turn would lead to a reduction of the campaign effectiveness.

After that, among all the available 3-bridges, the ones belonging to just the three categories of interest are selected.

Now, considering that the campaign success strongly depends on the capability of k-bridges to promote the new services, a metric to measure it must be introduced. This metric should consider the inclination of a bridge to review businesses, her proneness to create an articulated friend network, and her constant activity level over time. In Equation 2.1, we report a possible simple implementation of such a metric (clearly, future research efforts could be made to define a more sophisticated metric):

$$\mu_i = \frac{nr_i \cdot nf_i}{nd_i} \tag{2.1}$$

Here, nr_i represents the number of reviews performed by the 3-bridge u_i, nf_i denotes the dimension of the network of her friends, and, finally, nd_i indicates the number of days u_i is enrolled in the platform. Here, nr_i directly measures the activity level of u_i; however, this is not sufficient because early adopters of the platform typically make a very high number of reviews in a very short amount of time, but not all of them remain active over time. For this reason, we consider two other important factors, i.e., the number of friends and the time interval in which they performed their activities. As the creation of a strong and rich network of friends requires time, nf_i allows us to exclude early adopters who left the platform too soon. Instead, nd_i acts as a weight and allows the estimation of the real activity level over time.

Now, the business can use the metric above to sort the set of 3-bridges according to their capability of promoting its services. Finally, it selects the top bridges as the target for its marketing campaign. The fact that the selected 3-bridges are members of all the three categories of interest increases the possibility that they can help the business to be known in the new communities.

The solution above, sketched for the simple example considered, can be easily extended and generalized for any similar application scenario with any number of involved categories. The overall process is described by Algorithm 2.

Input

- D, a dataset of a Social Network
- k, the number of communities of interest for the marketing campaign

Output

- $\overline{B_k}$, the k-bridges to consider for the marketing campaign

Require: `getInfo`(u_i), a function returning a DataFrame containing information about the number of reviews, the number of friends, and the days of enrollment in the platform of a user u_i; `bridgeExtraction(k)`, a function implementing Algorithm 1 and returning the set of k-bridges; S_k, a set of scores

B_k = `bridgeExtraction(k)`
for $u_i \in B_k$ **do**
  $info_{u_i}$ = `getInfo`(u_i)
  $nr_i = info_{u_i}[\text{“}reviews\text{”}]$, $nf_i = info_{u_i}[\text{“}friends\text{”}]$, $nd_i = info_{u_i}[\text{“}days\text{”}]$
  $\mu_i = (nr_i \cdot nf_i)/nd_i$
  add μ_i to S_k
end for
$\overline{B_k}$ = sort B_k by S_k
return $\overline{B_k}$

Algorithm 2: Algorithm for finding the best targets of a marketing campaign

Finding new products/services to propose

This second case study is strictly related to the previous one. However, it deals with a situation in which a business is still conducting a market analysis to identify new services, belonging to new categories, that it can propose. In this context, the knowledge acquired by analyzing k-bridges can be used to know the most popular categories related to the ones already covered by the business. Indeed, in this scenario, the review activities of k-bridges implicitly encode association rules among categories. Such rules can be represented as:

$$review(\mathcal{C}_k) \Rightarrow \bigwedge_{i=1}^{k-1} review(\mathcal{C}_i)$$

Here, the term $\bigwedge_{i=1}^{k-1} review(\mathcal{C}_i)$ represents the logic conjunction of a sequence of reviewing activities in $k-1$ different categories.

Intuitively, the larger k the more disparate are the different categories included in the conjunction. For this reason, it is first necessary to identify the optimal value of k in the extraction of meaningful association rules among categories. For this purpose, it is possible to adopt a modified version of the Elbow-method [344], a very

common strategy to identify the correct number of clusters in a typical clustering scenario. The basic idea underlying our approach to perform this task is to carry out an iterative task. At each iteration:

1. the value of k is increased;
2. Algorithm 2 is used to identify k-bridges;
3. k-bridges being members of the original category of the business are selected;
4. all the additional categories (involved by the identified k-bridges) are considered;
5. their average semantic distance with respect to the starting ones is estimated.

This procedure ends when, during an iteration, the average estimated distance for the new categories is considered too high with respect to the marketing objectives of the business.

At this point, by analyzing the k-bridges involving the original categories and the closest ones identified during the iterations, it is possible to identify a set of association rules between the original categories of the business and the new ones. For each rule, it is possible to estimate the corresponding *support* and *confidence*[4]. The obtained information can be used by the business to decide which new categories are more suitable for its development.

[4] Observe that, borrowing some ideas from the association rules theory, in our scenario, support can be defined as a measure of how frequently the new categories and the old ones appear together in k-bridges; instead, confidence quantifies how often the new categories appear in those k-bridges where the original categories appear too.

3

Detecting user stereotypes and their assortativity

In recent years, Reddit has attracted the interest of many researchers due to its popularity all over the world. In this chapter, we aim at providing a contribution in the knowledge of this social network by investigating three of its aspects, interesting from the scientific viewpoint, and, at the same time, by analyzing a large number of applications. In particular, we first propose a definition and an analysis of several stereotypes of both subreddits and authors. This analysis is coupled with the definition of three possible orthogonal taxonomies that help us to classify stereotypes in an appropriate way. Then, we investigate the possible existence of author assortativity in this social medium; specifically, we focus on co-posters, i.e. authors who submitted posts on the same subreddit.

The material presented in this chapter was derived from [233].

3.1 Methods

3.1.1 Dataset description

The dataset required for our activity was downloaded from the `pushshift.io` website, which is one of the most known Reddit data sources. Our dataset contains all the posts published on Reddit from January 1^{st}, 2019 to September 1^{st}, 2019. All the posts wrote in a month were added to the dataset at the end of the next month. The number of posts available for our investigation was 150,795,895. For each post, we considered the following set of attributes: `id`, `subreddit`, `title`, `author`, `created_utc`, `score`, `num_comments` and `over_18`.

In order to carry out our experiments, we used a server equipped with 16 Intel Xeon E5520 CPUs and 96 GB of RAM with the Ubuntu 18.04.3 operating system. We adopted Python 3.6 as programming language, its library Pandas to perform ETL operations on data, and its library NetworkX to perform operations on networks.

During the ETL phase, we observed that some of the available posts referred to authors that had left Reddit. We decided to remove these posts from our dataset. At the end of this last activity the number of posts at our disposal was 122,568,630.

We computed the number of authors who submitted these posts; it was equal to 12,464,188. Then, we found the number of the subreddits which they referred to; it was equal to 1,356,069.

Now, we describe some preliminary investigations on Reddit, concerning posts, comments, and authors.

Investigation on posts

We started this investigation by performing the following analyses on posts:

- distribution of subreddits against posts (Figure 3.1); it follows a power law with $\alpha = 1.651$ and $\delta = 0.014$;
- distribution of authors against posts (Figure 3.2); it follows a power law with $\alpha = 1.431$ and $\delta = 0.016$;
- distribution of posts against scores (Figure 3.3); it follows a power law with $\alpha = 1.600$ and $\delta = 0.005$.

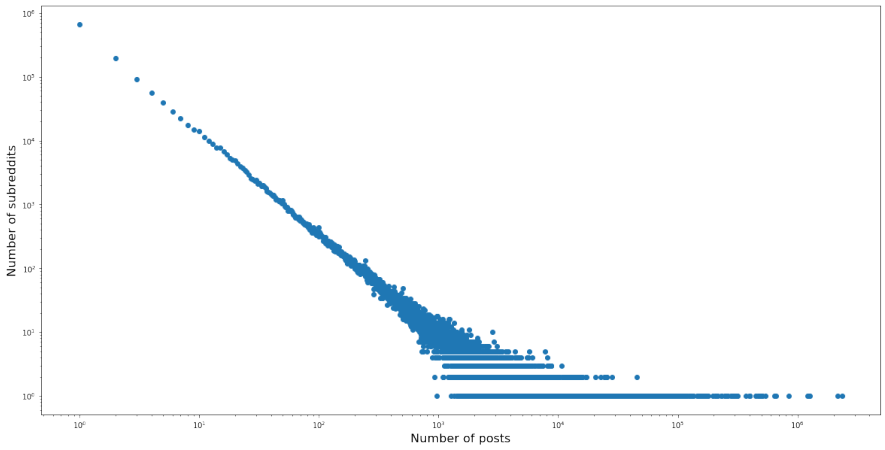


Fig. 3.1: Distribution of subreddits against posts (log-log scale)

The maximum number of posts with the same score is 51,721,824. Interestingly, these posts have associated a score equal to 1. Instead, the number of posts with a score equal to 0 or 2 is much smaller. This trend can be explained considering that a post submitted on Reddit starts with a score of 1. As a consequence, when no other author upvotes or downvotes it, the final score of the post is 1.

We also observe that no post has a negative score. This fact is due to Reddit that shows and returns a score equal to 0 for a post whenever the number of downvotes is higher than the number of upvotes, i.e., also when the real score of the post is negative. So, posts with a score equal to 0 are to all intents and purposes intended as "negative" posts.

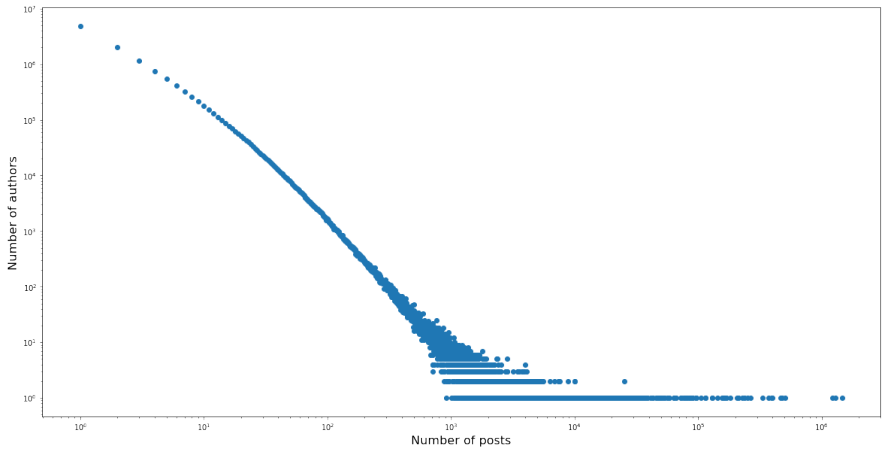


Fig. 3.2: Distribution of authors against posts (log-log scale)

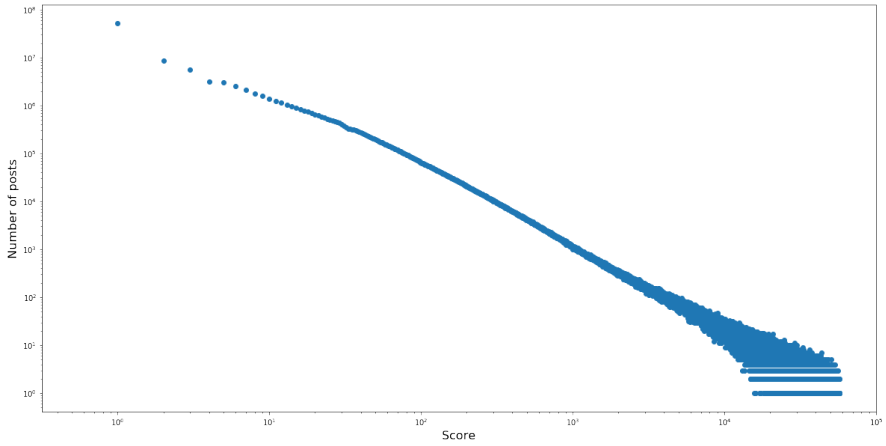


Fig. 3.3: Distribution of posts against scores (log-log scale)

At this point, we also computed:

- the distribution of authors against negative posts (Figure 3.4); it follows a power law with $\alpha = 2.274$ and $\delta = 0.030$.
- the distribution of authors against positive posts (Figure 3.5); it follows a power law with $\alpha = 2.074$ and $\delta = 0.014$.

As for these two distributions, we found that the number of positive posts is about 16 times the number of negative ones.

Analysis of positive and negative posts for SFW and NSFW cases

In the previous section, we have observed that each post has a score, initially equal to 1, which can increase or decrease based on the upvotes or downvotes of users. Actually, Reddit does not report the posts with a negative score in its database. For this reason, the values of the scores both in Reddit and in `pushshift.io` range in the interval $[0, +\infty)$. In this setting, posts with a score equal to 0 are particularly relevant,

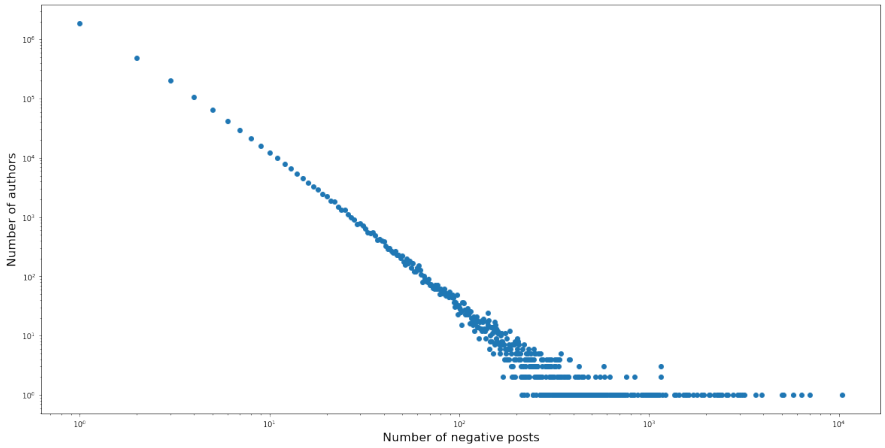


Fig. 3.4: Distribution of authors against negative posts (log-log scale)

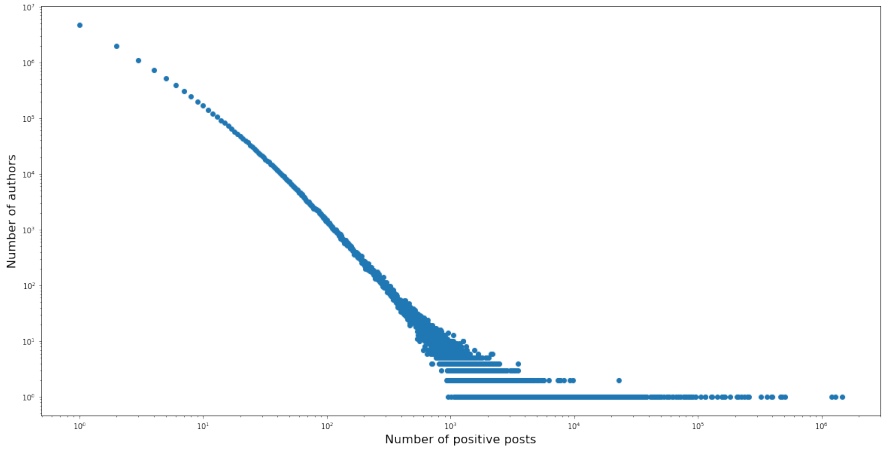


Fig. 3.5: Distribution of authors against positive posts (log-log scale)

because they are the only ones that have been rated negatively by at least one user, or have received more downvotes than upvotes.

We computed the distributions of authors against negative posts for both SFW and NSFW posts. In both cases, we have found that they follow a power law. We report the main parameters of these distributions in Table 3.1.

A Wilcoxon rank sum test showed that the number of authors of Jan-Feb SFW negative posts was statistically significantly higher than the corresponding one of NSFW posts ($\tau = 5.1 \cdot 10^{-4}, p < 0.01$).

These conclusions, although interesting, must be intertwined with those regarding positive posts, to better characterize the features of negative ones. For this reason, we computed the distributions of authors against positive posts. Also in this case, the distributions follow a power law similar to the previous ones. We report the values of the main parameters of these distributions in Table 3.2.

Parameter	*SFW posts Jan-Feb*	*NSFW posts Jan-Feb*	*SFW posts Mar-Apr*	*NSFW posts Mar-Apr*
Maximum number of authors	66,162 (92.31%)	24,607 (74.86%)	61,254 (91.98%)	24,172 (73.87%)
Number of authors of the 99 percentile	40,028	11,606	40,024	11,598
Maximum number of posts	133 (9.64%)	460 (14.38%)	103 (8.98%)	399 (13.76%)
Number of posts of the 99 percentile	126	369	122	370
Average number of authors	1,666	505	1,691	544
Average number of posts	32	49	28	47
α (power law parameter)	1.4360	1.4349	1.5512	1.4360
δ (power law parameter)	0.0615	0.0,0616	0.0543	0.0616

Table 3.1: Parameters of the distributions of authors against negative posts

Parameter	*SFW posts Jan-Feb*	*NSFW posts Jan-Feb*	*SFW posts Mar-Apr*	*NSFW posts Mar-Apr*
Maximum number of authors	522,540 (79.66%)	124,054 (56.56%)	519,774 (79.54%)	126,602 (56.89%)
Number of authors of the 99 percentile	9,083	4,346	9,080	4,352
Maximum number of posts	18,684 (11.88%)	16,383 (5.77%)	16,481 (10.67%)	15,564 (5.73%)
Number of posts of the 99 percentile	5,165	4,638	5,160	4,641
Average number of authors	2,018	418	1,944	394
Average number of posts	483	541	493	514
α (power law parameter)	1.4318	1.5145	1.4855	1.5498
δ (power law parameter)	0.0311	0.0263	0.0275	0.0291

Table 3.2: Parameters of the distributions of authors against positive posts

A Wilcoxon rank sum test indicated that the number of authors of Jan-Feb SFW positive posts was statistically significantly higher than the corresponding one of NSFW posts ($\tau = 1.1 \cdot 10^{-4}, p < 0.01$).

We now compare Tables 3.1 and 3.2 to extract the features characterizing negative posts versus positive ones. There are no significant differences between positive and negative posts in the maximum and average number of authors of NSFW and SFW posts. The same is true for the average number of posts and the trends of the power law distributions. However, there is a very interesting aspect that differentiates negative posts from positive ones. Indeed, the maximum number of negative posts is much higher for NSFW posts than for SFW ones. This trend is not found in positive posts.

The explanation behind this result is the same as the one seen previously.

Investigation on comments

As for this investigation, we computed:

- The distribution of subreddits against comments (Figure 3.6); it follows a power law with $\alpha = 1.730$ and $\delta = 0.015$.
- The distribution of the average number of comments against the scores of the posts they refer to (Figure 3.7). Interestingly, in this case, we have a roughly

Gaussian distribution, whose mean is at a score near to 50,000. The distribution presents several outliers. For instance, for a score equal to 79,470, we have a post with a number of comments equal to 71,225.

- the distribution of posts against comments (Figure 3.8); it follows a power law with $\alpha = 1.455$ and $\delta = 0.011$.

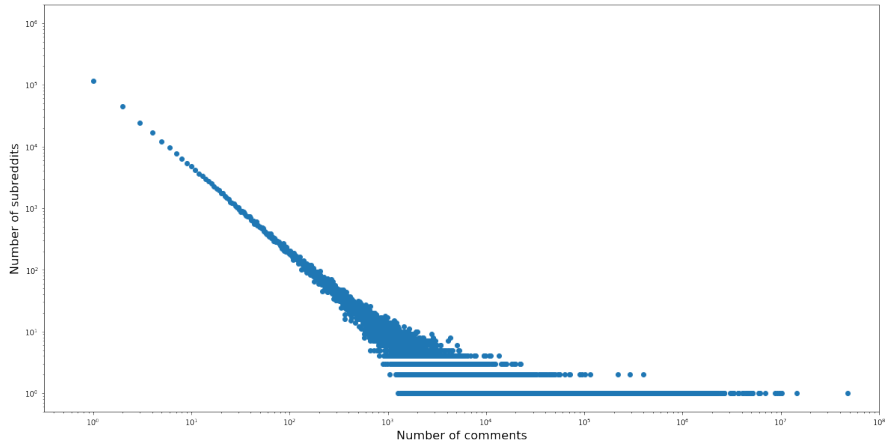


Fig. 3.6: Distribution of subreddits against comments (log-log scale)

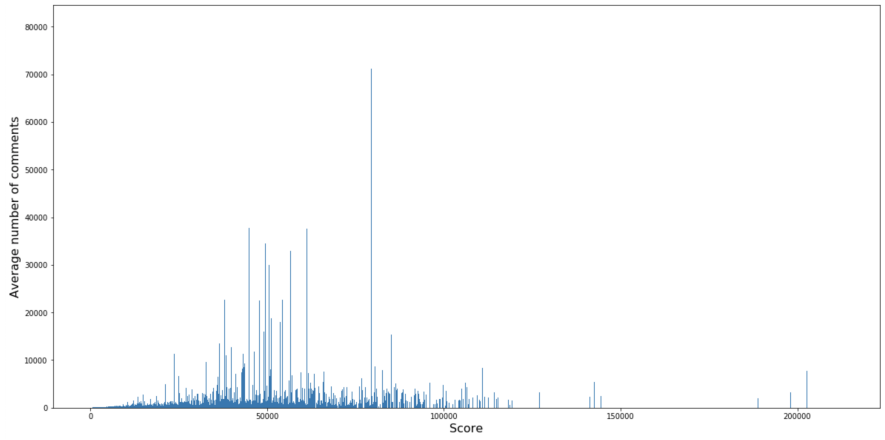


Fig. 3.7: Distribution of the average number of comments against the scores of the posts they refer to

Finally, we considered the 150 posts with the highest number of comments and the subreddits they were submitted to. We obtained only 31 subreddits. Then we computed the average number of comments for *all* the posts submitted in each of these subreddits. The results obtained are reported in Figure 3.9. From the analysis of this figure, we can observe that the distribution is very irregular. It decreases

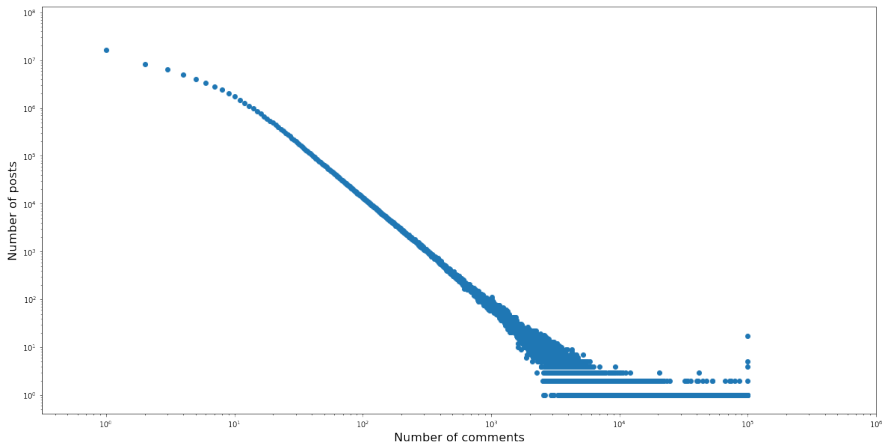


Fig. 3.8: Distribution of posts against comments (log-log scale)

quickly for the first three subreddits, very slowly for the next 13 subreddits, quickly for the next 9 subreddits and, finally, it suddenly drops and becomes almost zero.

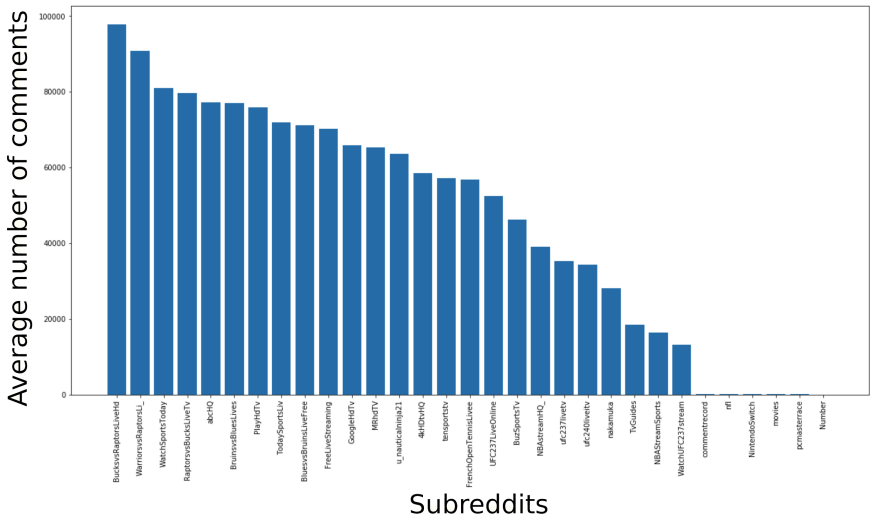


Fig. 3.9: Distribution of the average number of comments submitted to the subreddits receiving the 150 most commented posts

Investigation on authors

First, we determined the distribution of authors against subreddits (Figure 3.10). It follows a power law with $\alpha = 1.702$ and $\delta = 0.081$.

Afterwards, we selected the 150 posts with the highest number of comments and the corresponding authors. Interestingly, we had only 26 authors for all the 150 posts. These can be considered as the most commented authors in Reddit and, maybe, they are influencers. Then, we computed the average number of comments

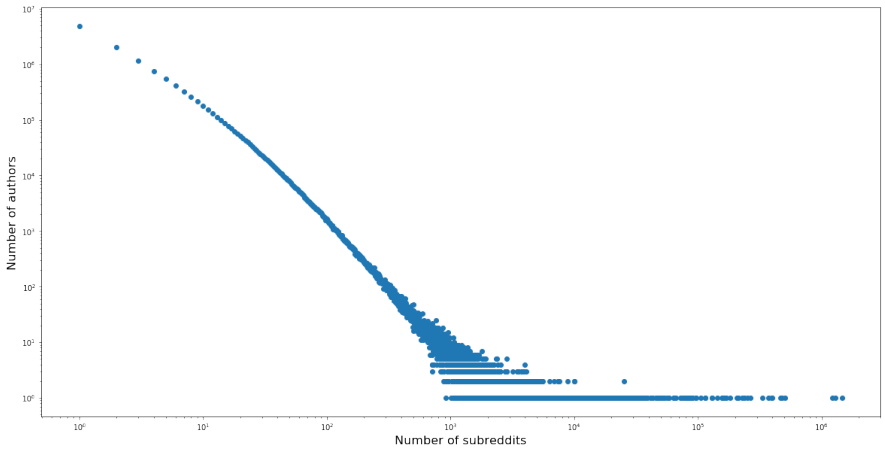


Fig. 3.10: Distribution of authors against subreddits (log-log scale)

for *all* the posts each author submitted. The results obtained are reported in Figure 3.11. From the analysis of this figure we can observe that the decrease of the distribution is roughly stepwise.

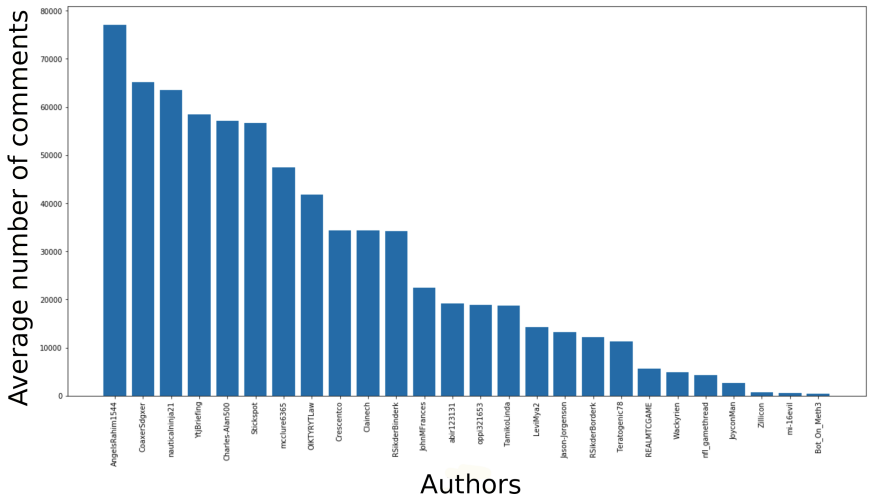


Fig. 3.11: Distribution of the average number of comments received against the authors submitting the 150 most commented posts

3.1.2 Stereotyping subreddits

In order to determine some possible stereotypes of subreddits, we start investigating the subreddit lifespan. As a first step, we considered the subreddits created in January 2019 and then verified the month when they performed their last activity (and, therefore, presumably died). The results obtained are reported in Figure 3.12. Here, an activity level of 1 implies that the subreddit died in the same month it was born,

an activity level of 2 suggests that it died one month after it was born, and so on. An activity level of 8 indicates that it is still alive (we recall that our dataset comprises data from January 1^{st}, 2019 to September 1^{st}, 2019). We proceeded in the same way for the subreddits created in February, March, and so forth. For instance, in Figure 3.13, we report the trends of the subreddits created in February 2019 and in March 2019.

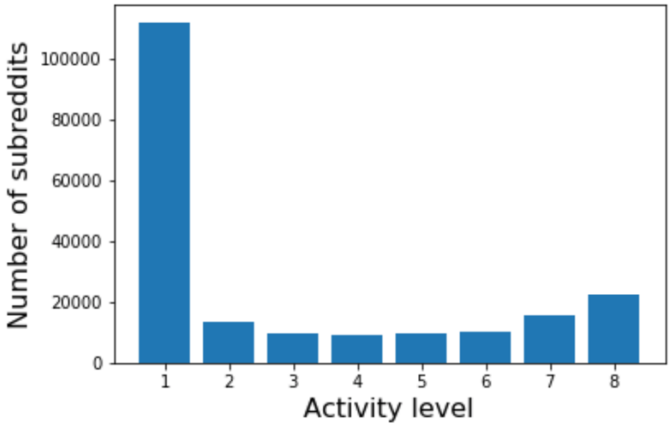


Fig. 3.12: Lifespan of the subreddits created in January 2019

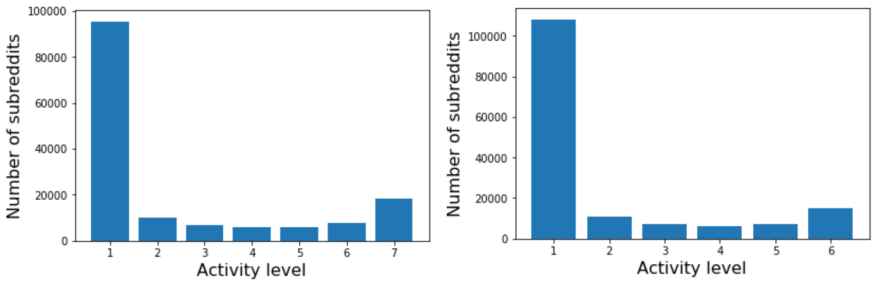


Fig. 3.13: Lifespan of the subreddits created in February 2019 (at left) and March 2019 (at right)

After this, we focused on those subreddits died in the same month they were born. We analyzed their corresponding lifespan and we observed that almost all of them died in the same day they were born. For instance, in Figure 3.14, we report the trends of the subreddits born and died in February 2019 and in March 2019.

Then, we decided to deeply investigate those subreddits died in the same day they were born. We computed their distribution against the number of their posts. Figure 3.15 shows what happens for January 2019; the same trend can be observed for the other months of this year. Clearly, this distribution follows a power law, a trend that can be observed also for similar subreddits born in the other months. From

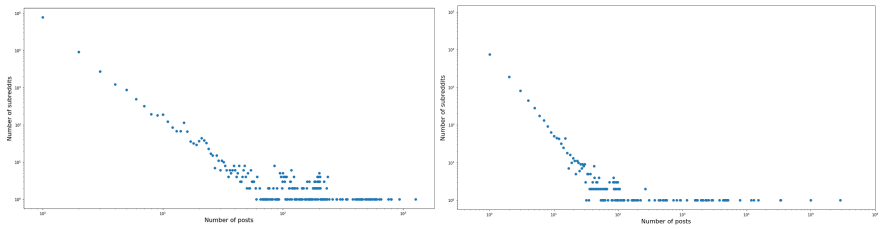


Fig. 3.14: Lifespan of the subreddits born and died in February 2019 (at left) and March 2019 (at right)

its analysis we observe that most of the subreddits, which died in the same day they were born, have only one post. At this point, we computed the distribution of these subreddits against the number of comments. In Figure 3.16, we show the subreddits of January 2019, even if the same trend can be observed for the other months of this year. From the analysis of this figure we can note that this distribution follows a power law. Furthermore, most of these subreddits have no comments.

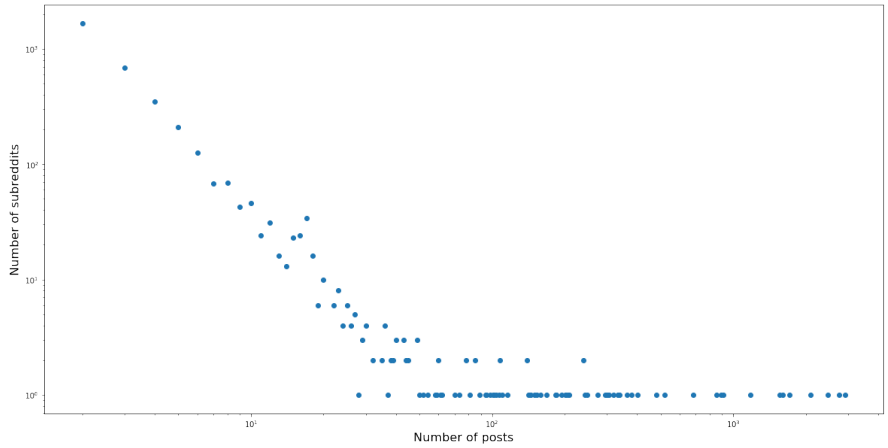


Fig. 3.15: Distribution of the subreddits of January 2019 died in the same day they were born against the number of their posts

Next, we examined a second class of subreddits, similar to the previous one. In fact, we selected all those subreddits that died one day after they were born. Again, we first computed their distribution against the number of posts. In Figure 3.17, we show what happens for the subreddits of January 2019; again, the same trend was found for all the other months. This distribution follows a power law, which was expected. The unexpected thing was that the minimum number of posts was 2 and not 1. Even more unexpectedly, this trend is also confirmed for the subreddits with the same features born in the other months. After that, we computed the distribution of these subreddits against the number of comments. In Figure 3.18, we show it for

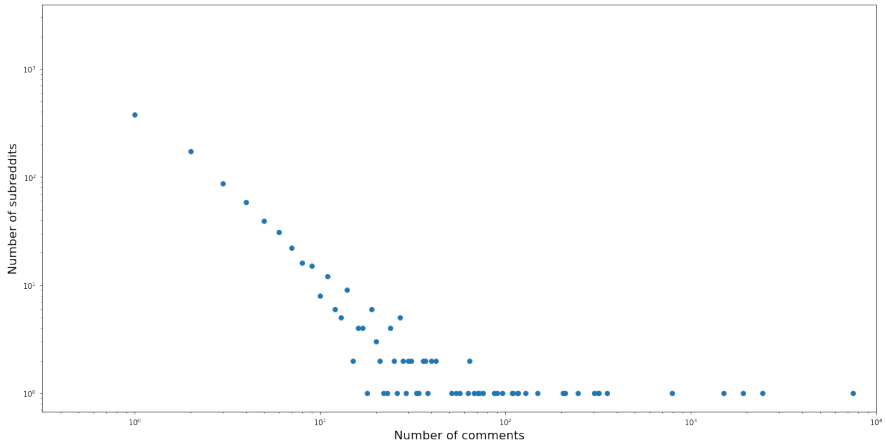


Fig. 3.16: Distribution of the subreddits of January 2019 died in the same day they were born against the number of their comments

the subreddits of January 2019; the same trend can be observed for all the other months. From the analysis of this figure, we note that this distribution follows a power law. Furthermore, most of these subreddits have no comments.

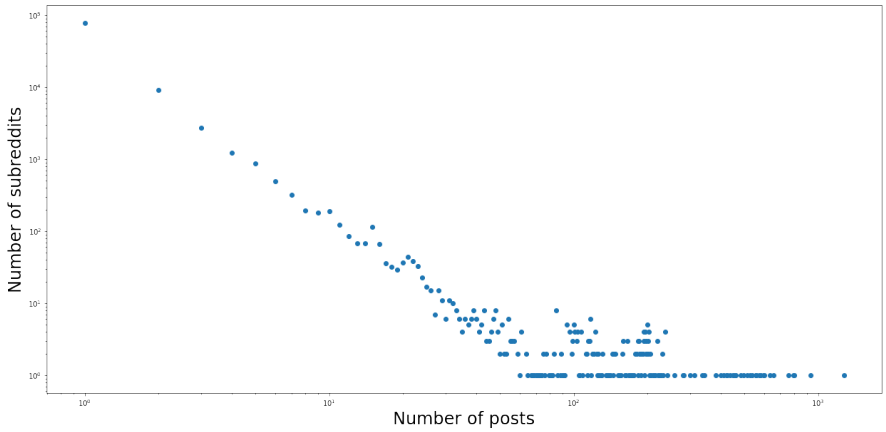


Fig. 3.17: Distribution of the subreddits of January 2019 died one day after they were born against the number of their posts

Note that the two classes of subreddits above have a proper characterization that differentiates them from all the other classes of subreddits (for instance, the ones that survived for some months). They also have few features distinguishing them from each other. However, the number of their similarities is much higher than the number of their differences. As a consequence, both these two classes can be considered as a "macro-category" of stereotypes that we call "dead in crib". At this point, by deepening what we have found previously, we have determined the following

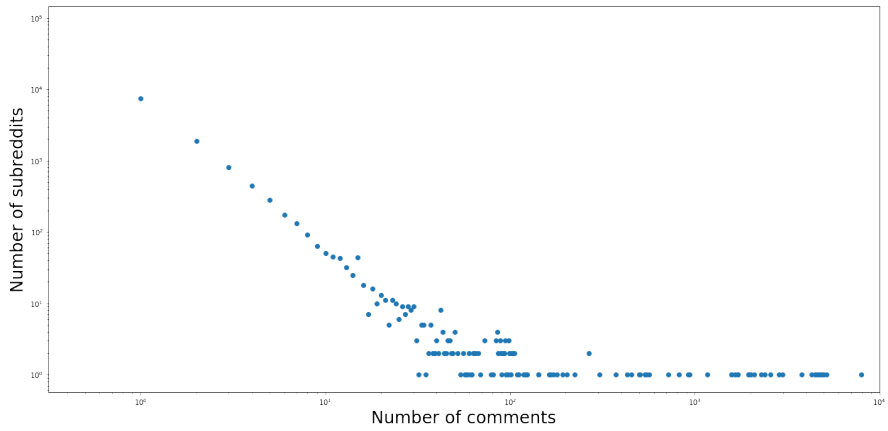


Fig. 3.18: Distribution of the subreddits of January 2019 died one day after they were born against the number of their comments

stereotypes characterizing the subreddits "dead in crib" (i.e., those subreddits who died at most one day after they were born):

- *User Profile*: it is associated with a user profile.
- *Unsuccessful Subreddit*: it initially stimulated several interactions. However, after few hours, these interactions finished and it quickly died.
- *Comment Grabber*: it had at least one post capable of stimulating a debate, even if minimal.
- *Private Community*: it requires an invitation to be accessed. It is often associated with a specific event of interest for a specific community.
- *Banned Subreddit*: it was banned probably because it was associated with a spammer.
- *Bot*: it can be recognized because its posts are always similar and consist of links and comments with links.

In order to characterize these stereotypes, and all the others that we will consider in the following, we have defined three possible orthogonal taxonomies. These are based on:

- the number of posts; we considered two possible classes, i.e., few posts and many posts;
- the number of comments; we considered two possible classes, i.e., few comments and many comments;
- the number of authors; we considered two possible classes, i.e., few authors and many authors.

Taking these three taxonomies into consideration, the previous stereotypes can be classified as shown in Tables 3.3 and 3.4.

Observe that a stereotype can often belong to both the classes of a taxonomy. This implies that it cannot be "categorized" based on that taxonomy. For instance, *Comment Grabber*, in presence of many comments and many authors, can be found with both few posts and many posts. This implies that this stereotype can be characterized only by the number of comments and the number of authors, but not by the number of posts. Analogously, in presence of many posts, *Banned Subreddit* cannot be characterized by the number of comments or the number of authors. By contrast, in presence of few posts, *Banned Subreddits* is characterized by few comments and few authors.

	Few Authors	**Many Authors**
Few Comments	User Profile Unsuccessful Subreddit Banned Subreddit	Unsuccessful Subreddit
Many Comments	Unsuccessful Subreddit Comment Grabber User Profile	Private Community Bot Unsuccessful Subreddit Comment Grabber

Table 3.3: Classification of stereotypes concerning the subreddits "dead in crib" - Few posts case

	Few Authors	**Many Authors**
Few Comments	User Profile Unsuccessful Subreddit Banned Subreddit	Unsuccessful Subreddit Bot Banned Subreddit
Many Comments	User Profile Banned Subreddit	Private Community Banned Subreddit Unsuccessful Subreddit Comment Grabber

Table 3.4: Classification of stereotypes concerning the subreddits "dead in crib" - Many posts case

After having investigated the stereotypes of the subreddits "dead in crib", we focused on the opposite category of subreddits, i.e., those survived for all the months of reference for our dataset. We collectively call them "survivors" in the following. We applied the same reasoning and tasks that we have made for the subreddits "dead in crib" and we obtained the following stereotypes:

- *User Profile*, *Bot*: these are the same ones we have seen for the subreddits "dead in crib".

- *Cringe / NSFW Subreddit*: it contains strange or strong-content posts, submitted by only one user, or, alternatively, it is an NSFW subreddit.
- *Niche Subreddit*: its topics are niche ones, and it draws the attention of users interested in them.
- *Successful Subreddit*.
- *Big Comment Grabber*: almost all the posts submitted in it stimulate a debate.
- *Utility Subreddit*: it is conceived to support a specific activity (think, for instance, of a subreddit where users ask for a translation).

Based on the three taxonomies defined above, the previous stereotypes can be classified as shown in Tables 3.5 and 3.6.

	Few Authors	**Many Authors**
Few Comments	User Profile Bot Cringe /NSFW Subreddit Niche Subreddit	Successful Subreddit Niche Subreddit
Many Comments	Successful Subreddit Niche Subreddit Big Comment Grabber	Big Comment Grabber Successful Subreddit Niche Subreddit

Table 3.5: Classification of stereotypes concerning the subreddits “survivors” - Few posts case

	Few Authors	**Many Authors**
Few Comments	Niche Subreddit	Cringe / NSFW Subreddit Niche Subreddit
Many Comments	Big Comment Grabber Utility Subreddit	Successful Subreddit

Table 3.6: Classification of stereotypes concerning the subreddits “survivors” - Many posts case

After these analyses on the stereotypes belonging to the two extreme categories “dead in crib” and “survivors”, we decided to apply the same reasonings and tasks to investigate a third category of stereotypes, intermediate between the two previous ones. Specifically, we focused on those subreddits that lived five months after their creation and, then, died. We call this category “undelivered promises” and we obtained the following stereotypes for it:

- *User Profile*, *Niche Subreddit*, *Bot*, *Cringe / NSFW Subreddit*, *Private Community*, *Banned Subreddit*: these are the same ones we have seen for the previous categories.

- *Unsuccessful Boomer*: it was successful for a while, but died after a period of decline.
- *Unsuccessful Zombie*: it was born without praise or blame, managed to survive for a while in a gray way and, finally, died.

Based on the three taxonomies that we defined above, the previous stereotypes can be classified as shown in Tables 3.7 and 3.8.

	Few Authors	**Many Authors**
Few Comments	User Profile Niche Subreddit Bot	Bot Cringe / NSFW Subreddit Niche Subreddit Unsuccessful Boomer
Many Comments	User Profile Private Community Unsuccessful Boomer Niche Subreddit	Niche Subreddit Private Community Unsuccessful Boomer

Table 3.7: Classification of stereotypes concerning the subreddits "undelivered promises" - Few posts case

	Few Authors	**Many Authors**
Few Comments	User Profile Cringe / NSFW Subreddit Bot Unsuccessful Zombie	Private Community Banned Subreddit Niche Subreddit
Many Comments	User Profile Bot Cringe / NSFW Subreddit	Cringe / NSFW Subreddit Banned Subreddit Unsuccessful Boomer

Table 3.8: Classification of stereotypes concerning the subreddits "undelivered promises" - Many posts case

3.1.3 Stereotyping authors

In order to determine the possible author stereotypes, we proceeded in a way analogous to what we have done to define subreddit stereotypes. In fact, also for authors, we found three macro-categories of stereotypes, namely "very positive", "neutral" and "very negative" authors. To better understand the reasoning underlying these categories, we recall that, in Section 3.1.1, we have found that the number of positive posts is about 16 times the number of negative ones in Reddit. As a consequence, it is possible to use this result as a baseline for a preliminary author classification. Specifically, we considered an author as "very positive" if the number of positive

posts submitted by her is at least $2 \cdot 16 = 32$ times the number of negative ones, which means at least twice the typical number of positive posts submitted for each negative one by a user. Instead, we considered an author as "neutral" if the number of positive posts submitted by her is between 1 and 16 times the number of negative ones. Finally, we considered an author as "very negative" if the number of negative posts submitted by her is at least 16 times the number of positive ones. Clearly, this classification is not exhaustive and it is also empirical because it derives from our observation on the behaviors of users in Reddit. However, we feel that it is useful to provide a first definition of three macro-categories of author stereotypes possibly interesting for application scenarios.

Analogously to what we have done for subreddit stereotypes, we have defined two possible orthogonal taxonomies, namely:

- the number of posts: the possible classes are few posts and many posts;
- the number of comments: the possible classes are few comments and many comments.

Afterwards, we determined the following stereotypes characterizing the "very positive" authors, proceeding in a way analogous to the one we adopted for subreddit stereotypes:

- *Unsuccessful Author*: she submits posts but she is never capable of stimulating interactions with other authors.
- *Fame Seeker*: she submits (and/or she is still submitting) an impressive amount of posts in order to reach fame in Reddit.
- *Cringe / NSFW Author*: she often submits cringe / NSFW posts.
- *FBG Publisher* (Few But Good Publisher): she does not publish a very high number of posts; however, her posts are generally appreciated by other users.
- *Content Creator*: she creates and submits contents for people.
- *Successful Author*: she submits many posts that receive many positive comments and are appreciated by other users.
- *Reposter*: she simply re-submits posts of other authors.

Based on the two taxonomies that we defined above, the previous stereotypes can be classified as shown in Table 3.9.

After the "very positive" authors, we focused on the opposite macro-category of author stereotypes, i.e., the "very negative" ones. We obtained the following stereotypes, applying the same reasoning and performing the same tasks that we made for "very positive" authors:

	Few Posts	**Many Posts**
Few Comments	Unsuccessful Author	Fame Seeker Cringe / NSFW Author
Many Comments	FBG Publisher Content Creator	Successful Author Reposter

Table 3.9: Classification of the stereotypes concerning "very positive" authors

- *Unsuccessful Author*: this stereotype is the same as we have seen for "very positive" authors.
- *Spammer*: she is an author submitting a lot of spam posts evaluated negatively by other users.
- *Hatred Sower*: she is a user whose goal is attacking minority groups with hate posts or comments.
- *Instigator*: she is an author using every opportunity to make herself known. For her, it is not important how she is judged, but the fact that one speaks of her.

Based on the two taxonomies defined above, the previous stereotypes can be classified as shown in Table 3.10.

	Few Posts	**Many Posts**
Few Comments	Unsuccessful Author	Spammer
Many Comments	Hatred Sower	Instigator

Table 3.10: Classification of the stereotypes concerning "very negative" authors

After having analyzed the stereotypes belonging to the two extreme categories, i.e., "very positive" and "very negative" authors, we decided to investigate "neutral" authors as representative of a third macro-category, intermediate between the two previous ones. We obtained the following stereotypes, applying the same reasoning and tasks that we made for the other two macro-categories:

- *Unsuccessful Author* and *Fame Seeker*: these stereotypes are the same ones we have seen for the previous macro-categories.
- *PP Author* (Private Purpose Author): she often creates subreddits for private purposes, for instance to talk about specific topics of interest for a particular community. Often, her subreddits require an invitation for being accessed.
- *Bot*: it is a bot; it can be recognized because it always submits similar posts consisting of links and comments with links.
- *Moody Author*: she creates subreddits and submits posts whose topics, expressed positions, and evaluations apparently swing without a logic.

- *Comment Grabber*: she occasionally submits posts capable of stimulating a debate, even if minimal.
- *Big Comment Grabber*: almost all the posts submitted by her stimulate a debate.

Based on the two taxonomies defined above for authors, the previous stereotypes can be classified as shown in Table 3.11.

	Few Posts	**Many Posts**
Few Comments	Unsuccessful Author	Fame Seeker Bot
Many Comments	PP Author Comment Grabber	Moody Author Big Comment Grabber

Table 3.11: Classification of the stereotypes concerning "neutral" authors

3.2 Results

3.2.1 Evaluating author assortativity

In the past, assortativity has been largely analyzed in several social media [109]. In this section, we aim at checking if a form of assortativity exists in Reddit; in particular, we focus on co-posters, i.e., authors submitting posts on the same subreddit.

In order to perform our analyses, we define a support network $\mathcal{P}$, which we call co-post network. Formally speaking:

$$\mathcal{P} = \langle N, E \rangle$$

Here, N is the set of the nodes of $\mathcal{P}$; there is a node $n_i \in N$ for each author a_i who submitted at least one post. There is an edge $(n_i, n_j, w_{ij}) \in E$ if the authors a_i and a_j (associated with the nodes n_i and n_j, respectively) submitted at least one post in the same subreddit. w_{ij} indicates the number of subreddits having at least one post of a_i and, simultaneously, at least one post of a_j.

The number of nodes of $\mathcal{P}$ is equal to the number of authors in our dataset, i.e., 12,464,188. The number of arcs of $\mathcal{P}$ is about 925 billion. The density of this network is 0.00596, whereas the average clustering coefficient is 0.43753.

First of all, we computed the degree centrality of the nodes of $\mathcal{P}$. In Figure 3.19, we report the corresponding distribution. This figure shows that degree centrality follows a power law, even if disturbed. This result is in line with the theory regarding this kind of centrality [613]. The maximum value of degree centrality is 1,820,412, while the minimum value is 0.

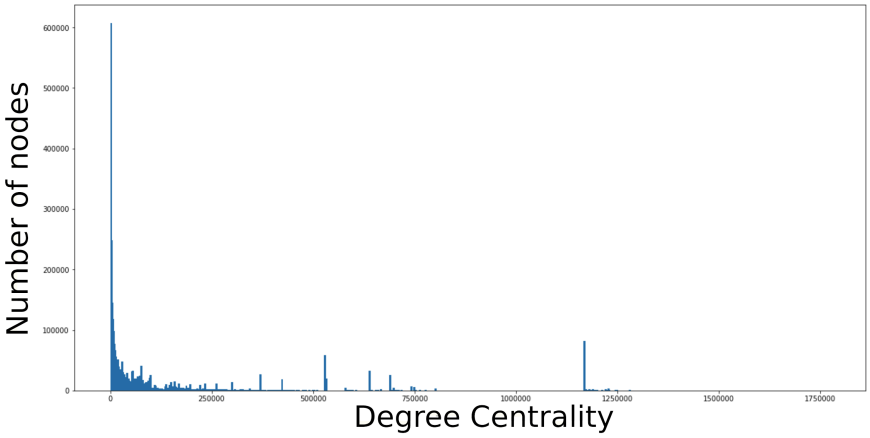
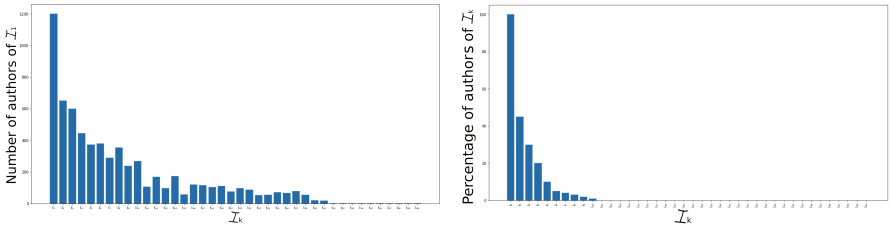


Fig. 3.19: Distribution of degree centrality for the nodes of $\mathcal{P}$

We sorted the corresponding authors in a descending order, based on their degree centrality, to verify the possible presence of a degree assortativity in Reddit. Then, we divided the sorted list into intervals of authors. In particular, we considered equi-width intervals $\{\mathcal{I}_1, \mathcal{I}_2, \cdots, \mathcal{I}_{40}\}$, each consisting of 312,500 authors[1]. As a consequence, the interval $\mathcal{I}_k$, $1 \leq k \leq 39$, contained the authors of the sorted list comprised in the interval $(312,500 \cdot (k-1), 312,500 \cdot k]$, open at left and closed at right. The interval $\mathcal{I}_{40}$ contained the authors comprised in the interval $(12,187,500\ ,\ 12,464,188]$.

First of all, we considered the first interval $\mathcal{I}_1$ and, for each interval $\mathcal{I}_k$, $1 \leq k \leq 40$, we determined how many authors of $\mathcal{I}_1$ are connected to at least one author of $\mathcal{I}_k$. The results obtained are reported in Figure 3.20(a). Then, we computed the percentage of authors of $\mathcal{I}_k$ connected with at least one author of $\mathcal{I}_1$. The results obtained are reported in Figure 3.20(b). From the analysis of Figure 3.20, it is clear that a strict correlation (i.e., a sort of backbone) exists among the authors with the highest degree centrality.

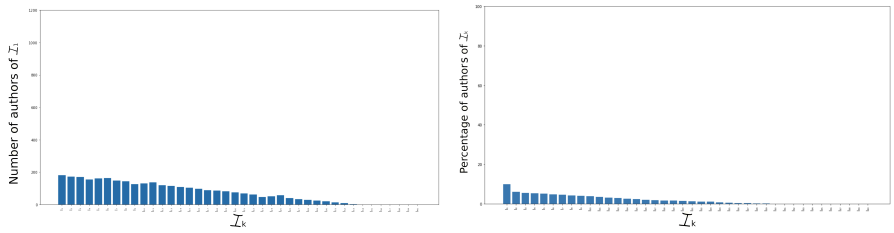


(a) **(b)**

Fig. 3.20: (a) Number of authors of $\mathcal{I}_1$ connected to at least one author of $\mathcal{I}_k$ - (b) Percentage of authors of $\mathcal{I}_k$ connected to at least one author of $\mathcal{I}_1$

[1] Actually, the last interval had a width slightly lower than the other ones.

In order to prove the statistical significance of our results, we generated a null model to compare our findings with the ones obtained in an unbiasedly random scenario. Specifically, we built our null model shuffling the arcs of $\mathcal{P}$ (that, in our case, represent co-posting relationships) among the nodes of this network. In this way, we left unchanged all the original features of $\mathcal{P}$ with the exception of the distribution of co-posting tasks, which became unbiasedly random in the null model. After that, we repeated the previous analyses on the null model. The results obtained are reported in Figure 3.21. Comparing this figure with Figure 3.20, we can see that the distributions represented therein are similar, in a way that many of the intervals with the highest values in Figure 3.20 continue to reach the highest values in Figure 3.21. However, in this last case, the values are much smaller. Therefore, we can conclude that the behavior observed in Figure 3.20 (and the consequent possible degree assortativity revealed by them) is not random but it is intrinsic to Reddit.



(a) **(b)**

Fig. 3.21: (a) Number of authors of $\mathcal{I}_1$ connected to at least one author of $\mathcal{I}_k$ in the null model - (b) Percentage of authors of $\mathcal{I}_k$ connected to at least one author of $\mathcal{I}_1$ in the null model

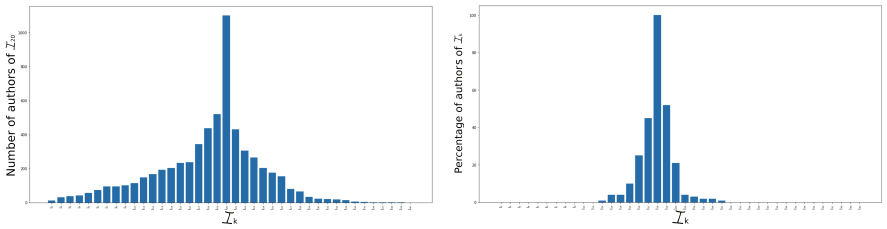
However, this is not sufficient to conclude that there is a degree assortativity for authors in Reddit. In fact, we must check if this trend is also confirmed for the authors with an intermediate degree centrality and for those with a low degree centrality.

Clearly, for an exhaustive analysis, we should repeat the tasks we have previously done for $\mathcal{I}_1$ for all intervals. Due to space constraints, we limit our analysis to the interval $\mathcal{I}_{20}$, representative of intermediate degree centrality intervals, and $\mathcal{I}_{39}$, representative of the low degree centrality intervals[2].

Figure 3.22(a) reports the number of authors of $\mathcal{I}_{20}$ connected to at least one author of $\mathcal{I}_k$, whereas Figure 3.22(b) shows the percentage of authors of $\mathcal{I}_k$ connected

[2] We did not choose $\mathcal{I}_{40}$ because the number of its authors is less than the ones of the other intervals.

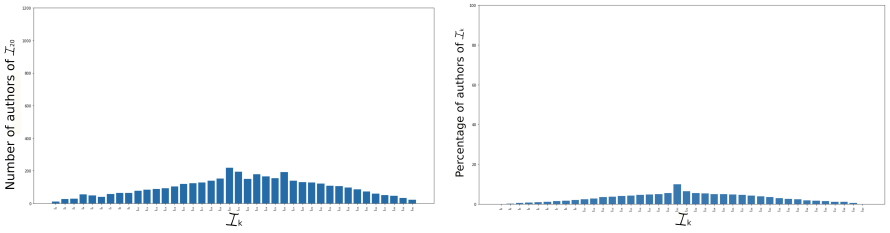
with at least one author of $\mathcal{I}_{20}$. From the analysis of this figure, it emerges a strict correlation between the authors with an intermediate degree centrality.



(a) **(b)**

Fig. 3.22: (a) Number of authors of $\mathcal{I}_{20}$ connected to at least one author of $\mathcal{I}_k$ - (b) Percentage of authors of $\mathcal{I}_k$ connected to at least one author of $\mathcal{I}_{20}$

Also in this case, we compared these findings with the ones obtained in the null model. These last ones are reported in Figure 3.23. Looking at these results and the ones represented in Figure 3.22, we can conclude that, again, the behavior observed in these last figures is not random but it is a property of Reddit.

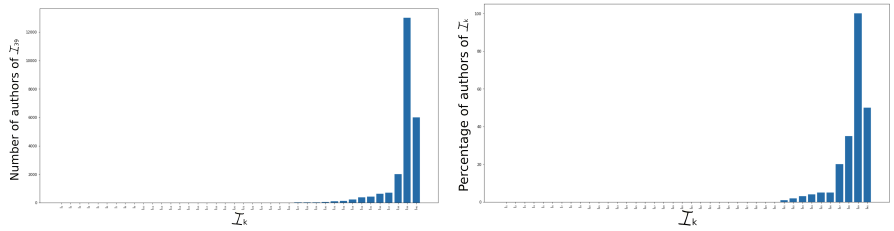


(a) **(b)**

Fig. 3.23: (a) Number of authors of $\mathcal{I}_{20}$ connected to at least one author of $\mathcal{I}_k$ in the null model - (b) Percentage of authors of $\mathcal{I}_k$ connected to at least one author of $\mathcal{I}_{20}$ in the null model

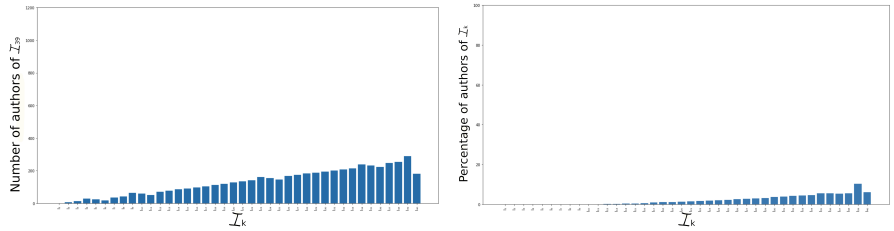
Finally, Figure 3.24(a) reports the number of authors of $\mathcal{I}_{39}$ connected to at least one author of $\mathcal{I}_k$, whereas Figure 3.24(b) shows the percentage of authors of $\mathcal{I}_k$ connected with at least one author of $\mathcal{I}_{39}$. Again, there is a strict correlation between authors with a low degree centrality. Also for this last case, we compared the results obtained with the ones returned using the null model. We report these last ones in Figure 3.25. The comparison of these figures confirms that the behavior observed in them is a property intrinsic to Reddit.

Having verified that there exists a sort of backbone among the authors with a high (resp., intermediate, low) degree centrality, we can conclude that actually Reddit is



(a) **(b)**

Fig. 3.24: (a) Number of authors of $\mathcal{I}_{39}$ connected to at least one author of $\mathcal{I}_k$ - (b) Percentage of authors of $\mathcal{I}_k$ connected to at least one author of $\mathcal{I}_{39}$



(a) **(b)**

Fig. 3.25: (a) Number of authors of $\mathcal{I}_{39}$ connected to at least one author of $\mathcal{I}_k$ in the null model - (b) Percentage of authors of $\mathcal{I}_k$ connected to at least one author of $\mathcal{I}_{39}$ in the null model

assortative with respect to degree centrality, as far as the co-posting relationship is concerned.

This important result can be explained considering the concept of karma and the posting rules in Reddit. Indeed, in this platform, each user has associated a karma, which is a score taking her past "reputation" into account. Generally, users with high karma are very active and, often, submit a lot of appreciated posts. As a consequence, it is presumable that they have a high degree centrality. In other words, a direct correlation between karma and degree centrality can be recognized for authors. Now, the posting rules of Reddit state that each subreddit has associated a minimum threshold of karma [437, 449, 37] so that only the authors with a karma higher than this threshold can submit a post on it. This threshold is dynamic and changes over time. Clearly, when it is low, all the authors can submit their posts on the subreddit. When it grows, the authors with a low karma (and, presumably, with a low degree centrality) cannot submit posts on it. Finally, when it becomes high, only the authors with a high karma (and, presumably, a high degree centrality) can submit posts on it. This way of proceeding tends to segment users into groups having homogeneous degree centralities.

Having verified the assortativity of Reddit with respect to degree centrality, it is natural to wonder whether this property depends on the type of centrality or is intrinsic in this social platform. As a premise to this investigation, it is worth underlying that each form of assortativity is a unique history *per se*. Therefore, it is impossible to define a general rule. Nevertheless, it is possible to verify if a trend exists, and we have operated in this direction.

To this end, we have chosen a second form of centrality (i.e., the eigenvector centrality) and we have repeated for it all the steps previously seen for degree centrality. The results obtained are shown in Figures 3.26 - 3.28

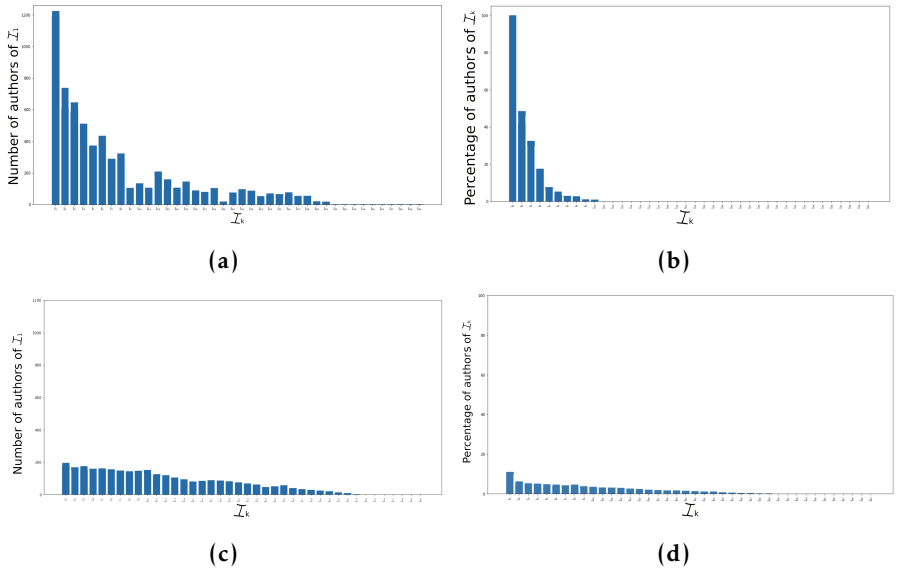
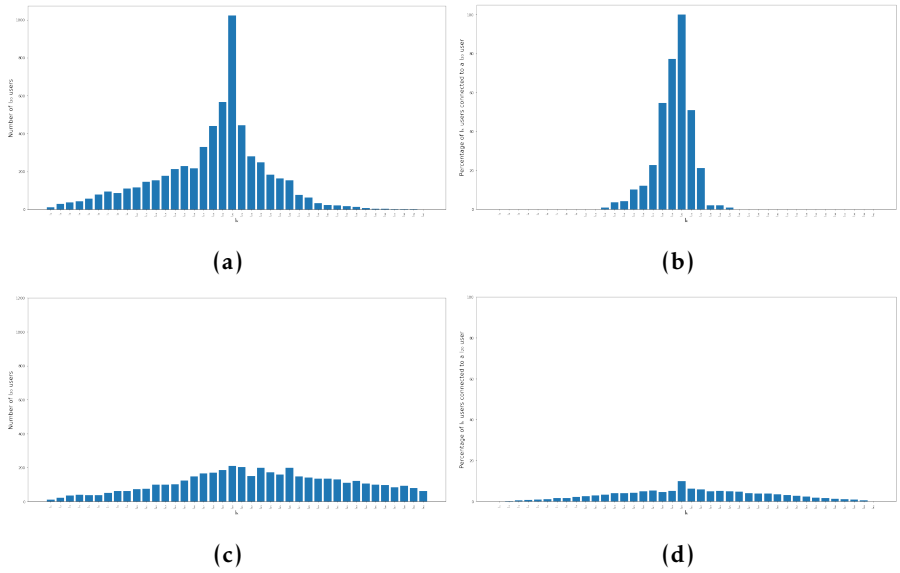


Fig. 3.26: (a) Number of authors of $\mathcal{I}_1$ connected to at least one author of $\mathcal{I}_k$ - (b) Percentage of authors of $\mathcal{I}_k$ connected to at least one author of $\mathcal{I}_1$ - (c) Number of authors of $\mathcal{I}_1$ connected to at least one author of $\mathcal{I}_k$ in the null model - (d) Percentage of authors of $\mathcal{I}_k$ connected to at least one author of $\mathcal{I}_1$ in the null model

They confirm that there is an assortativity among the authors of Reddit also with respect to the eigenvector centrality. As a consequence, we can conclude that the assortativity of Reddit authors is not limited to degree centrality but represents a trend characterizing this social platform beyond the form of centrality taken into consideration.

3.2.2 Correlation between subreddits and author stereotypes

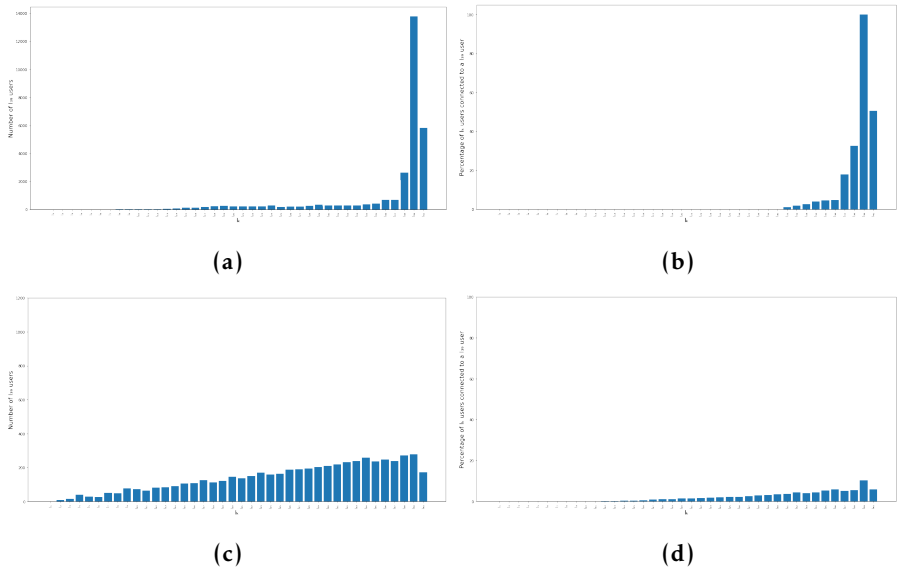
First of all, we observe that, although in principle subreddit stereotypes and author stereotypes are two orthogonal concepts, in practice there are strong correlations be-



(a) **(b)**

(c) **(d)**

Fig. 3.27: (a) Number of authors of $\mathcal{I}_{20}$ connected to at least one author of $\mathcal{I}_k$ - (b) Percentage of authors of $\mathcal{I}_k$ connected to at least one author of $\mathcal{I}_{20}$ - (c) Number of authors of $\mathcal{I}_{20}$ connected to at least one author of $\mathcal{I}_k$ in the null model - (d) Percentage of authors of $\mathcal{I}_k$ connected to at least one author of $\mathcal{I}_{20}$ in the null model



(a) **(b)**

(c) **(d)**

Fig. 3.28: (a) Number of authors of $\mathcal{I}_{39}$ connected to at least one author of $\mathcal{I}_k$ - (b) Percentage of authors of $\mathcal{I}_k$ connected to at least one author of $\mathcal{I}_{39}$ - (c) Number of authors of $\mathcal{I}_{39}$ connected to at least one author of $\mathcal{I}_k$ in the null model - (d) Percentage of authors of $\mathcal{I}_k$ connected to at least one author of $\mathcal{I}_{39}$ in the null model

tween them. In fact, certain subreddit stereotypes are the ideal and perfectly tailored places for certain user stereotypes, and vice versa.

Let us now examine these correlations more closely. In the following of this section, for more clarity and to avoid heavy speech, we use the Successful Subreddit notation to indicate the name of a subreddit stereotype, whereas we adopt the *Successful Author* notation to denote an author stereotype.

User Profile is a fairly generic subreddit stereotype and can be related, at least partially, to various author stereotypes. Surely, a *Fame Seeker* can create a User Profile subreddit to advertise her profile. A similar argument probably applies to a *Content Creator* and a *Successful Author*.

Unsuccessful Subreddit could be at least partially related to *Unsuccessful Author* because if a subreddit was not successful then its posts did not attract Reddit users. Clearly, the authors of those posts, if this fact happens several times, would tend to become unsuccessful authors.

Clearly, there are very strong and direct correlations between Comment Grabber and the homonymous author stereotype, between Big Comment Grabber and *Big Comment Grabber*, between Private Community and *PP Author*, between Bot and the homonymous author stereotype, and between Cringe / NSFW Subreddit and *Cringe / NSFW Author*.

There is at least a partial relationship between Banned Subreddit and *Spammer* and *Hatred Sower*, because it is very likely that subreddits with many authors of those two categories are banned. Similarly, there is a correlation between Successful Subreddit and *Successful Author*; in fact, it is likely that if many successful authors write in a subreddit, then that subreddit will be successful.

A less obvious, but extremely interesting correlation exists between Niche Subreddit and *FBG Publisher*.

Again, Unsuccessful Boomer may be related to *Fame Seeker*, *Cringe / NSFW Author*, *Hatred Sower* or *Investigator*. In all these cases, the authors of these subreddits may have initially succeeded in stimulating the attention of other Reddit users but, after a while, this attention was lost.

Finally, there is a quite evident correlation between Unsuccessful Zombie and *Unsuccessful Author*, in the sense that if an author activates subreddits that become Unsuccessful Zombie, in the long run she risks to become an *Unsuccessful Author*. Finally, Unsuccessful Zombie could have a slightly subtler and hidden correlation with *Moody Author* because, if in a subreddit many posts of moody authors are published, it is likely that this subreddit will not attract people and eventually will become an Unsuccessful Zombie.

3.2.3 Considerations about author stereotypes and assortativity

After having examined the correlation between subreddit stereotypes and author stereotypes, we continue our discussion by examining the correlations between the results obtained for author stereotypes and those concerning assortativity. In Section 3.2.1, we found that there is a degree (resp., eigenvector) assortativity between Reddit authors. This implies that authors with similar degree (resp., eigenvector) centrality tend to form a backbone. Keeping in mind the definition and properties of these two forms of centrality, it is possible to make some interesting deductions.

The first one is that *Fame Seekers*, who generally have a high degree centrality, tend to form a backbone and, therefore, to support each other. An analogous reasoning can be imagined for *Successful Authors* and *Reporters*, who are also characterized by a very high degree centrality. Continuing in this direction, even many authors characterized by negative stereotypes tend to support each other; in particular, this happens for *Spammers*, *Hatred Sowers* and *Investigators*. In these cases, a post published by one of them tends to provoke the reaction of the others, giving rise to very long discussions that often involve a huge number of people. A similar situation, even if with a neutral and not negative connotation, can concern the *Big Comment Grabbers*. Even these authors tend to form communities in which large discussions take place; however, unlike the previous cases, these discussions are not necessarily harmful.

As far as eigenvector centrality is concerned, in addition to all the communities mentioned above, the presence of backbones between *FBG Publishers* or *Content Creators* appears possible. In fact, these authors, who tend to use Reddit as a utility tool, may be strongly attracted by subreddits created by authors with the same intentions and, therefore, may tend to form communities. It is interesting to highlight that these types of figures (a sort of "grey cardinals") are the classical ones having a high eigenvector centrality and, as far as we are concerned, a high eigenvector assortativity.

A final discussion concerns the results on assortativity described in this chapter and the ones on assortativity in social networks described in the past literature. As previously pointed out, Newman's seminal work showed that social networks are generally assortative, unlike other types of networks, such as technological and biological ones, which are disassortative [462].

Next, the authors of [17] demonstrated that: *(i)* Cyworld is slightly disassortative with respect to degree centrality on a network built taking users and their friendships into account, while it is strongly assortative with respect to degree centrality on a network built considering users and the "testimonial" relationships (a kind of

relationship specific of this social network) existing between them; *(ii)* Orkut is assortative with respect to degree centrality on a network built starting from users and their friendships; *(iii)* MySpace is neutral (that is neither assortative nor disassortative) with respect to degree centrality on a network that takes users and their friendships into account.

The authors of [107] showed that Twitter is strongly assortative with respect to degree centrality on a network that takes the sharing of interest among users into account. Furthermore, the authors of [109] studied assortativity in Facebook and showed that such a social network is assortative with respect to the tendency of a bridge (i.e., a user joining more social networks) to communicate with other bridges.

Finally, in [293], the authors considered Reddit and investigated the concept of assortativity but for a very particular aspect, i.e., loyal communities. In particular, they showed that loyal communities are not assortative with respect to the activity level of the users belonging to them, while assortativity exists in the case of non-loyal communities. The lack of assortativity in loyal communities implies that users belonging to them are willing to communicate with all the other users of the same community, regardless the corresponding activity level. By contrast, the presence of assortativity in non-loyal communities implies that the corresponding users tend to partition themselves into subgroups based on their activity level. Indeed, a user with a certain activity level tend to communicate only with users having similar activity levels.

As said before, we want to provide a contribution in the study of assortativity in social networks. First, besides degree centrality, it also considers eigenvector centrality. Furthermore, it focuses on the study of assortativity in Reddit, a social platform that was not analyzed in the past as far as this feature is concerned, except for the investigations described in [293]. However, in this last paper, the main topic of the author investigation was not assortativity but loyalty, while assortativity simply served as a feature to assess whether loyal and non-loyal communities could be partitioned into smaller groups. Therefore, compared to the general studies on assortativity presented in [17, 107, 109], the analysis of [293] can be considered of niche. As a proof of this, we can observe that, contrary to all studies on assortativity proposed in the past, in [293] the presence of assortativity among the nodes of a network is seen as a negative factor (leading highly active users to disregard little active and new ones), rather than a positive feature.

Compared to [293], our approach aims at bringing the study of assortativity into Reddit in the general mainstream of the study of assortativity in social networks, analyzing this feature by itself, independently from other features, such as loyalty. As a matter of fact, the results we found are in line, and even strengthen, the trends

on assortativity in social networks hypothesized by Newman and next found by most of the other authors.

3.2.4 Applications of stereotypes

This section presents two possible applications of the stereotypes previously investigated. The first regards the usage of subreddit stereotypes to make a subreddit successful. The second concerns the exploitation of particular types of author stereotypes to improve the content quality of subreddits.

Application of subreddit stereotypes

In Section 3.1.2, we defined several subreddit stereotypes belonging to three macrocategories, namely "dead in crib", "survivors" and "undelivered promises". A first application of this research can be the definition of some guidelines to follow in order to make a subreddit successful. Indeed, knowing how a subreddit became successful (resp., unsuccessful) can lead to the characterization of "positive" (resp., "negative") actions that can influence the "lifespan" of a new subreddit. For instance, consider the subreddit /r/meme. It started during 2008 and, at the time of writing, has about 806,000 users. Certainly, it represents an example of a successful subreddit. Here, the authors post high quality and engaging contents. This kind of behavior could be registered as a "best practice" in the guidelines. On the other hand, a subreddit containing only few contents from few authors is an example of an unsuccessful subreddit. This failure could be caused by a lack of engaging contents posted in it. Clearly, what said above provides just an idea of what these guidelines could contain.

Another possible application of subreddit stereotypes could regard the definition and realization of recommender systems for Reddit. These systems would aim at recommending to a user subreddits with the same stereotype (or the same content) as the ones characterizing the subreddits accessed by her in the past. In any case, the recommender system should avoid "dead in crib" subreddits or, more generally, unsuccessful ones. On the other hand, the same system should suggest to a user successful subreddits, subreddits currently expanding their community and/or subreddits characterized by contents in line with her profile.

A further example of possible usage of subreddit stereotypes could be the definition of an algorithm that finds subreddits to merge or, at least, to integrate. For instance, consider two zombie subreddits with related topics, where authors are posting contents that were not able to attract other users. These two subreddits are surviving, but their interactions with users are so low that they can actually be considered dead. If they would be merged or integrated into a unique subreddit, they

could have more chances of becoming successful. Joining together two, or even more, subreddits having the same (or related) topics/characteristics brings more visibility and more contents to them. These contents would be, otherwise, dispersed in different unsuccessful subreddits. Even if the new integrated subreddit is made up of past zombies, it could become so successful to attract authors and co-posters from other communities.

Application of author stereotypes

In Section 3.1.3, we defined some possible author stereotypes. Some of them are strictly related to the homonymous or corresponding subreddit stereotypes. Other ones, instead, are intrinsic to human behavior and, in particular, to the concept of author. For example, consider "Fame Seekers" and "Content Creators". These users could represent the target of a proposal of an advertising campaign aiming at promoting them. Take, for instance, a painter or a digital artist, who has been classified as "Fame Seeker". An advertising company can easily persuade her to give it an engagement to promote her image.

Another possible usage of author stereotypes is the definition and implementation of different categories of recommender systems. A first category could help bootstrapping a subreddit. Consider, for instance, a newborn subreddit where authors post comics strips created by them. Knowing successful authors of comics strips and being able to convince them to become "Content Creators" in the new subreddit could help this last one to get visibility. Complementary to this case, a second category of recommender systems could be used for talent scouting. In this case, a "Fame Seeker", who is also a creator of comics strips, could be recommended to successful subreddits if her contents are high-quality ones.

The last application we present in this overview is the definition of an algorithm that builds blacklists of users based on author stereotypes. As an example, we can define a "dangerousness level" of an author for one subreddit, a set of subreddits or all subreddits. For instance, in such a scenario, "Hatred Sowers" can be automatically banned from subreddits attended by sensitive people. This way of proceeding could certainly maintain the discussion in these subreddits clean, thus avoiding their visitors being harassed by fake news and cyberbullying.

4

Detecting backbones of information diffusers among different communities of a social platform

Information diffusion in social networks is a classic and, at the same time, very current problem. In fact, information diffusers are always looking for new techniques to disseminate information of their interest by creating backbones among them. In this chapter, we focus on a specific, but very current and relevant, scenario regarding this way of proceeding. In fact, we propose an approach for the detection of possible backbones of information diffusers among different communities of a social network. Our approach is based on a new centrality measure that we call disseminator centrality. It is specifically designed to detect the so-called disseminator bridges, i.e., users belonging to multiple communities of a single social network, who want to disseminate information of their interest from one community to another by supporting each other. This paper describes the proposed approach, presents the disseminator centrality, illustrates the differences with respect to the related literature and presents the results of the experiments carried out to evaluate its performance.

The material presented in this chapter was derived from [257].

4.1 Methods

4.1.1 Network model

In this section, we define the network model on which our approach is based. Let S be a set of communities (intended as groups of users having common interests - e.g., subreddits or Facebook groups) and let $s \in S$ be a community of S. Let U be the set of users who had at least one interaction with at least one community of S. We denote by U_x^s the set of users who had at least one interaction of type x with s. Currently, $x \in \{p, c\}$; p denotes the posting activity while c represents the commenting activity. Regarding this, it should be pointed out that a comment can refer to a post or to another comment published previously. In the future, x can be extended if we want to consider other types of user-community interactions. We define by U_x the set of all users who had at least one interaction of type x with at least one community of S:

$$U_x = \bigcup_{s \in S} U_x^s \tag{4.1}$$

Our model consists of a network:

$$\mathcal{G}_x = \langle V_x, E_x \rangle \tag{4.2}$$

V_x is the set of nodes of $\mathcal{G}_x$; there is a node $v \in V_x$ for each user $u \in U_x$. Since there is a biunivocal correspondence between a node $v \in V_x$ and a user $u \in U_x$, in the following we will use these two terms interchangeably and, with a little abuse of notation, we will write $\mathcal{G}_x = \langle U_x, E_x \rangle$.

E_x is the set of edges of $\mathcal{G}_x$. An edge $(u_i, u_j, w_{ij}) \in E_x$ denotes that u_i and u_j performed at least one interaction of type x in the same community at least once. w_{ij} represents the number of times this fact happened.

Starting from $\mathcal{G}_x$, it is possible to define two networks, namely $\mathcal{G}_p$, which models all users who published posts, and $\mathcal{G}_c$, which represents all users who made comments.

Finally, it is possible to define the network $\mathcal{N} = \langle U, E \rangle$, where U is the set of users defined above and $E = E_p \cup E_c$. From its definition, we can see that $\mathcal{N}$ is a sort of "merge" of $\mathcal{G}_p$ and $\mathcal{G}_c$.

4.1.2 Detection of a backbone of information diffusers

After introducing a network model capable of representing our reference context, in this section we use it for the definition of the backbone of information diffusers, which we will call "disseminator bridges".

A first definition of the concept of disseminator bridge. The concept of disseminator bridge is very articulate, rich and complex. In this section, we introduce a first definition of it, based on concepts already existing in the literature.

Starting from the network $\mathcal{N}$ introduced above, we say that a node $u \in U$ is a disseminator bridge if:

- it is directly connected to many neighboring nodes in $\mathcal{N}$, which implies that the associated user can interact with many other ones;
- it is connected to other nodes in $\mathcal{N}$ through short paths, which implies that the posts and comments of the corresponding user reach most of the other users through few interactions;
- it is located in many paths of $\mathcal{N}$, which implies that the corresponding user is a key node for reaching the users of $\mathcal{N}$.

In Social Network Analysis, the previous three hypotheses can be modeled by means of the concept of centrality. In fact, they are equivalent to saying that a user $u \in U$ is a disseminator bridge if she has high degree, closeness and betweenness centralities in $\mathcal{N}$.

To formally define the set of disseminator bridges, we start with the definition of the list L^{dc} (resp., L^{cc}, L^{bc}); it is obtained by sorting the users of U in an ascending order against their degree (resp., closeness, betweenness) centrality.

Afterwards, we define the operator $\pi(L, u)$, which receives an ordered list L of users and a user u and returns the position of u in L.

Now, for each user, we combine the three previous centralities in a "combined centrality" as follows:

$$CombC(\mathcal{N}, u) = \pi(L^{dc}, u) + \pi(L^{cc}, u) + \pi(L^{bc}, u) \tag{4.3}$$

$CombC(\mathcal{N}, u)$ ranges in the integer interval $[0, 3 \cdot (|U| - 1)]$; the higher its value the greater the ability of u to be an information diffuser.

After this, we can define the list L^{CombC} obtained by sorting the users of U in a descending order against the values of the operator $CombC(\mathcal{N}, u)$.

Finally, we can define the set $\hat{U}_Y$ of the disseminator bridges as:

$$\hat{U}_Y = \left\{ b \mid b \in Top(L^{CombC}, Y) \right\} \tag{4.4}$$

where $Top(L, Y)$ returns the first Y% elements of the list L.

Clearly, our definition is parametric with respect to the value of Y. A high value of this parameter allows for the selection of many users and should be considered in scenarios where it is believed that there are many active information disseminators. Conversely, a low value of Y allows the selection of few users and is particularly suitable in scenarios where it is thought that there are few information disseminators who, alone, can convey the information of their interest.

A refined definition of the concept of disseminator bridges. In the previous section, we introduced the concept of disseminator bridge and provided a definition based on three classic centrality measures already existing in Social Network Analysis. Each of these measures captures a prerogative of the disseminator bridge. However, we believe that this concept is even more complex and richer than what emerges from the previous definition. Therefore, in this section, we propose a new definition of disseminator bridge that is more articulated and holistic. It will lead us to the definition of a new centrality measure, which we call *disseminator centrality*. It represents the second main contribution of this paper. In Section 4.2, we compare this new centrality measure with the ones used in the previous definition and show

how it is actually able to capture the concept of disseminator bridge better than the classic centralities and their combination. Thanks to this new centrality measure, it is possible to identify the disseminator bridges in a network.

The new definition of disseminator bridge assumes that a user's ability to be an information disseminator is directly related to:

- the number of posts and comments she submitted;
- the number of communities she reached at least once through a post or comment published by her;
- her ability to equidistribute her posting and commenting activity across communities.
- the fact that she is not a spammer or an author of junk posts and/or comments.

The third and fourth properties deserve a more in-depth comment. As for the third one, the basic idea is that if a user publishes posts and comments in a set S of communities, but almost all her publishing activity is concentrated on only one community, then her impact is very high in the latter but extremely marginal in the other $|S|-1$ ones. Instead, if a user publishes homogeneously in a set S of communities, her impact is good in all of them and overall she has a higher impact on the reference social platform than in the previous case.

To evaluate the equidistribution of the communities in which a user publishes, we adapt the Herfindahl-Hirschman Index (HHI) [304] to our context. This index has been widely used in various fields of economics research for several decades. For example, it has been adopted to evaluate the concentration ratio in a certain market. In this case, it is defined as $H = \sum_{i=1}^{N} s_i^2$, where N is the number of firms operating in the market and s_i is the market share of the i^{th} firm. H ranges in the real interval $[\frac{1}{N}, 1]$; the higher H, the higher the concentration rate in that market.

The adaptation of the HHI to our case is done as follows. Let $u \in U$ be a user and let $s \in S$ be a community. Finally, let $\nu(u,s,x)$ be the number of interactions of type x, $x \in \{p,c\}$, that u performed in s. The Herfindahl-Hirschman Index H_x^u of u in S relative to x can be defined as:

$$H_x^u = \sum_{s=1}^{|S|} \left(\frac{\nu(u,s,x)}{\sum_{s=1}^{|S|} \nu(u,s,x)} \right)^2 = \frac{1}{\left(\sum_{s=1}^{|S|} \nu(u,s,x) \right)^2} \cdot \sum_{s=1}^{|S|} \nu(u,s,x)^2 \tag{4.5}$$

Regarding the fourth property, it is necessary to prevent the disseminator centrality of a user from being high due to the fact that she publishes junk/spam posts and comments on many social networks. To prevent this, it is first necessary to recognize such users and then penalize them. A variety of approaches for recognizing spammers in social platforms have been proposed in the literature; therefore, it is possible

to adopt one of them in our context. For example, one could adopt the approach described in [426], which has an accuracy of 98% with only 2% false positives.

Once we have introduced H_x^u, we are able to define the disseminator centrality. To do this, we introduce the following support lists:

- L^P (resp., L^C) is the list obtained by sorting the users of U in an ascending order with respect to the number of posts (resp., comments) they submitted. All users who are recognized as spammers are put at the top of the list regardless of the number of posts and comments they published.
- L^{PC} (resp., L^{CC}) is the list obtained by sorting the users of U in an ascending order with respect to the number of communities in which they published at least one post (resp., comment). All users who are recognized as spammers are put at the top of the list regardless of the number of posts and comments they published.
- L^{PH} (resp., L^{CH}) is the list obtained by sorting the users of U in a descending order with respect to the corresponding Herfindahl-Hirschman Index H_p^u (resp., H_c^u).

It is worth pointing out that this way of proceeding could penalize a user if she is erroneously recognized as a spammer. However, if this happens for only one of the six lists the penalty is very small. Instead, if this happens for three or four lists the penalty would be significant. However, given the accuracy levels of the approach of [426], the possibility of this happening is $(0.02)^3 = 0.0008\%$ (resp., $(0.02)^4 = 0.000016\%$) in case of the simultaneous presence of the same user as a false positive in 3 (resp., 4) lists. These numbers are extremely low and allow us to conclude that the benefits in a system capable of blocking the authors of spam/junk posts and comments are much greater than the risks associated with the misclassification of users.

Based on these lists, the disseminator centrality $DC(\mathcal{N}, u)$ of a user u in a network $\mathcal{N}$ is defined as:

$$DC(\mathcal{N}, u) = \pi(L^P, u) + \pi(L^C, u) + \pi(L^{PC}, u) + \pi(L^{CC}, u) + \pi(L^{PH}, u) + \pi(L^{CH}, u) \quad (4.6)$$

Here, the operator $\pi(L, u)$ has been introduced in the previous section. $DC(\mathcal{N}, u)$ ranges in the integer interval $[0, 6 \cdot (|U| - 1)]$; the higher its value the greater the ability of u to be an information diffuser.

In our definition of disseminator centrality, regarding the fifth and sixth components in Equation 4.6, there is an implicit assumption that we make, which is that the communities considered in our platform deal with similar topics in most cases.

For example, as we will see in the experiments described in Section 4.2, the communities could cover all the subreddits of Reddit dealing with topics related to COVID-19. If this implicit assumption is true, the fifth and sixth components conceptually provide a valuable input in identifying disseminator bridges, and thus in defining disseminator centrality. If this assumption is false, two categories of users would be favored, namely: *(i)* users who are highly active and equidistribute their posts and comments across communities; *(ii)* users who are low active but submit their posts and comments equally across communities. Now, the contribution of the fifth and sixth lists must be still supplemented with that of the other four lists. When such integration is done, the first category of users would still be favored while the second one would be penalized in any case. Thus, we can conclude that, in cases in which our implicit assumption is false, the overall definition of disseminator centrality is such as to limit potential errors as much as possible.

Note that the definition of $DC(\mathcal{N}, u)$ is completely modular. Therefore, if we want to consider additional features, it is sufficient to associate an ordered list with each new feature and insert the operator π associated with that list in the formula of $DC(\mathcal{N}, u)$.

After introducing the disseminator centrality $DC(\mathcal{N}, u)$ of u in $\mathcal{N}$, we can define the list L^{DC} obtained by sorting the users of U in a descending order against the values of $DC(\mathcal{N}, u)$.

Finally, we define the set $\tilde{U}_Y$ of the disseminator bridges similarly to what we have done for $\hat{U}_Y$. More specifically:

$$\tilde{U}_Y = \left\{ b \mid b \in Top(L^{DC}, Y) \right\} \tag{4.7}$$

where $Top(L, Y)$ is the operator introduced in the previous section.

In Section 4.2, we illustrate the various positive features of this new centrality through a series of tests. These concern not only accuracy but also efficiency. In fact, combined centrality is based on the use of some centralities that require path computation (in particular, closeness and betweenness centralities). Such calculation is computationally heavy, and it is well known by social network analysts that closeness and betweenness centralities are difficult to be obtained for networks of a certain size. In contrast, disseminator centrality is essentially based on sorting a set of lists, which, as we know, is a problem with a much less computational complexity, i.e. $O(n \cdot log(n))$, where n is the number of nodes in the network. The computations required to form such lists are negligible in case of L^P, L^C, L^{PC} and L^{CC}. Instead, for what concerns L^{PH} and L^{CH}, it is necessary to calculate the Herfindahl Index, whose value is obtained by combining very elementary calculations such as, for instance, sums of fractions. As a proof of this, when we compute disseminator centrality on a

real dataset (see Section 4.2.4), the computation time needed is more than an order of magnitude less than the one of combined centrality.

Definition of the backbone of disseminator bridges. In the previous sections, we have provided a simple (Section 4.1.2) and refined (Section 4.1.2) version of the set of the disseminator bridges. In this section, we want to introduce the concept of backbone of disseminator bridges. The idea behind this concept is to check if all or part of the disseminator bridges found in a social platform form a structured organization (i.e., a group of people who organize to support each other for achieving a common goal) through which they manage to disseminate as much content as possible to as many users as possible.

Our reasoning for identifying the backbone is as follows. In our model, we consider two kinds of interaction, namely posting and commenting. Regarding this second kind, we can distinguish between the case in which a user comments directly on a post (we call "top-level comment" such a kind of comment) and the one in which a user comments on another comment. For our goal, the first case is much more interesting because it is more likely that, when a disseminator bridge wants to support another one, the former will do so by posting a favorable comment directly to the latter's post.

To assess whether a comment is favorable, we can apply a sentiment analysis based approach, such as VADER [317], to it. VADER receives a text as input and returns a so called compound sentiment value; this is a real number ranging in the interval $[-1,1]$. A value close to 1 (resp., -1) indicates that the text expresses an extremely positive (resp., negative) sentiment. A value close to 0 denotes a neutral sentiment. The compound sentiment value is currently recognized as one of the most useful metrics when a single unidimensional sentiment measure is needed. Moreover, it has already been successfully adopted for examining Reddit posts or comments [308, 342].

An additional reasoning we considered in identifying the existence of a backbone concerns the reciprocity of the support. In fact, if the presence of positive comments to the posts of a disseminator bridge b published by a disseminator bridge b' can be already an indicator of the existence of a backbone, the simultaneous presence of comments to the posts of b' published by b is a much stronger indicator.

To integrate all this reasoning into a formal representation, we introduce a network called *Interaction Network*. Specifically:

$$IN = \langle V_{IN}, E_{IN} \rangle \tag{4.8}$$

There is a node $v_i \in V_{IN}$ for each user $u_i \in U$. Once again, since there is a biunivocal correspondence between a node $v_i \in V_{IN}$ and a user $u_i \in U$, in the following we will use these two terms interchangeably. There is an arc $(u_i, u_j, w_{ij}) \in E_{IN}$ if u_j published at least one top-level positive comment to a post of u_i, and vice versa; w_{ij} indicates the number of times this happened.

After formalizing the Interaction Network, we are able to introduce and formalize the concept of Disseminator Bridge Backbone.

First, based on the previous reasoning, we can assume that the users of the Disseminator Bridge Backbone should belong to the maximum connected component of IN[1]. Based on this assumption, we introduce the function $CC(IN, u_i)$. It receives an Interaction Network IN and a user u_i as input, and returns `true` if u_i belongs to the maximum connected component of IN, `false` otherwise.

Having $CC(IN, u_i)$ at disposal, the Disseminator Bridge Backbone can be defined as:

$$\mathcal{DBB}_Y = \{b \mid b \in \tilde{U}_Y, CC(IN, b) = \texttt{true}\} \tag{4.9}$$

In other words, it consists of those users of U who are disseminator bridges and, at the same time, belong to the maximum connected component of IN.

Discussion on a possible extension of our approach to a Social Internetworking Scenario. In this paper, we focused on a scenario involving a single social platform. However, more and more often we are in presence of situations in which users join multiple social networks and operate simultaneously on them. In the literature, this scenario has already been investigated under various names, e.g., Social Internetworking System [110, **?**] , Multi-Social Network Scenario [428, **?**], etc. In these scenarios, a user registered on multiple social networks is often referred to as a "bridge". In fact, she allows the interaction between users of different social networks who could not communicate otherwise [103]. The user's join to multiple social networks can be explicitly stated by her through the so-called "me edges", or it can be inferred by applying approaches that detect two accounts on different social networks belonging to the same user [111, **?**].

Our approach can easily be extended to such a scenario. Specifically:

- The definition of $\mathcal{G}_x$, as well as that of $\mathcal{N}$ - see Section 4.1.1 - could involve partitioning users (and the corresponding nodes) into two subsets. The first includes users operating on a single social network while the second groups bridges. This could allow bridges to be weighted differently from non-bridges when centrality computation is performed.

[1] Actually, in the next section, we will verify the correctness of this assumption.

- The definition of the Combined Centrality $CombC(\mathcal{N}, u)$ - see Section 4.1.2 - is intrinsically modular. Taking advantage of this feature, one can think of adding a fourth centrality that privileges bridges, and thus the ability of a user to transfer information from one social network to another. An example of such a centrality is bridge centrality [103].
- The definition of the Disseminator Centrality $DC(\mathcal{N}, u)$ - see Section 4.1.2 - is also inherently modular. By exploiting this property, new features, which take into account the posting and commenting activities performed by users on different social networks, can be added to it. Again, with such an expedient, bridges would be favored.
- The definition of the Interaction Network IN, on which the final computation of the Disseminator Bridge Backbone depends - see Section 4.1.2 - requires a node for each user. Similar to what we saw for $\mathcal{G}_x$ and $\mathcal{N}$, we could partition the users involved and the corresponding nodes into bridges and non-bridges and take this partitioning into account when computing connected components. For example, we could assign different weights to the two types of nodes and take those weights into account when computing the maximum connected component. The latter might not necessarily be the one with the maximum number of arcs, but rather the one with the best combination of the number of arcs and the number of bridges.

4.2 Results

In this section, we illustrate the experiments we carried out to evaluate the ability of our approach to detect the possible existence of a backbone of information diffusers. In particular, we describe our dataset in Section 4.2.1. In Section 4.2.2, we present the network $\mathcal{N}$ that we constructed for our experiments. In Section 4.2.3, we compute the set $\hat{U}_Y$ of disseminator bridges by applying the first definition for them, introduced in Section 4.1.2. In Section 4.2.4, we determine the set $\tilde{U}_Y$ of disseminator bridges by applying the second definition for them, introduced in Section 4.1.2. Furthermore, we compare the sets $\hat{U}_Y$ and $\tilde{U}_Y$ to see if and how the second definition of disseminator centrality is better than the first. Finally, in Section 4.2.5, we test our approach to compute the Disseminator Bridge Backbone.

4.2.1 Dataset

In order to construct the dataset for our experiments, we chose Reddit[2] as the reference social medium. The reason for this choice is that Reddit is well suited to test our

[2] www.reddit.com

approach because it is already structured into communities called subreddits. The reference time interval for our dataset is from January 1^{st}, 2020 to June 30^{th}, 2021. To retrieve the data of our interest, we used Pushshift [65], which is one of the main Reddit data repositories available online. We decided to focus our analysis on information diffusers related to a specific theme to avoid biases due to the presence of different themes possibly correlated to each other. The theme we chose was COVID-19 and we considered all the posts and comments published in the reference time interval on the main subreddits (i.e., those with the most users and submissions) dealing with this theme. We chose COVID-19 as reference theme because it is current and because Reddit hosts several subreddits related to it, which, as a whole, provide various points of view on this topic. For all these reasons, the ecosystem of communities dealing with this topic is rich, attractive and well suited for checking the possible presence of disseminator bridges. In Table 4.1, we report the subreddits considered, along with a brief description for each of them.

Subreddit	*Members*	*Description*
r/Coronavirus	2.6M	Official subreddit regarding COVID-19; quality news.
r/Conspiracy	1.6M	Discussion on conspiracy theories.
r/Covid19	339k	Scientific discussion on COVID-19.
r/Vaxxhappened	326k	Ironic and mocking posts against anti-vaxxers.
r/China_Flu	102k	Report news and leaves room for opinion.
r/Covidiots	101k	Irony about anti-vaxxers and virus deniers.
r/NoNewNormal	97.3k	Unscientific discussions on restrictions.
r/CovidVaccinated	33.8k	Personal experiences about COVID-19 vaccines.
r/Antivax	30.5k	Irony about anti-vaxxers.
r/AntiVaxxers	28.6k	Discussions and experiences with anti-vaxxers.
r/CoronavirusCircleJerk	22.6k	Irony and memes about the virus.
r/DebateVaccines	6.9k	Discussions on vaccine issues.
r/CoronavirusFOS	6.4k	Discussions about the virus, little moderated.
r/Vaccines	5.2k	Scientific news and Q&A on vaccines.
r/Vaxxmemes	5.2k	Memes on anti-vaxxers.
r/CovidVaccine	4.7k	Discussions on COVID-19 vaccines.
r/GreatReset	2.3k	Discussions on conspiracy theories.
r/TrueAntiVaccination	2k	Debate between anti-vaxxers.
r/NoLockdownNoMasks	1.6k	Discussion against lockdowns and vaccines.
r/CovidVaccinatedUncut	1.4k	Skeptical debate on the virus and vaccines.
r/Vaccine	1.3k	Q&A on vaccines.

Table 4.1: List of the subreddits on COVID-19 composing our dataset

Exploratory Data Analysis. The number of posts initially present in our dataset is 1,057,273. However, only 271,686 of them represented original content, i.e., neither links to external content nor cross-posts. The number of comments associated with

the selected posts is 25,130,149. Tables 4.2 and 4.3 show the features associated with posts and comments in our dataset.

Field	*Description*
`id`	The unique identifier of the post.
`user`	The user who submitted the post.
`post`	The text of the post.
`subreddit`	The subreddit where the post was submitted.
`datetime`	The submission date and time.

Table 4.2: Features of a post in the dataset

Field	*Description*
`id`	The unique identifier of the comment.
`user`	The user who posted the comment.
`post_id`	The identifier of the post where the comment was written.
`comment`	The text of the comment.
`level`	The comment level: "1" means below the post, "2" means below a comment of level 1, etc.

Table 4.3: Features of a comment in the dataset

Figure 4.1 shows the number of posts and comments published each day in the subreddits of our dataset. It is possible to observe a peak during the first four months of 2020, when the pandemic spread rapidly around the world. Another peak, although much smaller than the previous one, can be observed during the first two months of 2021, when many countries started the vaccination campaign. The trends of posts and comments go in parallel; this was easily predictable since an increase (resp., decrease) in posts usually results in an increase (resp., decrease) in comments.

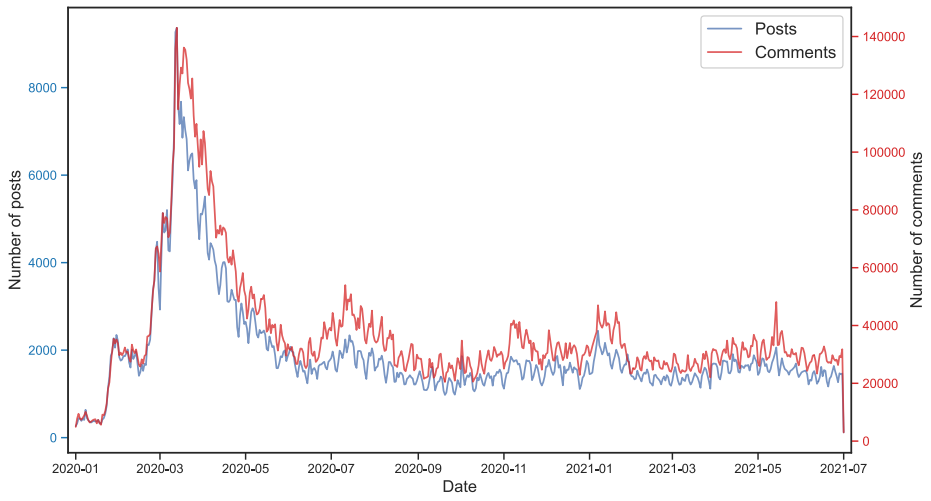


Fig. 4.1: Trends of the number of posts and comments over the time interval of our dataset

Table 4.4 shows the total number of posts and comments and the average number of comments per post for each subreddit. From the analysis of this table we can

observe that the subreddit r/Coronavirus has a much higher number of posts and comments than the other ones. This implies that it has an extremely impactful and dominant position within our dataset. To get a visual feedback of this fact, let us consider Figure 4.2, reporting the trends of the number of posts and comments of r/Coronavirus over the time interval of our dataset, and let us compare it with Figure 4.1. As we can see, the trend in Figure 4.1 is extremely influenced by the one in Figure 4.2.

Subreddit	*Total number of posts*	*Total number of comments*	*Average number of comments*
r/Coronavirus	358,986	13,390,178	37.3
r/conspiracy	236,207	7,561,394	32.01
r/NoNewNormal	69,627	1,567,644	22.51
r/China_Flu	64,421	1,111,147	17.24
r/COVID19	30,455	336,588	11.05
r/Covidiots	17,259	229,305	13.29
r/CoronavirusCirclejerk	16,329	237,712	14.55
r/CovidVaccinated	12,813	121,984	9.52
r/vaxxhappened	11,200	221,273	19.76
r/CoronavirusFOS	7,334	50,652	6.90
r/DebateVaccine	4,925	87,465	17.76
r/AntiVaxxers	3,344	43,876	13.12
r/TrueAntiVaccination	3,046	21,827	7.16
r/antivax	2,836	32,846	11.58
r/CovidVaccine	2,386	20,563	8.62
r/VACCINES	1,975	6,914	3.50
r/Vaccine	865	5,988	6.92
r/CovidVaccinatedUncut	546	1,851	3.39
r/GreatReset	252	1,362	12.31
r/Vaxxmemes	251	1,245	4.96

Table 4.4: Total number of posts and comments and average number of comments per post for each subreddit

Furthermore, let us consider the subreddit r/Conspiracy, which has the most posts and comments after r/Coronavirus in the dataset. Let us consider its trend over time shown in Figure 4.3 and let us compare it to the overall trend in Figure 4.1. We can see that it is completely different; this is an indicator that r/Conspiracy does not have a dominant and impactful position within the dataset.

The extremely dominant position of r/Coronavirus risks to represent a bias for our analysis because a user being an information diffuser only in this subreddit could appear as such in the whole network. By contrast, a user not present on r/Coronavirus, who is an information diffuser for all subreddits, could not be recognized as

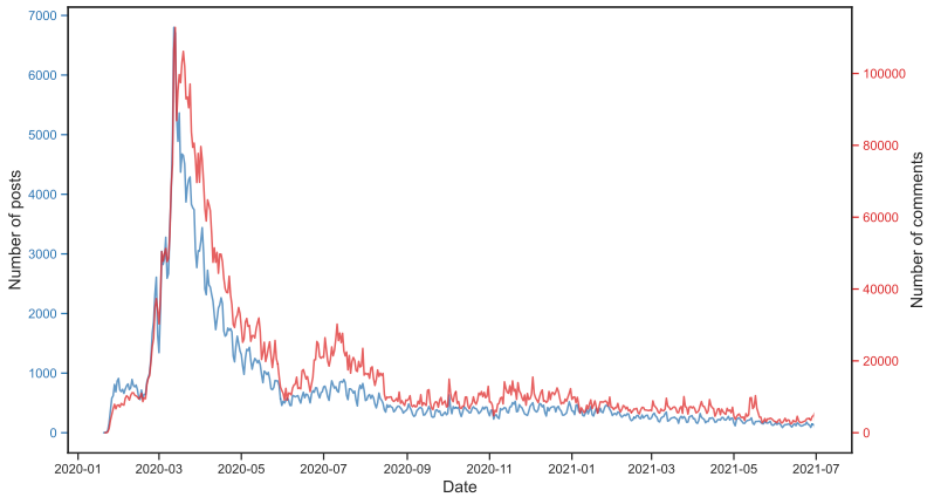


Fig. 4.2: Trends of the number of posts and comments of `r/Coronavirus` over the time interval of our dataset

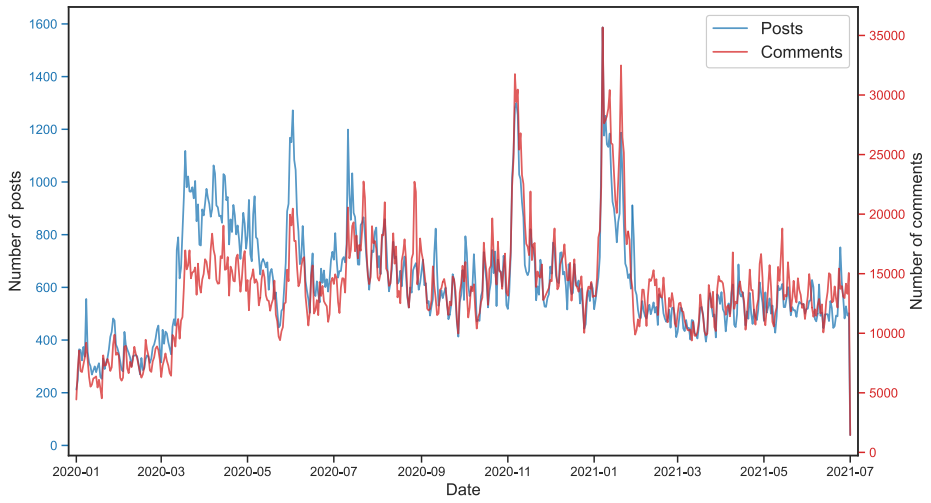


Fig. 4.3: Trends of the number of posts and comments of `r/Conspiracy` over the time interval of our dataset

such in the whole network. For this reason, we decided not to consider this subreddit in the next steps of our experimental campaign.

4.2.2 Construction of the network $\mathcal{N}$

In this section, we illustrate the construction of the network $\mathcal{N}$ from our dataset. It represents the basis for extracting the sets $\hat{U}_Y$ and $\tilde{U}_Y$ of the disseminator bridges and for building the backbones of disseminator bridges.

We recall that $\mathcal{G}_p = \langle U_p, E_p \rangle$, $\mathcal{G}_c = \langle U_c, E_c \rangle$ and $\mathcal{N} = \langle U, E \rangle$, where $E = E_p \cup E_c$. The main characteristics of the three networks are shown in Table 4.5.

Feature	$\mathcal{G}_p$	$\mathcal{G}_c$	$\mathcal{N}$
Number of nodes	76,205	514,473	545,482
Number of arcs	92,204	585,436	590,676
Density	$2.62 \cdot 10^{-5}$	$3.83 \cdot 10^{-6}$	$3.97 \cdot 10^{-6}$
Average clustering coefficient	0	0	0

Table 4.5: Main characteristics of $\mathcal{G}_p$, $\mathcal{G}_c$ and $\mathcal{N}$

From the analysis of this table we can observe that $\mathcal{G}_c$ has a much higher number of nodes and arcs than $\mathcal{G}_p$. As a consequence, the nodes and arcs of $\mathcal{N}$ are strongly influenced by those of $\mathcal{G}_c$. We can observe that all the three networks are loosely connected. In fact, in all cases, the number of arcs is only slightly higher than the number of nodes, the density is very low and the average clustering coefficient is zero. This result can be explained by considering the number of users, posts, comments and subreddits involved. Having removed the subreddit `r/Coronavirus` from the dataset for the reasons explained above, we have 76,205 users in $\mathcal{G}_p$, 514,473 users in $\mathcal{G}_c$, 545,482 users in $\mathcal{N}$, 486,071 posts, 11,661,636 comments and 19 communities, i.e., subreddits. This implies that a user of $\mathcal{G}_p$ makes an average of 6.38 posts, each of which can be published in one of the 19 available communities. Similarly, a user of $\mathcal{G}_c$ makes an average of 22.67 comments each of which can be published in one of the 19 communities. With such low average numbers of posts and comments published by users, the probability of two users publishing at least one post (resp., comment) in the same community, which is a necessary condition for an arc to exist in $\mathcal{G}_p$ (resp., $\mathcal{G}_c$), is very low. This explains why $\mathcal{G}_p$ and $\mathcal{G}_c$ have very few arcs and thus are loosely coupled. As for $\mathcal{N}$, it is obtained by a "merge" of $\mathcal{G}_p$ and $\mathcal{G}_c$; therefore, it is natural that if the latter have a very low density, $\mathcal{N}$ has a low density too, and is loosely coupled.

Observe that $|U_{pc}| = |U_p \cap U_c| = 45,196$, $\frac{|U_{pc}|}{|U_p|} = 0.5931$, $\frac{|U_{pc}|}{|U_c|} = 0.088$, $\frac{|U_{pc}|}{|U|} = 0.083$. This implies that the users publishing both posts and comments are a very small fraction of the overall users and of the ones publishing comments. This result was actually expected given the number of nodes in $\mathcal{G}_p$, $\mathcal{G}_c$ and $\mathcal{N}$. Instead, we were surprised by the existence of a very high fraction (equal to 41.69%) of users, who published posts and never published a comment.

Finally, $|E_{pc}| = |E_p \cap E_c| = 47,438$, $\frac{|E_{pc}|}{|E_p|} = 0.5145$, $\frac{|E_{pc}|}{|E_c|} = 0.0810$, $\frac{|E_{pc}|}{|E|} = 0.0803$. These results are in line with those concerning nodes. In fact, again, the arcs belonging to E_{pc} are only a very small fraction of the ones belonging to E_c and E. This was expected given the different order of magnitude of $\mathcal{G}_p$ compared to $\mathcal{G}_c$ and $\mathcal{N}$. What is surprising, instead, is the presence of a large fraction (i.e., 37.75%) of arcs belonging to E_{pc} that do not belong to E_p. This means that there are many pairs of users

such that one of them publishes a post and the other publishes a comment. The next experiments are devoted to understand if this behavior is random or if there is a backbone of users who adopt this strategy to spread as much as possible the information of their interest.

4.2.3 Construction of $\hat{U}_Y$

In this section, we illustrate the construction of the set $\hat{U}_Y$ of the disseminator bridges obtained by using the traditional centrality measures derived from Social Network Analysis. In Figure 4.4 (resp., 4.5, 4.6), we show the distribution of the users of U against their degree (resp., closeness, betweenness) centrality. These three figures confirm what is expected from Social Network Analysis for the various types of centralities under consideration. In fact, the distributions of users against degree centrality and betweenness centrality follow a power law, while the distribution of users against closeness centrality follows a Gaussian. In our case, both the power law and the Gaussian distributions are very "extreme". In other words, the two power law distributions are very steep while the Gaussian distribution is very narrow, with a very low standard deviation from the mean. In particular, the values of α and δ are equal to 1.32 and 0.16 for degree centrality, and to 1.43 and 0.11 for betweenness centrality. The mean and the standard deviation of the Gaussian distribution are 0.44 and 0.14.

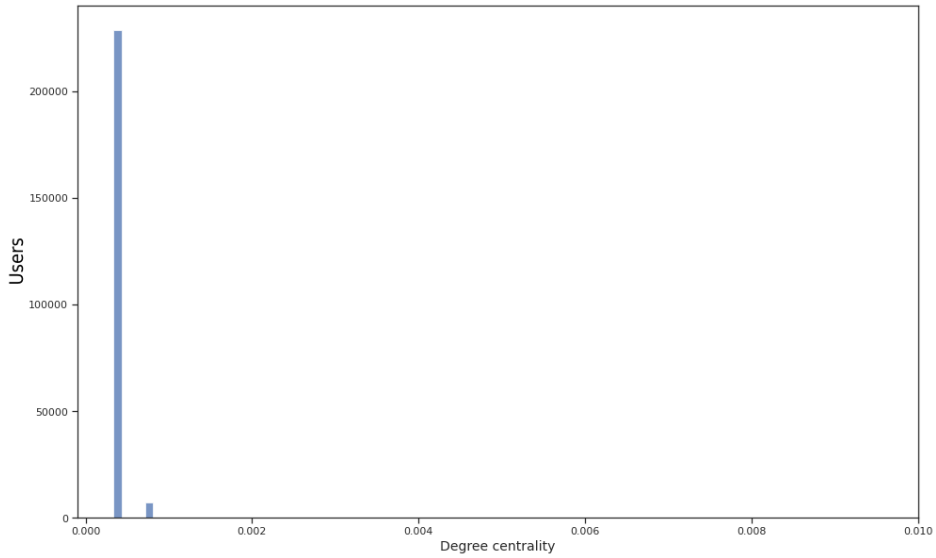


Fig. 4.4: Distribution of the users of U against their degree centrality

Given the steepness of the two power law distributions and the narrowness of the Gaussian one, it could be expected that most of the elements with high values

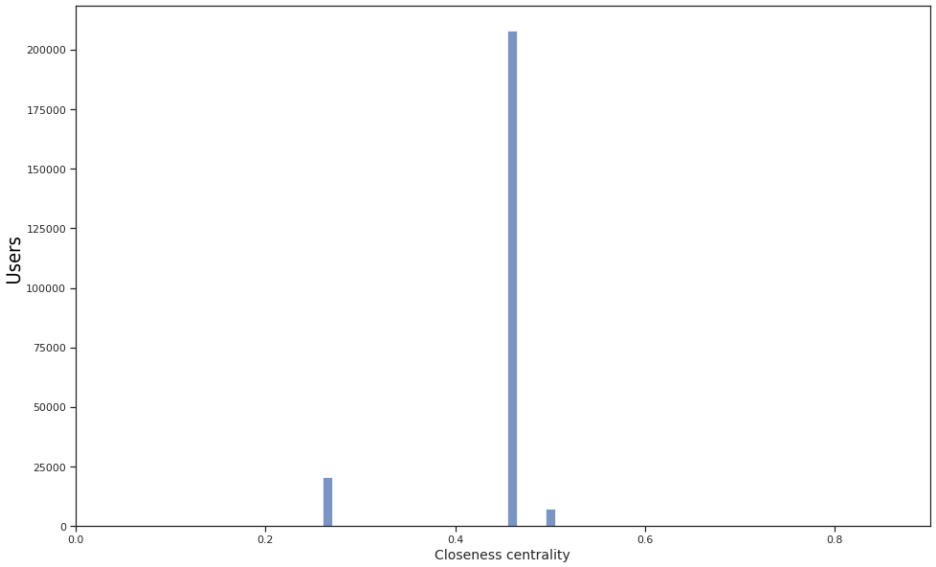


Fig. 4.5: Distribution of the users of U against their closeness centrality

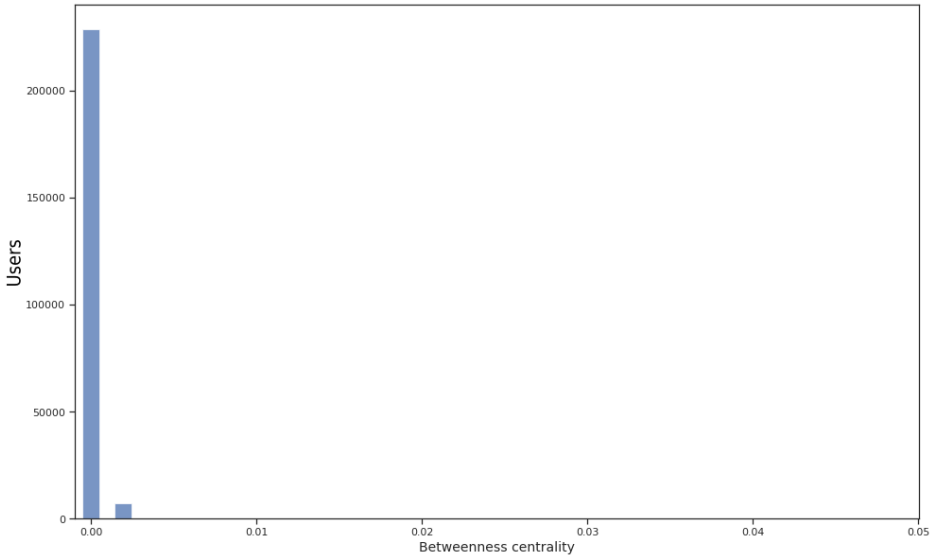


Fig. 4.6: Distribution of the users of U against their betweenness centrality

of a type of centrality have high values for the other two types. To test the latter hypothesis, we considered the top 250, 500, 1,000 and 10,000 users with the highest values of degree, closeness, betweenness and combined centrality. This is equivalent to computing the sets $Top(L^{dc}, Y)$, $Top(L^{cc}, Y)$, $Top(L^{bc}, Y)$ and $Top(L^{CombC}, Y)$, with Y equal to 0.0458%, 0.0917%, 0.183% and 1.8332%. As can be seen, these are very low values of Y, which are justified by the characteristics of the three distributions. Given a value of Y, we considered how many users belong to the intersection of two or more of the sets above. The results obtained are reported in Table 4.6.

Top users	*Sets*	*Percentage of top users belonging to it*
250	$Top(L^{dc}, Y) \cap Top(L^{cc}, Y)$	0.9925
	$Top(L^{dc}, Y) \cap Top(L^{bc}, Y)$	0.9923
	$Top(L^{cc}, Y) \cap Top(L^{bc}, Y)$	0.9991
	$Top(L^{dc}, Y) \cap Top(L^{CombC}, Y)$	0.9896
	$Top(L^{cc}, Y) \cap Top(L^{CombC}, Y)$	0.9854
	$Top(L^{bc}, Y) \cap Top(L^{CombC}, Y)$	0.9889
500	$Top(L^{dc}, Y) \cap Top(L^{cc}, Y)$	0.9895
	$Top(L^{dc}, Y) \cap Top(L^{bc}, Y)$	0.9867
	$Top(L^{cc}, Y) \cap Top(L^{bc}, Y)$	0.9943
	$Top(L^{dc}, Y) \cap Top(L^{CombC}, Y)$	0.9776
	$Top(L^{cc}, Y) \cap Top(L^{CombC}, Y)$	0.9621
	$Top(L^{bc}, Y) \cap Top(L^{CombC}, Y)$	0.9734
1,000	$Top(L^{dc}, Y) \cap Top(L^{cc}, Y)$	0.9887
	$Top(L^{dc}, Y) \cap Top(L^{bc}, Y)$	0.9745
	$Top(L^{cc}, Y) \cap Top(L^{bc}, Y)$	0.9901
	$Top(L^{dc}, Y) \cap Top(L^{CombC}, Y)$	0.9654
	$Top(L^{cc}, Y) \cap Top(L^{CombC}, Y)$	0.9343
	$Top(L^{bc}, Y) \cap Top(L^{CombC}, Y)$	0.9452
10,000	$Top(L^{dc}, Y) \cap Top(L^{cc}, Y)$	0.9421
	$Top(L^{dc}, Y) \cap Top(L^{bc}, Y)$	0.9608
	$Top(L^{cc}, Y) \cap Top(L^{bc}, Y)$	0.9496
	$Top(L^{dc}, Y) \cap Top(L^{CombC}, Y)$	0.9114
	$Top(L^{cc}, Y) \cap Top(L^{CombC}, Y)$	0.8945
	$Top(L^{bc}, Y) \cap Top(L^{CombC}, Y)$	0.9001

Table 4.6: Percentage of top users belonging to the intersection of some sets of interest

From the analysis of this table we can see that our hypothesis is fully confirmed. At least 94% of the top users with the highest values of a basic centrality (degree, closeness and betweenness) are present in the top users of another centrality. Furthermore, at least 89% of the top users with the highest value of combined centrality belong to the top users of a basic centrality. All these percentages increase significantly as the number of top users considered decreases from 10,000 to 1,000, 500 and 250.

4.2.4 Construction of $\tilde{U}_Y$ and comparison with $\hat{U}_Y$

As a first step of this analysis, we determined the set $\tilde{U}_Y$ for the same values of Y that we used to build $\hat{U}_Y$ in the previous section. Specifically, we set Y equal to 0.0458%, 0.0917%, 0.183% and 1.8332% so as to select the first 250, 500, 1,000 and 10,000 users with the highest values of disseminator centrality. Then, for each of these sets $Top(L^{DC}, Y)$, we considered the intersection with the corresponding sets $Top(L^{dc}, Y)$, $Top(L^{cc}, Y)$ and $Top(L^{bc}, Y)$. The obtained results are shown in Table 4.7.

From the analysis of this table we can see that the percentage of users belonging to the various intersections is very high but significantly lower than those seen in

Top users	*Sets*	*Percentage of top users belonging to it*
250	$Top(L^{DC}, Y) \cap Top(L^{dc}, Y)$	0.9701
	$Top(L^{DC}, Y) \cap Top(L^{cc}, Y)$	0.9786
	$Top(L^{DC}, Y) \cap Top(L^{bc}, Y)$	0.9754
	$Top(L^{DC}, Y) \cap Top(L^{CombC}, Y)$	0.9632
500	$Top(L^{DC}, Y) \cap Top(L^{dc}, Y)$	0.9332
	$Top(L^{DC}, Y) \cap Top(L^{cc}, Y)$	0.9426
	$Top(L^{DC}, Y) \cap Top(L^{bc}, Y)$	0.9503
	$Top(L^{DC}, Y) \cap Top(L^{CombC}, Y)$	0.9032
1000	$Top(L^{DC}, Y) \cap Top(L^{dc}, Y)$	0.8997
	$Top(L^{DC}, Y) \cap Top(L^{cc}, Y)$	0.9012
	$Top(L^{DC}, Y) \cap Top(L^{bc}, Y)$	0.8876
	$Top(L^{DC}, Y) \cap Top(L^{CombC}, Y)$	0.8407
1000	$Top(L^{DC}, Y) \cap Top(L^{dc}, Y)$	0.8431
	$Top(L^{DC}, Y) \cap Top(L^{cc}, Y)$	0.8398
	$Top(L^{DC}, Y) \cap Top(L^{bc}, Y)$	0.8121
	$Top(L^{DC}, Y) \cap Top(L^{CombC}, Y)$	0.7985

Table 4.7: Percentage of top users belonging to the intersection of $Top(L^{DC}, Y)$ with the other sets of interest

Table 4.6. This leads us to conclude that there is a group of users that fall among the disseminator bridges whatever the metric adopted to identify this type of users (i.e., one of the basic centralities, the combined or the disseminator ones). However, there are other users who are identified as disseminator bridges only if the disseminator centrality is adopted. The next step of the experiments aims to understand what are the characteristics of these users and what distinguishes them from those users who we would have been able to find by means of combined centrality alone.

For this purpose, we considered the set $Top(L^{DC}, Y)$ (resp., $Top(L^{CombC}, Y)$), with $Y = 0.183\%$ in order to select the top 1,000 users with the highest disseminator (resp., combined) centrality. We could have done this analysis by selecting the top 250, 500 or 10,000 disseminator bridges. However, the first two sets were too limited while the last one was too large and involved the risk of selecting as disseminator bridges some users who were not in reality.

After that, we considered the users belonging to $Top(L^{DC}, Y) - Top(L^{CombC}, Y)$ and those belonging to $Top(L^{CombC}, Y) - Top(L^{DC}, Y)$. In other words, we considered the sets of the users that resulted as disseminator bridges for the disseminator centrality but not for the combined centrality, and vice versa. In the following, for simplicity, we call the former set $D - C$ and the latter one $C - D$. Each set consists of 160 users. Comparing the characteristics of the two sets we can understand the aspects considered by disseminator centrality that allowed the identification of disseminator bridges not recognized by combined centrality. Recall that the aspects considered by disseminator centrality (Section 4.1.2) are: *(i)* the number of published posts (resp.,

comments); *(ii)* the number of communities in which at least one post (resp., comment) was published; *(iii)* the equidistribution of published posts (resp., comments) across communities. For each of these aspects, we considered the distribution of the users of $D - C$ and $C - D$.

Figure 4.7 (resp., 4.8) shows the distribution of the users of $D - C$ (on the left) and $C - D$ (on the right) against the posts (resp., comments) published by them. As for posts, there is a substantial difference between the two sets. In fact, the users of $D - C$ are uniformly distributed with a slight prevalence of the second quartile. By contrast, the users of $C - D$ are found almost exclusively in the third and fourth quartiles. As for comments, most of the users of $D - C$ are in the third and fourth quartiles while the users of $C - D$ predominantly occupy the first two quartiles.

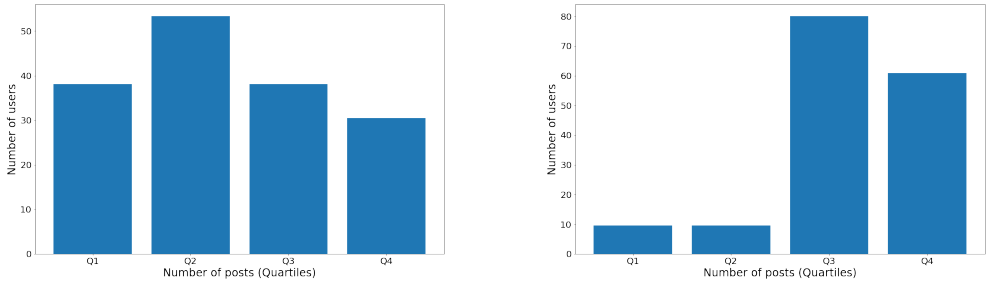


Fig. 4.7: Distribution of the users of $D-C$ (on the left) and $C-D$ (on the right) against the posts published by them

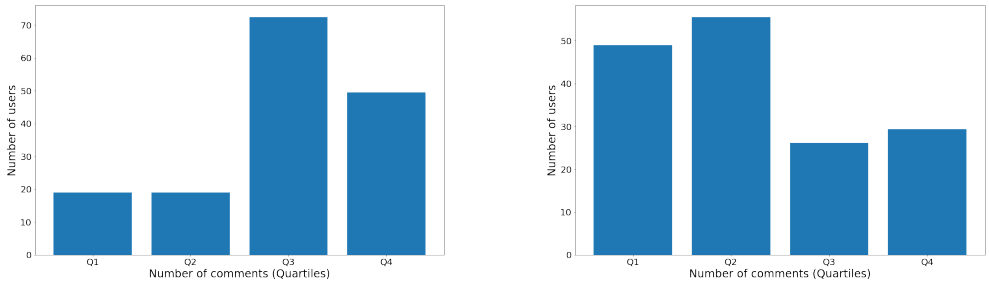


Fig. 4.8: Distribution of the users of $D-C$ (on the left) and $C-D$ (on the right) against the comments published by them

Figure 4.9 (resp., 4.10) illustrates the distribution of the users of $D-C$ (on the left) and $C-D$ (on the right) against the number of communities in which they published posts (resp., comments). Figure 4.9 shows a great difference between the two sets. In fact, the users of $D - C$ occupy only the first and third quartiles, with a dominance of the first one. Instead, the users of $C - D$ occupy the first, the third and the fourth quartiles, with a dominance of the third one. A great difference between $D - C$ and

$C-D$ can be also seen in Figure 4.10. In fact, the users of $D-C$ can be found mostly in the third and fourth quartiles, whereas the users of $C-D$ are more uniformly distributed, with a prevalence of the first two quartiles.

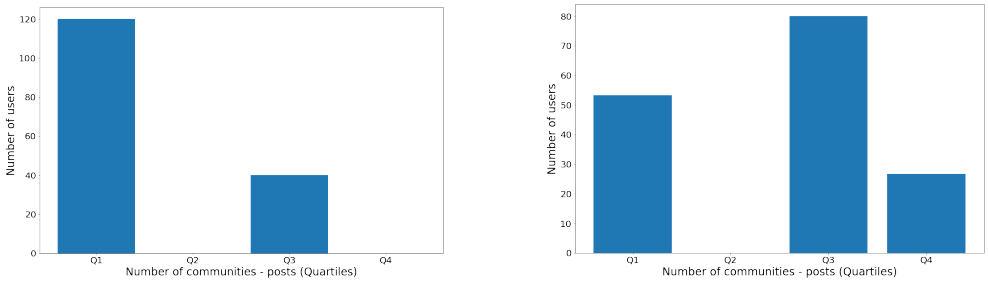


Fig. 4.9: Distribution of the users of $D-C$ (on the left) and $C-D$ (on the right) against the number of communities in which they published posts

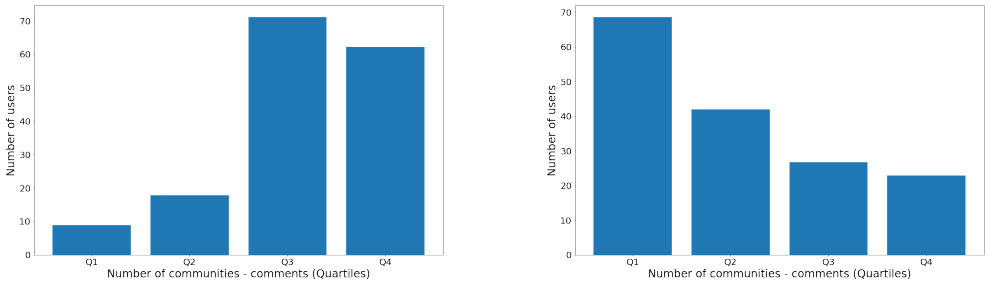


Fig. 4.10: Distribution of the users of $D-C$ (on the left) and $C-D$ (on the right) against the number of communities in which they published comments

In Figure 4.11 (resp., 4.12), we illustrate the distribution of the users of $D-C$ (on the left) and $C-D$ (on the right) with regard to the equidistribution of the posts (resp., comments) published by them across communities calculated by customizing the HHI to this context (see Section 4.1.2). First of all, we point out that, in this case, the best values are the smallest ones and not the highest ones, so the quartiles are constructed by sorting the values in ascending order. In Figure 4.11, we can observe that for both $D-C$ and $C-D$ there is a dominance of the second quartile. However, in $D-C$ this dominance is strong, whereas in $C-D$ it is slight because the first and the third quartiles comprise a high number of users. As for comments (Figure 4.12), we can observe a predominance of the first two quartiles for both $D-C$ and $C-D$. However, as for $D-C$, there is a slight dominance of the first quartile on the second one, whereas in $C-D$ there is a more marked dominance of the second quartile on the first one.

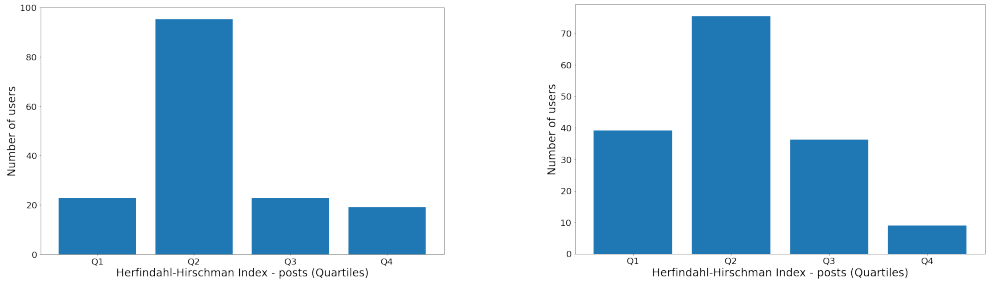


Fig. 4.11: Distribution of the users of $D-C$ (on the left) and $C-D$ (on the right) with regard the equidistribution of the posts published by them across communities

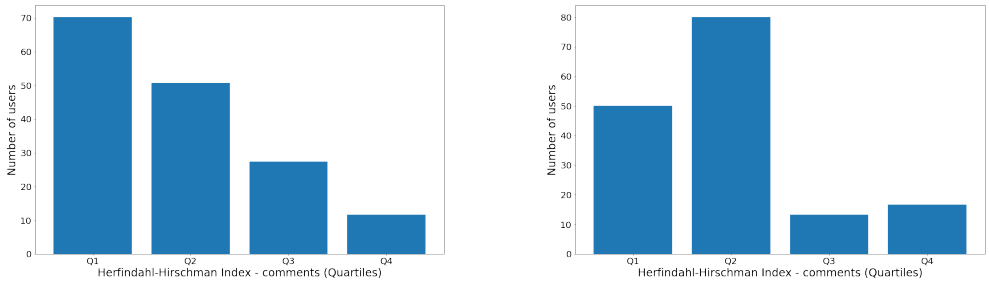


Fig. 4.12: Distribution of the users of $D-C$ (on the left) and $C-D$ (on the right) with regard the equidistribution of the comments published by them across communities

In conclusion, we can say that the six aspects that characterize disseminator centrality really contribute to identify new disseminator bridges not detectable through classical centralities. This result is important for both detecting current information disseminators and guiding users who aspire to become information disseminators in the future.

It is clear from the previous results that disseminator centrality is very well suited for finding information diffusers across different communities of a social platform. And, indeed, it belongs to the class of centrality measures that have been proposed in the literature to address a specific problem. From this point of view, it is very different from the classical centrality measures (degree, closeness, eigenvector and betweenness centralities, or various combinations of them obtained through the application of aggregate operators). These, being general, can be applied in various contexts, often providing very good results, even if lower than those obtained using ad hoc centrality measures.

However, there is one last issue that we must consider before we can say that disseminator centrality is really effective and efficient in identifying disseminator bridges. It concerns the cost necessary to compute such a centrality measure. To verify whether this cost is acceptable, we computed the time required to calculate the value of degree, closeness, betweenness, combined and disseminator centralities

for all the nodes of the network $\mathcal{N}$ associated with our dataset. This time is reported in Table 4.8.

Centrality	*Computation time*
Average Degree Centrality	3.16s
Average Closeness Centrality	37,825s
Average Betwenness Centrality	28,443s
Average Combined Centrality	66,272s
Average Disseminator Centrality	5.12s

Table 4.8: Computation time of the average degree, closeness, betweenness, combined and disseminator centralities for the nodes of the network $\mathcal{N}$ associated with our dataset

From the analysis of this table we can observe that disseminator centrality has an excellent computation time. In fact, only degree centrality has a slightly lower computation time. Instead, the other three forms of centrality have computation times that are orders of magnitude greater than that of disseminator centrality. This allows us to say that this last form of centrality is not only very effective but also very efficient in achieving the goals for which it was designed.

4.2.5 Construction of the backbone of disseminator bridges

The first activity we performed during this experiment involved building the Interaction Network IN. We recall that IN has a node for each user of U while there is an arc between two nodes v_i and v_j only if u_j published at least one top-level positive comment for a post of u_i, and vice versa. In Table 4.9, we report some information on IN. It is clear that the condition for the existence of an arc in IN is very stringent. And, in fact, Table 4.9 shows that the density of IN is very low. Furthermore, it has a connected component with a much larger size than the others; the density of this component is more than twice the one of IN. This observation leads us to hypothesize that the disseminator bridges are to be found in this connected component, which is the assumption we made in Section 4.2.5. Clearly, in the following, we must verify the correctness of this assumption.

At this point, we considered interesting to compute $\mathcal{DBB}_Y$, for the values of Y seen in the previous section, i.e., Y equal to 0.0458%, 0.0917% and 0.183%, which allow the selection of the top 250, 500 and 1,000 disseminator bridges. We did not consider the top 10,000 disseminator bridges because the number of nodes of the maximum connected component of IN is much smaller than this number and, therefore, this computation would make no sense. The results obtained are as follows:

Characteristic	*Value*
Number of nodes	545,482
Number of nodes with degree > 0	7,423
Number of arcs	24,378
Density	$1.2 \cdot 10^{-5}$
Number of connected components (excluding single nodes)	3,283
Size of the maximum connected component	1084 nodes
Density of the maximum connected component	$2.7 \cdot 10^{-5}$
Size of the second connected component	88 nodes
Size of the third connected component	12 nodes
Size of the fourth connected component	7 nodes
Size of the fifth connected component	5 nodes

Table 4.9: Some characteristics of IN and its main connected components

- $|\mathcal{DBB}_{0.0458\%}| = 250$, i.e., all 250 disseminator bridges of $\tilde{U}_Y$ are actually in the maximum connected component of IN.
- $|\mathcal{DBB}_{0.0917\%}| = 483$, i.e., given the 500 disseminator bridges of $\tilde{U}_Y$, 483 (equal to 96.6%) are actually in the maximum connected component of IN.
- $|\mathcal{DBB}_{0.183\%}| = 934$, i.e., given the 1000 disseminator bridges of $\tilde{U}_Y$, 934 (equal to 93.4%) are actually in the maximum connected component of IN.

As a last task, we considered interesting to repeat the previous computation replacing $\hat{U}_Y$ to $\tilde{U}_Y$, that is considering the disseminator bridges returned by combined centrality, instead of those returned by disseminator centrality. In this case, we obtained $|\mathcal{DBB}_{0.0458\%}| = 241$, $|\mathcal{DBB}_{0.0917\%}| = 455$ and $|\mathcal{DBB}_{0.183\%}| = 860$.

This last result is extremely interesting because it represents a further confirmation of the fact that disseminator centrality, besides being much less expensive than combined centrality, is more effective than the latter in identifying information diffusers among different communities of a social platform.

5

Investigating the NSFW phenomenon

In this chapter, we analyze the phenomenon of NSFW contents. The first analysis is structural; in it we study the characteristics of NSFW (Not Safe For Work) posts in Reddit, highlighting their differences from SFW posts, which have been much more studied in the past literature. In our investigation, we consider Reddit posts from 2019. Through both descriptive analytics and social network analysis techniques, we detect three insights on the main differences between NSFW and SFW posts in Reddit. Thanks to these insights, we are able to better understand the dynamics (authors, subreddits, readers) behind NSFW posts. In particular, it becomes clear that this is a niche world where authors are strongly cohesive. However, at the same time, the most popular ones show a clear opening to new authors, with whom they are willing to collaborate from the beginning. The material presented was derived from [223].

The second investigation is based on content analysis and proposes an approach for extracting and analyzing text patterns from NSFW adult content in Reddit. Some peculiarities of this approach are the following: (i) text patterns are extracted based not only on frequency but also, and mostly, on several utility measures; (ii) extracted patterns contribute to the definition of social networks whose analysis allows us to extract several useful information about the users publishing and/or accessing NSFW content and the language adopted by them; (iii) our approach is not only descriptive but also predictive, because, in addition to identifying already existing user communities, it is able to propose new ones; these are made up of users who do not yet know each other but share the same interests and the same language. The material presented was derived from [234].

5.1 Structural investigation

5.1.1 Methods

Dataset description. The dataset used for our analysis has been downloaded from the website `pushshift.io` [65], one of the main Reddit data sources. In particular,

we extracted all the posts published on Reddit from January 1^{st}, 2019 to September 1^{st}, 2019[1]. The number of posts available for our analysis was 150,795,895. In Reddit, an NSFW post must be marked as such by its author. Therefore, there is no need for automatic labeling by Reddit or manual labeling by third-parties. If the user specifies that a post she/he is publishing is NSFW, Reddit puts a red label when displaying it and sets the value of the `over_18` field in its database to `true`. We used the value of this field to separate NSFW posts from SFW ones in our analyses.

We performed a preliminary ETL (Extraction, Transformation and Loading) activity on our dataset. In Data Analytics, this activity is typically carried out prior to any data analysis campaign. It aims at cleaning the data in the dataset, removing any errors and inconsistencies, integrating any data from different sources, and transforming the cleaned and integrated data into a single format chosen for the next data analysis tasks [473].

During the ETL phase, we observed that some of the available posts were made by authors who had left Reddit. We decided to remove these posts from our dataset. At the end of this activity, the number of available posts was 122,568,630. NSFW posts were 11,908,377, equivalent to 9.72% of them.

As pointed out in the Introduction, the goal of our study is to understand the characteristics of NSFW posts and their authors, comparing them with the SFW posts and their authors. For this reason, we decided to extract from the dataset described above two sub-datasets, with the same number of posts each. Both of them are limited to January and February 2019. The first dataset $\mathcal{D}$ contains only SFW posts, while the second, called $\overline{\mathcal{D}}$, stores only NSFW posts. We randomly selected 1,250,000 posts for each of them to reduce the datasets' size and the computation time. It should be noted that this number is absolutely in line with the number of posts generally used in the analyses of Reddit [638, 461, 583, 286]. However, we repeated all the analyses on two other datasets $\mathcal{D}'$ and $\overline{\mathcal{D}}'$ to verify the stability of our results. The set $\mathcal{D}'$ (resp., $\overline{\mathcal{D}}'$) consists of 1,250,000 SFW (resp., NSFW) posts published in March and April 2019, randomly selected from the original dataset. In addition, we carried out a deeper stability check evaluating all posts of 2019 month by month.

As a preliminary analysis, we focused on the "context" of SFW and NSFW posts. Here, we use the term "context" of a post to denote its author, its comments and the subreddits in which it was published. In this analysis, we wanted to verify if the context of SFW posts and the one of NSFW posts are the same or not. To answer this

[1] Actually, only for stability analysis, we considered all the posts from January 1^{st}, 2019 to December 31^{st}, 2019 (see Section 5.1.1).

question, we calculated the values of some parameters on $\mathcal{D}$ and $\overline{\mathcal{D}}$ and, then, on $\mathcal{D}'$ and $\overline{\mathcal{D}}'$. The results obtained are shown in Table 5.1.

Parameter	$\mathcal{D}$ *and* $\overline{\mathcal{D}}$	$\mathcal{D}'$ *and* $\overline{\mathcal{D}}'$
Number of authors who published at least one SFW post	59,465	58,561
Number of authors who published only SFW posts	58,801	57,891
Percentage of authors publishing SFW posts who published only posts of this type	98.88%	98.52%
Number of authors who published at least one NSFW post	36,758	36,461
Number of authors who published only NSFW posts	36,094	36,131
Percentage of authors publishing NSFW posts who published only posts of this type	98,19%	99.09%
Number of subreddits containing at least one SFW post	89,360	92,445
Number of subreddits containing only SFW posts	82,050	85,157
Percentage of subreddits containing SFW posts that contain only posts of this type	91.82%	92.12%
Number of subreddits containing at least one NSFW post	41,365	45,910
Number of subreddits containing only NSFW posts	34,055	38,622
Percentage of subreddits containing NSFW posts that contain only posts of this type	82.33%	84.13%

Table 5.1: Parameters about the authors and the subreddits of SFW and NSFW posts - $\mathcal{D}$ (resp., $\overline{\mathcal{D}}$) stores SFW (resp., NSFW) posts of January and February 2019, while $\mathcal{D}'$ (resp., $\overline{\mathcal{D}}'$) stores the same kind of post but for March and April 2019

This table shows that the reference contexts for SFW and NSFW posts are basically independent. In fact, more than 98% of authors writing SFW posts do not write NSFW posts, and vice versa. In addition, more than 91% of subreddits containing SFW posts do not contain NSFW posts, and more than 82% of subreddits containing NSFW posts do not contain SFW posts. Another important result is that all the computations are stable over time because the values obtained for January and February 2019 (Jan-Feb, for short) are very similar to the ones returned for March and April 2019 (Mar-Apr, for short).

Investigating the NSFW posts. In this section, we present some analyses directly involving NSFW and SFW posts. In particular, we study the distribution of subreddits and authors against posts and the distribution of posts against the scores assigned to them by Reddit users.

Firstly, we computed the distributions of the subreddits against NSFW and SFW posts for the datasets $\mathcal{D}$ and $\overline{\mathcal{D}}$. The results obtained are reported in Figure 5.1.

This figure shows that the two distributions follow a power law. We also computed some parameters for the two power law distributions; they are shown in the second and third columns of Table 5.2. To verify the stability of results found, we made the same computations on $\mathcal{D}'$ and $\overline{\mathcal{D}}'$ datasets. They are shown in the fourth and fifth columns of Table 5.2.

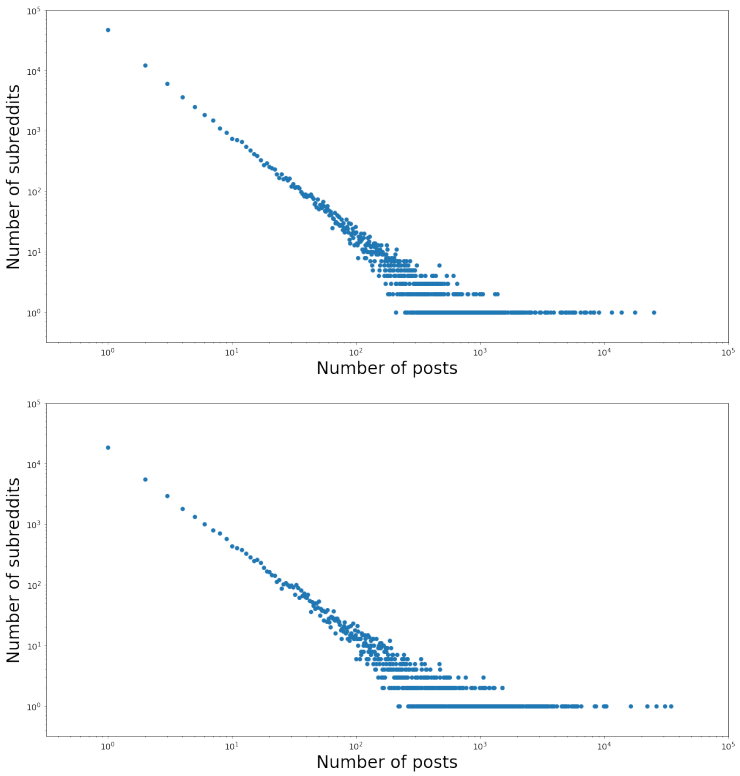


Fig. 5.1: Log-log plots of the distributions of subreddits against SFW posts (on top) and NSFW posts (on bottom) - Datasets regarding January and February 2019

Parameter	*SFW posts Jan-Feb*	*NSFW posts Jan-Feb*	*SFW posts Mar-Apr*	*NSFW posts Mar-Apr*
Maximum number of subreddits	47,480 (53.13%)	18,332 (44.31%)	49,502 (53.24%)	21,034 (45.02%)
Number of subreddits of the 99 percentile	1,095	571	1,101	569
Maximum number of posts	25,006 (4.62%)	34,424 (4.57%)	26,650 (4.98%)	31,329 (4.76%)
Number of posts of the 99 percentile	7,719	9,862	7,721	9,859
Average number of subreddits	126	54	137	57
Average number of posts	767	981	768	905
α (power law parameter)	1.6539	1.6974	1.6767	1.6859
δ (power law parameter)	0.0266	0.0364	0.0306	0.0432

Table 5.2: Parameters of the distributions of subreddits against posts

From this table, we can observe that the maximum and the average numbers of subreddits for SFW posts is more than twice the value obtained for NSFW posts. The maximum and the average numbers of NSFW posts in a subreddit are slightly higher than SFW posts. There are no significant differences in the α and δ parameters of the two power law distributions. Indeed, both of them are very steep. The comparison of the second and the third columns of Tables 5.2, on the one hand, and the fourth and fifth columns of the same table, on the other hand, also tells us that the trends

obtained are stable over time, because their variations between Jan-Feb and Mar-Apr are not significant.

Although the two curves show almost identical trends, as confirmed by the similar values of α and δ, we found interesting the differences in the maximum and average values. In other words, the curve shapes are similar but the ranges of values are different. To confirm these results we compared the two distributions through the *Wilcoxon rank sum test* [639].

This test indicated that the number of subreddits in which Jan-Feb SFW posts were published was statistically significantly higher than the corresponding one of NSFW posts ($\tau = 2.8 \cdot 10^{-4}, p < 0.01$).

This result can be explained taking into account the intrinsic nature of NSFW posts, whose content is certainly less suitable for the general public than the one of SFW posts.

Then, in Figure 5.2 we show the distributions of authors against SFW and NSFW posts for the datasets $\mathcal{D}$ and $\overline{\mathcal{D}}$. From the analysis of this figure we can see that both distributions follow a power law.

In Table 5.3, we report the main parameters of these two power law distributions for the datasets $\mathcal{D}$ and $\overline{\mathcal{D}}$, on one hand, and $\mathcal{D}'$ and $\overline{\mathcal{D}}'$, on the other hand.

Parameter	*SFW posts Jan-Feb*	*NSFW posts Jan-Feb*	*SFW posts Mar-Apr*	*NSFW posts Mar-Apr*
Maximum number of authors	555,854 (79.06%)	131,070 (56.43%)	551,863 (78.97%)	133,594 (57.01%)
Number of authors of the 99 percentile	11,471	5,055	11,469	5,052
Maximum number of posts	18,724 (11.85%)	16,383 (5.70%)	16,513 (10.98%)	15,674 (5.48%)
Number of posts of the 99 percentile	5,426	5,393	5,424	5,393
Average number of authors	2,190	439	2,083	416
Average number of posts	491	543	491	521
α (power law parameter)	1.4631	1.5566	1.4505	1.5435
δ (power law parameter)	0.0473	0.0353	0.0304	0.0287

Table 5.3: Parameters of the distributions of authors against posts

A Wilcoxon rank sum test showed that the number of authors of Jan-Feb SFW posts was statistically significantly higher than the corresponding one of NSFW posts ($\tau = 1.2 \cdot 10^{-4}, p < 0.01$).

This result can also be explained taking into account the topics of NSFW posts. Indeed, these are more specific than those involving SFW posts. Differently from SFW posts that can be written by anyone, the authors who generally publish NSFW posts are a small circle of people almost exclusively dedicated to this type of post. Consequently, while it is true that NSFW posts are much fewer than SFW posts, it

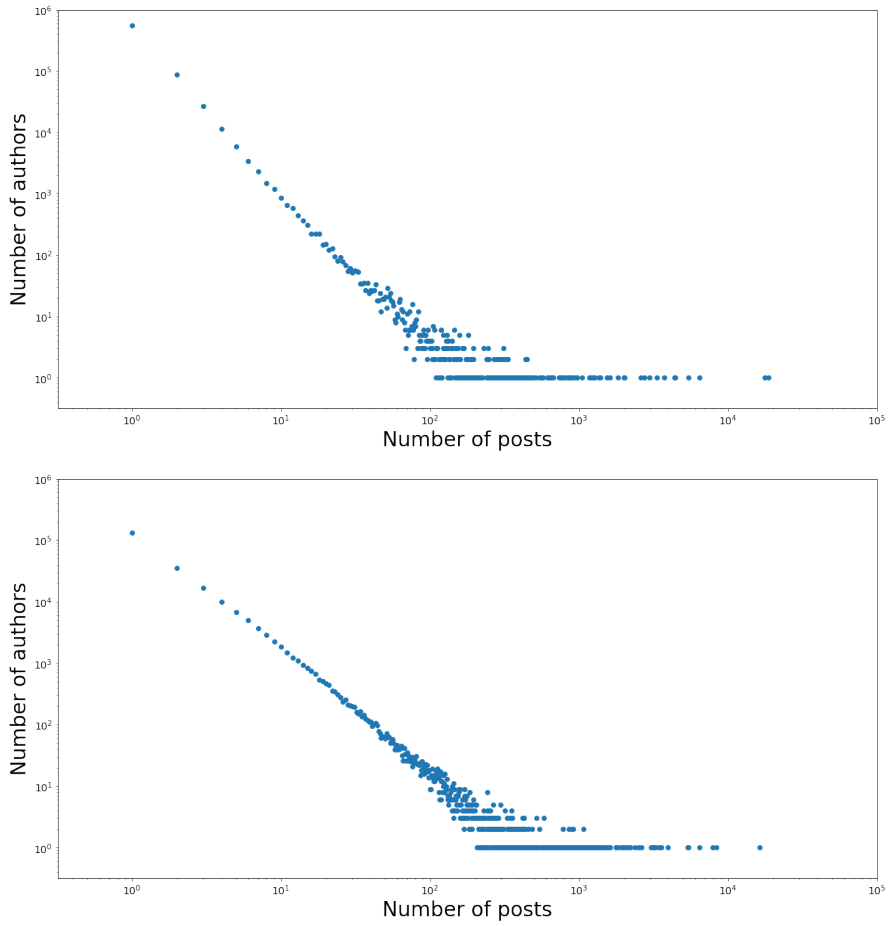


Fig. 5.2: Log-log plots of the distributions of authors against SFW posts (on top) and NSFW posts (on bottom) - Datasets regarding January and February 2019

is also true that they are published by an extremely limited number of authors. This explains the result.

Now, we want to evaluate the distribution of posts and their relative scores. A newly submitted post on Reddit has a score of 1. A user can upvote (resp., downvote) the post, increasing (resp., decreasing) its score by 1. We have computed the distributions of SFW and NSFW posts against scores for the datasets $\mathcal{D}$ and $\overline{\mathcal{D}}$, and, then, for $\mathcal{D}'$ and $\overline{\mathcal{D}}'$, on the other hand. For the sake of simplicity, in Table 5.4, we report the main parameters of these distributions, which again follow a power law.

A Wilcoxon rank sum test showed that the score of Jan-Feb SFW posts was statistically significantly higher than the corresponding one of NSFW posts ($\tau = 0.00109, p < 0.01$).

Once again, this result can be explained by the type of contents that generally characterizes NSFW posts.

Finally, we computed the distributions of subreddits against the authors of SFW and NSFW posts. In both cases, we saw that they follow a power law similar to those

Parameter	*SFW posts Jan-Feb*	*NSFW posts Jan-Feb*	*SFW posts Mar-Apr*	*NSFW posts Mar-Apr*
Maximum score	183,453 (57.98%)	106,947 (47.26%)	191,864 (61.87%)	112,830 (49.62%)
Number of score of the 99 percentile	4,746	3,645	4,825	3,275
Average score	9,881	4,191	8,809	3,819
α (power law parameter)	1.5998	1.5140	1.6061	1.5165
δ (power law parameter)	0.0197	0.0366	0.0154	0.0355

Table 5.4: Parameters of the distributions of posts against scores

shown in the previous figures. We report the values of the most important parameters in Table 5.5.

Parameter	*SFW posts Jan-Feb*	*NSFW posts Jan-Feb*	*SFW posts Mar-Apr*	*NSFW posts Mar-Apr*
Maximum number of subreddits	62,839 (70.32%)	29,798 (72.03%)	65,861 (71.12%)	33,963 (72.01%)
Number of subreddits of the 99 percentile	932	538	930	533
Average number of subreddits	151	87	161	101
Maximum number of authors	20,285 (5.70%)	11,161 (4.70%)	21,801 (5.64%)	11,326 (4,59%)
Number of authors of the 99 percentile	6,435	4,627	6,431	4,635
Average number of authors	604	499	601	481
α (power law parameter)	1.7143	1.7992	1.6944	1.7343
δ (power law parameter)	0.0302	0.0.0382	0.0288	0.0362

Table 5.5: Parameters of the distributions of subreddits against authors

A Wilcoxon rank sum test showed that: *(i)* the number of subreddits of Jan-Feb SFW posts was statistically significantly higher than the corresponding one of NSFW posts; *(ii)* the number of authors of Jan-Feb SFW posts was statistically significantly higher than the corresponding one of NSFW posts ($\tau = 6.3 \cdot 10^{-4}, p < 0.01$).

The explanation behind this result is essentially related to the fact that NSFW posts have particular contents that are of interest to a minority of people. Therefore, they are published in a limited number of subreddits.

In the next analyses, to save space, we will avoid highlighting those cases where the values α and δ of power law distributions are similar, as well as those cases where the parameter values are stable when switching from Jan-Feb to Mar-Apr. Only if one or both of these conditions are not valid in some analysis, we will explicitly highlight this situation.

Investigating the comments to NSFW posts. In this section, we analyze the comments to NSFW posts investigating their authors, the scores they get and the subreddits they are submitted to. Firstly, we present the distributions of comments against SFW posts and NSFW posts, which follow a power law. Table 5.6 shows the values of the main parameters of these distributions.

Parameter	*SFW posts Jan-Feb*	*NSFW posts Jan-Feb*	*SFW posts Mar-Apr*	*NSFW posts Mar-Apr*
Maximum number of posts	499,068 (2.29%)	667,942 (5.79%)	522,477 (2.94%)	676,606 (5.81%)
Number of posts of the 99 percentile	8,257	10,707	8,362	10,719
Maximum number of comments	41,478 (39.93%)	28,227 (53.43%)	36,283 (40.01%)	23,485 (51.32%)
Number of comments of the 99 percentile	10,582	21,983	9,985	22,735
Average number of comments	1,237	771	1,402	656
α (power law parameter)	1.4836	1.3990	1.4779	1.4353
δ (power law parameter)	0.0178	0.0304	0.0160	0.0291

Table 5.6: Parameters of the distributions of comments against posts

A Wilcoxon rank sum test showed that the number of comments of Jan-Feb SFW posts was statistically significantly higher than the corresponding one of NSFW posts ($\tau = 8.68 \cdot 10^{-5}, p < 0.01$).

As a further investigation on this topic, we considered both the top 150 most commented SFW and NSFW posts. As a first analysis, we observed that SFW (resp., NSFW) posts have been submitted by 141 (resp., 130) authors in 55 (resp., 77) different subreddits. This result highlights that there is no author or subreddit able to monopolize post comments. Indeed, the phenomenon is highly distributed.

Then, we computed the distributions of the number of these comments against subreddits. They are reported in Figure 5.3. Plots (a) and (b) of this figure show that the two distributions follow a power law. We computed the parameter values of these power laws and we obtained $\alpha = 3.41$ and $\delta = 0.075$ for SFW post comments, and $\alpha = 3.53$ and $\delta = 0.07$ for NSFW post comments. A Wilcoxon rank sum test indicated that the number of comments associated with the subreddits containing Jan-Feb SFW posts was statistically significantly higher than the corresponding one of NSFW posts ($\tau = 0.16493, p < 0.01$).

Finally, we computed the distribution of the number of these comments against authors. Also in this case, we found that it follows a power law. The values of the corresponding parameters are $\alpha = 3.06$ and $\delta = 0.03$ for SFW post comments and $\alpha = 2.20$ and $\delta = 0.03$ for NSFW post comments. The conclusions about the trend and the values are analogous to the previous ones.

A Wilcoxon rank sum test indicated that the number of comments for Jan-Feb SFW posts was statistically significantly higher than the corresponding one of NSFW posts ($\tau = 0.34951$, $p < 0.01$).

The motivations behind this result are the same as those related to the distribution of the subreddits against authors.

We then computed the distributions of subreddits against the comments to SFW and NSFW posts. In both cases we obtained that they follow a power law and show

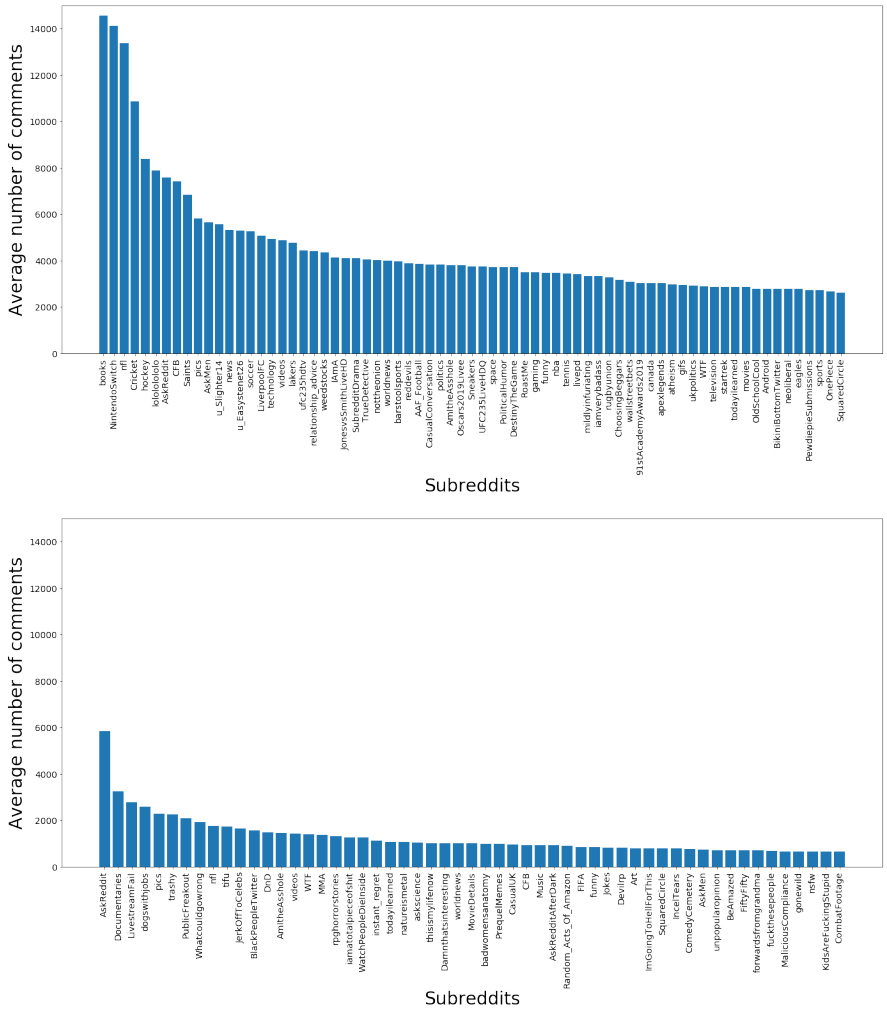


Fig. 5.3: Distributions of comments to the top 150 most commented SFW posts (on top) and NSFW posts (on bottom) against subreddits - Datasets regarding January and February 2019

trends similar to those shown in the previous figures. The main parameters of these distributions are reported in Table 5.7.

Parameter	*SFW posts Jan-Feb*	*NSFW posts Jan-Feb*	*SFW posts Mar-Apr*	*NSFW posts Mar-Apr*
Maximum number of comments	484,792 (5.45%)	301,040 (9.17%)	462,415 (5.41%)	244,912 (9.73%)
Number of comments of the 99 percentile	47,590	25,056	47,698	28,635
Average number of comments	3,942	2,607	3,800	2,391
α (power law parameter)	1.8025	1.7659	1.7981	1.7507
δ (power law parameter)	0.0236	0.0235	0.0217	0.0310

Table 5.7: Parameters of the distributions of subreddits against comments

A Wilcoxon rank sum test showed that the number of comments associated with the subreddits containing Jan-Feb SFW posts was statistically significantly higher than the corresponding one of NSFW posts ($\tau = 6.34 \cdot 10^{-6}, p < 0.01$).

Once again, the motivations behind this result are the same as those related to the distribution of the subreddits against authors.

Moreover, we computed the distributions of comments to SFW and NSFW posts against scores. They are reported in Figures 5.4 and 5.5 for the datasets $\mathcal{D}$ and $\overline{\mathcal{D}}$. These figures show that the corresponding distributions do not follow a power law, and this is the first case. As we can see from figures, the distributions are irregular, even if both of them seem having a Gaussian trend.

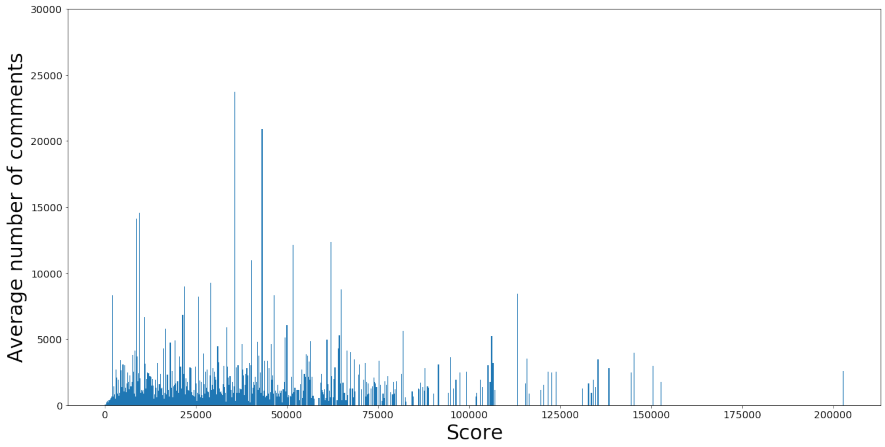


Fig. 5.4: Distribution of comments to SFW posts against scores - Datasets regarding January and February 2019

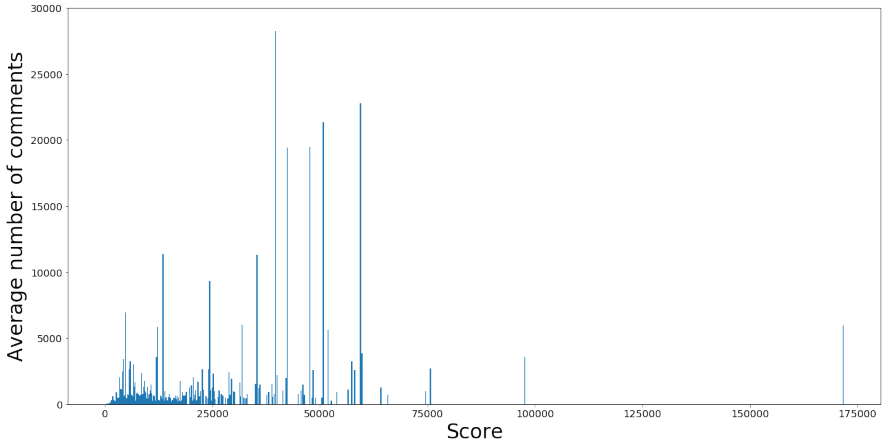


Fig. 5.5: Distribution of comments to NSFW posts against scores - Datasets regarding January and February 2019

Also in this case, we computed some parameters for the two distributions. They are shown in Table 5.8.

Parameter	*SFW posts Jan-Feb*	*NSFW posts Jan-Feb*	*SFW posts Mar-Apr*	*NSFW posts Mar-Apr*
Average score	9,881	4,191	8,809	3,819
Score of the last comment of the first quartile	2,035	1,157	1,993	1,215
Score of the last comment of the second quartile	4,686	2,357	4,551	2,484
Score of the last comment of the third quartile	11,106	4,486	9,953	4,667
Score of the last comment of the fourth quartile	202,696	69,591	209,154	71,566

Table 5.8: Parameters of the distributions of comments to posts against scores

A Wilcoxon rank sum test indicated that the score of comments for Jan-Feb SFW posts was statistically significantly higher than the corresponding one of NSFW posts ($\tau = 5.88 \cdot 10^{-5}, p < 0.01$).

The motivations behind this result are the same as those related to the distribution of the posts against scores.

A deeper analysis of the stability of the investigations. All the distributions we have seen so far are based on a data sample recovered from January 1^{st}, 2019 to September 1^{st}, 2019. Due to computational complexity reasons, we could not process the whole sample at the same time and, therefore, we divided it into bi-months, i.e. Jan-Feb and Mar-Apr. In all the distributions we have presented so far, we could verify that the Jan-Feb and Mar-Apr data led to very similar results. This is a strong remark of the stability of the results of our investigations.

However, before continuing with the next analyses, which will have an even higher computational complexity, we decided to carry out a further stability check. To this end, we considered all the posts published in Reddit from January 1^{st}, 2019 to December 31^{st}, 2019, and split them months by months. Then, for each month, we computed several parameters previously seen for the two bi-months. The results obtained are shown in Table 5.9 for SFW posts, and in Table 5.10 for NSFW posts. The analysis of these tables fully confirms that the results of our investigations are stable.

5.1.2 Results

Co-posting activity of NSFW posts authors. The goal of this analysis is to verify whether there is any correlation between the authors of NSFW posts. As shown previously, we will extract the information of interest and we will compare the behavior of authors of NSFW posts with the ones of SFW posts. In this activity, we will use

Parameter	*Jan*	*Feb*	*Mar*	*Apr*	*May*	*Jun*
GENERAL CHARACTERISTICS						
Number of authors who published at least one SFW post	391,898	387,458	365,785	389,154	387,562	374,531
Number of authors who published only SFW posts	380,261	374,564	359,851	378,582	377,423	365,751
Percentage of authors publishing SFW posts who published only posts of this type	97.03%	96.67%	98.37%	97.28%	97.38%	97.65%
Number of subreddits containing at least one SFW post	58,843	57,965	58,786	57,653	58,426	57,953
Number of subreddits containing only SFW posts	54,189	53,482	53,952	54,236	54,873	52,432
Percentage of subreddits containing SFW posts that contain only posts of this type	92.09%	92.22%	91.77%	94.07%	93.91%	90.47%
DISTRIBUTION OF SUBREDDITS AGAINST POSTS						
Maximum number of subreddits	47,480	47,116	47,996	49,502	48,294	47,733
Maximum number of posts	25,006	23,746	26,055	26,650	28,743	24,211
α (power law parameter)	1.6321	1.5806	1.7512	1.8358	1.6293	1.7024
δ (power law parameter)	0.0256	0.0238	0.0362	0.0357	0.0263	0.029
DISTRIBUTION OF AUTHORS AGAINST POSTS						
Maximum number of authors	555,854	559,602	566,139	540,511	551,863	541,585
Maximum number of posts	18,724	17,401	18,268	16,513	17,226	19,949
α (power law parameter)	1.4531	1.6718	1.3565	1.399	1.5478	1.3742
δ (power law parameter)	0.0465	0.0359	0.0545	0.0233	0.0428	0.0757
DISTRIBUTION OF POSTS AGAINST SCORES						
Maximum score	183,453	185,056	180,553	191,864	180,578	179,099
α (power law parameter)	1.5986	1.631	1.4672	1.6026	1.6507	1.5681
δ (power law parameter)	0.0189	0.0186	0.0198	0.0086	0.0179	0.0359
DISTRIBUTION OF SUBREDDITS AGAINST AUTHORS						
Maximum number of subreddits	62,839	65,934	70,585	65,861	63,087	62,325
Maximum number of authors	20,285	19,571	18,808	21,801	20,029	19,801
α (power law parameter)	1.7185	1.7064	1.6209	1.608	1.7013	1.7853
δ (power law parameter)	0.0298	0.0485	0.0315	0.02	0.0379	0.0327

Parameter	*Jul*	*Ago*	*Sep*	*Oct*	*Nov*	*Dec*
GENERAL CHARACTERISTICS						
Number of authors who published at least one SFW post	59,465	60,563	59,489	59,873	58,985	60,236
Number of authors who published only SFW posts	58,801	59,423	58,965	58,742	58,632	59,542
Percentage of authors publishing SFW posts who published only posts of this type	98.88%	98.11%	99.11%	98.11%	99.40%	98.84%
Number of subreddits containing at least one SFW post	89,360	87,953	89,236	88,462	87,932	88,167
Number of subreddits containing only SFW posts	82,050	82,587	85,496	83,647	83,146	84,963
Percentage of subreddits containing SFW posts that contain only posts of this type	91.82%	90.74%	93.68%	91.76%	91.7%	94.4%
DISTRIBUTION OF SUBREDDITS AGAINST POSTS						
Maximum number of subreddits	46,283	46,882	48,777	47,676	48,886	47,070
Maximum number of posts	22,261	19,071	23,642	29,330	26,346	28,419
α (power law parameter)	1.582	1.8481	1.7838	1.7313	1.5937	1.5125
δ (power law parameter)	0.0186	0.0305	0.0535	0.0329	0.0468	0.0154
DISTRIBUTION OF AUTHORS AGAINST POSTS						
Maximum number of authors	541,585	574,678	542,568	569,611	576,835	556,736
Maximum number of posts	16,823	19,320	18,692	18,460	16,499	17,766
α (power law parameter)	1.3323	1.406	1.4688	1.4054	1.3093	1.525
δ (power law parameter)	0.0713	0.0491	0.0561	0.0424	0.064	0.038
DISTRIBUTION OF POSTS AGAINST SCORES						
Maximum score	194,305	176,975	164,394	186,004	172,001	177,739
α (power law parameter)	1.5089	1.5785	1.4772	1.6389	1.4331	1.6354
δ (power law parameter)	0.0114	0.054	0.0245	0.0389	0.0226	0.0012
DISTRIBUTION OF SUBREDDITS AGAINST AUTHORS						
Maximum number of subreddits	59,963	57,573	59,898	52,885	62,111	63,232
Maximum number of authors	18,901	20,056	20,285	19,962	21,078	20,909
α (power law parameter)	1.7622	1.6287	1.4544	1.8174	1.5256	1.7388
δ (power law parameter)	0.0159	0.0263	0.043	0.0254	0.0184	0.0378

Table 5.9: Monthly trend of some parameters related to SFW posts

a support data structure that we call *co-posting network*. Having observed in all the previous experiments that the results obtained for the Jan-Feb datasets (i.e., $\mathcal{D}$ and $\overline{\mathcal{D}}$) are stable, from now on we will refer to these two datasets only, avoiding to report the analysis of Mar-Apr datasets, too. In addition, since most of the operations that we will perform on the co-posting network are computationally expensive, we

Parameter	*Jan*	*Feb*	*Mar*	*Apr*	*May*	*Jun*
GENERAL CHARACTERISTICS						
Number of authors who published at least one NSFW post	36,758	35,452	36,542	36,874	36,863	36,453
Number of authors who published only NSFW posts	36,094	35,259	36,501	36,165	36,135	36,023
Percentage of authors publishing NSFW posts who published only posts of this type	98.19%	99.45%	99.88%	98.07%	98.02%	98.82%
Number of subreddits containing at least one NSFW post	41,365	40,985	41,298	41,547	41,235	40,958
Number of subreddits containing only NSFW posts	34,055	33,254	34,587	32,982	33,563	34,159
Percentage of subreddits containing NSFW posts that contain only posts of this type	82.33%	81.13%	83.74%	79.38%	81.39%	83.40%
DISTRIBUTION OF SUBREDDITS AGAINST POSTS						
Maximum number of subreddits	18,332	17,985	19,547	21,034	20,135	20,235
Maximum number of posts	34,424	32,547	31,854	31,329	30,896	32,541
α (power law parameter)	1.6896	1.6721	1.6874	1.6852	1.6796	1.6852
δ (power law parameter)	0.0258	0.0254	0.0251	0.0254	0.0214	0.0261
DISTRIBUTION OF AUTHORS AGAINST POSTS						
Maximum number of authors	131,070	130,152	131,250	133,594	131,452	132,654
Maximum number of posts	16,383	16,125	14,214	15,674	16,540	14,210
α (power law parameter)	1.5463	1.7985	1.6222	1.8407	1.9456	1.4833
δ (power law parameter)	0.03345	0.0233	0.0239	0.0639	0.0388	0.0458
DISTRIBUTION OF POSTS AGAINST SCORES						
Maximum score	106,947	146,561	75,657	112,830	105,566	66,095
α (power law parameter)	1.6062	1.5162	1.6933	1.8989	1.6951	1.4956
δ (power law parameter)	0.0145	0.0265	0.042	0.0611	0.0346	0.0139
DISTRIBUTION OF SUBREDDITS AGAINST AUTHORS						
Maximum number of subreddits	62,839	63,382	61,204	33,963	50,609	53,781
Maximum number of authors	20,285	17,549	19,347	11,326	18,495	19,324
α (power law parameter)	1.7156	1.7682	1.6166	1.9204	1.753	1.6321
δ (power law parameter)	0.0312	0.0241	0.0384	0.0236	0.0187	0.0418

Parameter	*Jul*	*Ago*	*Sep*	*Oct*	*Nov*	*Dec*
GENERAL CHARACTERISTICS						
Number of authors who published at least one NSFW post	37,165	35,986	36,432	36,540	36,354	36,589
Number of authors who published only NSFW posts	36,984	35,421	35,962	35,986	35,756	35,852
Percentage of authors publishing NSFW posts who published only posts of this type	99.51%	98.42%	98.77%	98.48%	98.35%	97.98%
Number of subreddits containing at least one NSFW post	41,542	40,986	41,246	41,258	40,983	41,496
Number of subreddits containing only NSFW posts	34,478	33,352	34,254	34,165	33,241	33,986
Percentage of subreddits containing NSFW posts that contain only posts of this type	82.99%	81.37%	83.04%	82.80%	81.10%	81.90%
DISTRIBUTION OF SUBREDDITS AGAINST POSTS						
Maximum number of subreddits	20,135	18,564	17,423	19,631	18,328	20,124
Maximum number of posts	30,451	32,598	30,125	29,874	34,210	32,021
α (power law parameter)	1.6236	1.6454	1.59874	1.6598	1.6432	1.6953
δ (power law parameter)	0.0265	0.0259	0.0298	0.0265	0.0264	0.0254
DISTRIBUTION OF AUTHORS AGAINST POSTS						
Maximum number of authors	130,254	134,250	133,247	132,478	136,587	131,489
Maximum number of posts	16,125	14,256	15,879	16,325	14,369	16,362
α (power law parameter)	1.6992	1.4551	1.5295	1.5527	1.5524	1.6091
δ (power law parameter)	0.0446	0.048	0.0201	0.0268	0.0031	0.0428
DISTRIBUTION OF POSTS AGAINST SCORES						
Maximum score	97,462	143,430	102,590	100,844	104,027	81,167
α (power law parameter)	1.6422	1.5874	1.4948	1.7059	1.7936	1.3969
δ (power law parameter)	0.040	0.028	0.0386	0.0324	0.0184	0.0354
DISTRIBUTION OF SUBREDDITS AGAINST AUTHORS						
Maximum number of subreddits	49,210	76,791	64,241	54,351	50,864	34,037
Maximum number of authors	17,425	20,605	23,952	20,608	18,613	16,594
α (power law parameter)	1.7653	1.7342	1.5258	1.9738	1.6143	1.5882
δ (power law parameter)	0.0317	0.037	0.0204	0.0371	0.0207	0.0401

Table 5.10: Monthly trend of some parameters related to NSFW posts

randomly extracted a subset $\mathcal{D}^*$ (resp., $\overline{\mathcal{D}}^*$) of $\mathcal{D}$ (resp., $\overline{\mathcal{D}}$) consisting of 75,000 SFW (resp., NSFW) posts to work on.

As a first task of this analysis, we give a formal definition of the co-posting network $\mathcal{P}$ (resp., $\overline{\mathcal{P}}$) built from the authors of SFW (resp., NSFW) posts stored in $\mathcal{D}^*$ (resp., $\overline{\mathcal{D}}^*$).

Formally speaking,

$$\mathcal{P} = \langle N, E \rangle \qquad \overline{\mathcal{P}} = \langle \overline{N}, \overline{E} \rangle$$

Here, N (resp., $\overline{N}$) is the set of the nodes of $\mathcal{P}$ (resp., $\overline{\mathcal{P}}$). There is a node $n_i \in N$ (resp., $\overline{N}$) for each author a_i of SFW (resp., NSFW) posts of $\mathcal{D}^*$ (resp., $\overline{\mathcal{D}}^*$). There is an edge $(n_i, n_j, w_{ij}) \in E$ (resp., $\overline{E}$) if the authors a_i and a_j (associated with n_i and n_j, respectively) submitted at least one post in the same subreddit. w_{ij} is the number of subreddits having at least one SFW (resp., NSFW) post of a_i and, simultaneously, at least one SFW (resp., NSFW) post of a_j.

Then, we calculated some of the basic parameters of $\mathcal{P}$ and $\overline{\mathcal{P}}$; they are shown in Table 5.11. From the analysis of this table, we can deduce that:

- The number of co-posting authors of NSFW posts is smaller than the number of co-posting authors of SFW posts.
- The authors of NSFW posts are more interconnected with each other. This is shown by both the density of $\overline{\mathcal{P}}$ (which is about three times the one of $\mathcal{P}$) and the average degree of $\overline{\mathcal{P}}$ (which is much greater than twice the degree of $\mathcal{P}$). As we will see in the following, this can be explained considering that they are authors belonging to a niche context.
- The average clustering coefficient of $\overline{\mathcal{P}}$ is greater than the one of $\mathcal{P}$, but not as much as the density. This suggests that in $\overline{\mathcal{P}}$ fewer triads are closed than in $\mathcal{P}$. This implies that, probably, in $\overline{\mathcal{P}}$ there are more "bridge" authors than in $\mathcal{P}$. These authors tend to act as intermediaries between other authors who do not know each other. They could be expert authors who cooperate with many new authors initially unknown to each other.

Parameter	$\mathcal{P}$	$\overline{\mathcal{P}}$
Number of nodes	59,465	36,758
Number of edges	3,164,169	5,398,082
Density	0.001789	0.007990
Maximum Degree	2,593	3,670
Average Degree	106.42	293.70
Average Clustering Coefficient	0.7388	0.7755

Table 5.11: Basic parameters of the co-posting networks $\mathcal{P}$ and $\overline{\mathcal{P}}$

After this, we computed the distribution of the nodes of $\mathcal{P}$ and $\overline{\mathcal{P}}$ against their degree centrality. The results obtained are reported in Figures 5.6 and 5.7.

From the analysis of these figures we can see that both distributions follow a power law. We computed the corresponding values of α and δ and obtained that $\alpha = 2.2929$ and $\delta = 0.0470$ for $\mathcal{P}$ and $\alpha = 2.6811$ and $\delta = 0.0678$ for $\overline{\mathcal{P}}$. These values tell us that the two distributions are similar.

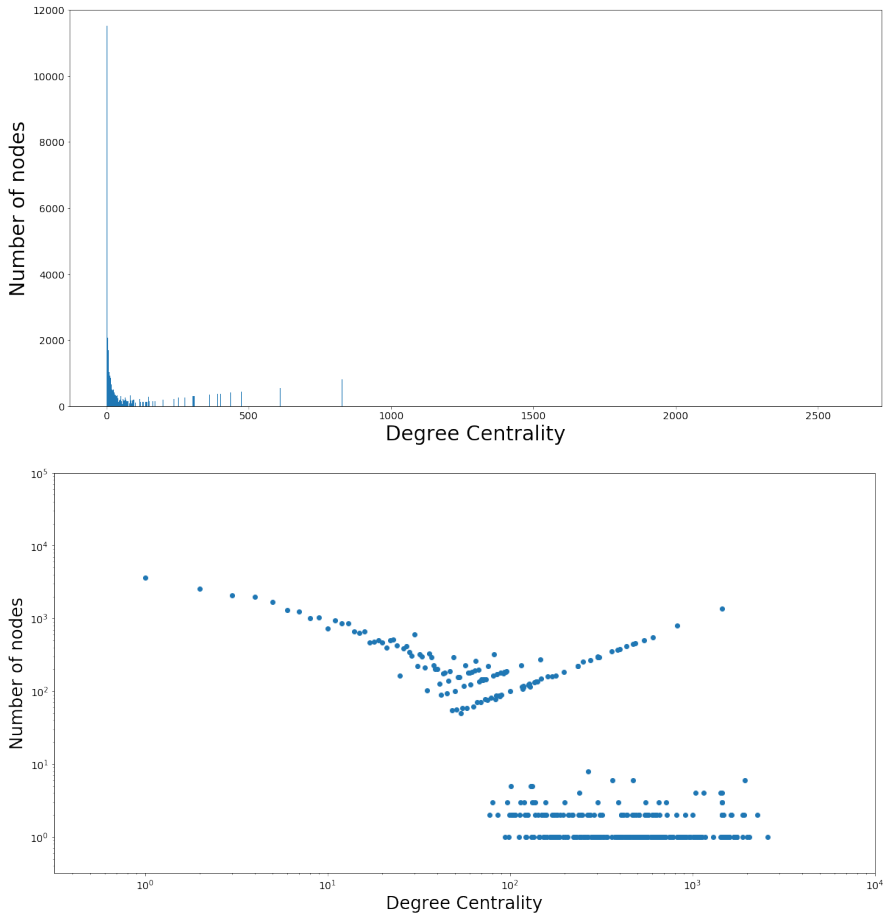


Fig. 5.6: Distribution of the nodes of $\mathcal{P}$ against their degree centrality - linear scale (on top) and log-log scale (on bottom)

Furthermore, looking carefully at the distributions in Figures 5.6 and 5.7, it emerges another unexpected, extremely peculiar, feature. In fact, we can observe some spikes. Excluding that these spikes are noise, they could be caused by the fact that the networks $\mathcal{P}$ and $\overline{\mathcal{P}}$ are actually disconnected and each network consists of a set of connected components. We found extremely interesting to check if this hypothesis was true. Therefore, we carried out this analysis and verified that, actually, we were right. In fact, we found that $\mathcal{P}$ consists of 15,952 connected components. Of these, 11,514 are made up of a single node. The maximum connected component includes 21,364 nodes (equal to 35,92% of the network nodes) and 2,909,206 arcs (equal to 91.94% of the network arcs). The distribution of the connected components against their size (i.e., the number of nodes they include) follows a power law with $\alpha = 1.562$ and $\delta = 0.060$. The network $\overline{\mathcal{P}}$ consists of 6,032 connected components, where 5,214 are made of a single node. The maximum connected component comprises 28,165 nodes (equal to 76.62% of the network's nodes) and 5,382,255 arcs

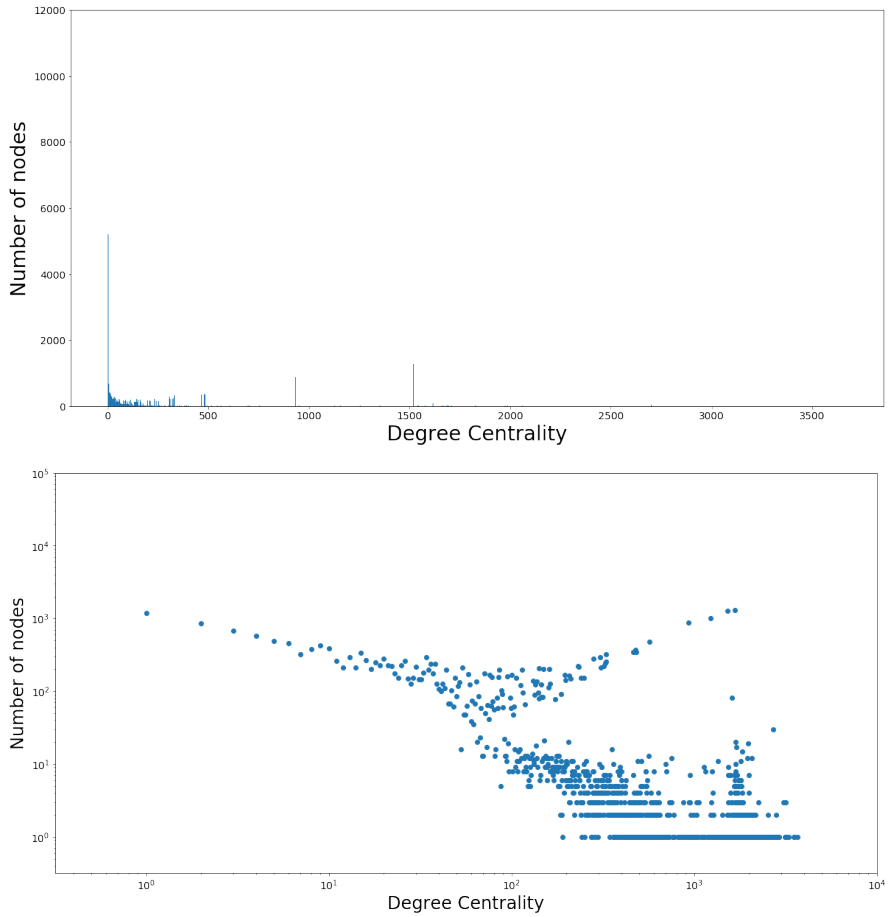


Fig. 5.7: Distribution of the nodes of $\overline{\mathcal{P}}$ against degree centrality - linear scale (on top) and log-log scale (on bottom)

(equal to 99.71% of the network's arcs). The distribution of the connected components against their size follows a power law with $\alpha = 1.548$ and $\delta = 0.065$.

The analysis of connected components strengthens some results obtained previously, in particular: *(i)* the number of co-posting authors of SFW posts is greater than the corresponding number of co-posting authors of NSFW posts; *(ii)* the authors of NSFW posts are more connected to each other (probably due to the presence of the "bridge" users mentioned above) than the ones of SFW posts.

At this point, we wanted to investigate more on the behavior of the authors of SFW and NSFW posts. Specifically, we treated three activities, namely the writing of posts, the tendency to publish on many subreddits and the ability to attract interest. For each of these activities, we selected the top-ten authors from the maximum connected component of $\mathcal{P}$ and $\overline{\mathcal{P}}$ and we studied their behavior. In particular, Figure 5.8 (resp., 5.9 and 5.10) shows the top-ten authors who wrote the highest number of posts (resp., published in the largest number of subreddits, received the highest number of comments). The left part of this figure refers to the authors of SFW posts

(belonging to the network $\mathcal{P}$), while the right part refers to the authors of NSFW posts (belonging to the network $\overline{\mathcal{P}}$).

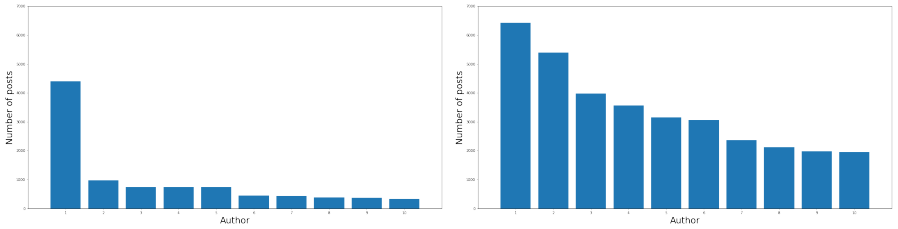


Fig. 5.8: Top-ten authors who submitted more posts - authors of SFW posts at left and of NSFW posts at right

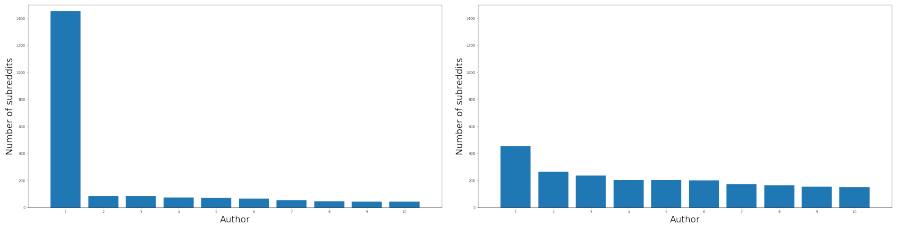


Fig. 5.9: Top-ten authors who published on more subreddits - authors of SFW posts at left and of NSFW posts at right

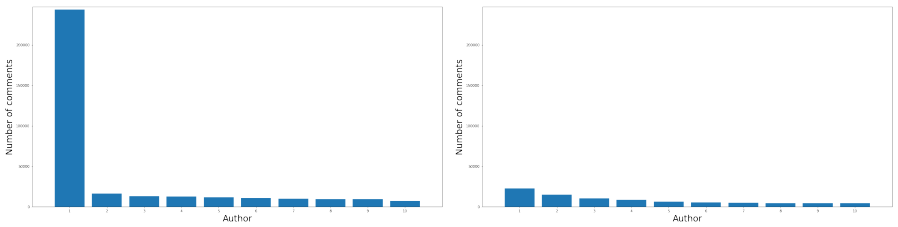


Fig. 5.10: Top-ten authors who received more comments - authors of SFW posts at left and of NSFW posts at right

These figures altogether outline a very precise author behavior. In fact, it can be noted that, regardless of the activity considered, the authors of SFW posts show a power law distribution, while the authors of NSFW posts show a very slowly decreasing distribution. This allows us to conclude that there are few very active authors of SFW posts and many inactive ones in Reddit. By contrast, there are many quite active authors of NSFW posts. Once again, it seems that these last tend to "team up" much more than the ones of SFW posts.

These results can be explained considering that the phenomenon of NSFW posts is a niche one involving mostly particular kinds of user. These are very cohesive and form a fairly closed group. On the other hand, as we will see better in Section 5.1.2, all the knowledge extracted confirms this reasoning about the context behind NSFW posts.

Evaluating assortativity of NSFW posts authors. The concept of "assortativity", or "assortative mixing", in a social network points out the predilection of its nodes to be connected with other nodes that are somehow similar to them. This concept, introduced by Newman [462], can be seen as an evolution of the concept of homophily [435], typical of Social Network Analysis. Assortativity is orthogonal to node similarity metrics considered, even if most of the authors in the literature have studied it with respect to node degree. According to this definition of assortativity, the nodes of a social network tend to be linked with other nodes having a degree similar to their own.

Assortativity is considered an extremely important property to be investigated by social network researchers. So we decided to analyze it for the authors of SFW and NSFW posts in Reddit. We would also pinpoint that: *(i)* like in the previous analyses reported above, the goal is to characterize the assortativity of the authors of NSFW posts versus the one of the authors of SFW posts; *(ii)* the similarity property we decided to test for assortativity is node degree, because it is the most investigated one in the past literature on assortativity[2].

To carry out our assortativity analyses, we used the co-posting networks $\mathcal{P}$ and $\overline{\mathcal{P}}$ defined in Section 5.1.2. We showed the distributions of the nodes of these networks against degree centrality in Figures 5.6 and 5.7. As a first task, we sorted the authors of the two networks in descending order of degree centrality. After that, we split this ordered list into intervals. In particular, we considered 40 equi-width intervals $\{\mathcal{I}_1, \mathcal{I}_2, \cdots, \mathcal{I}_{40}\}$ for $\mathcal{P}$ and $\{\overline{\mathcal{I}}_1, \overline{\mathcal{I}}_2, \cdots, \overline{\mathcal{I}}_{40}\}$ for $\overline{\mathcal{P}}$. Since the number of nodes of $\mathcal{P}$ (resp., $\overline{\mathcal{P}}$) was 59,465 (resp., 36,578), each interval $\mathcal{I}_k$ (resp., $\overline{\mathcal{I}}_k$) contained 1,487 (resp., 915) authors[3].

At this point, we considered the interval $\mathcal{I}_1$ (resp., $\overline{\mathcal{I}}_1$) and, for each interval $\mathcal{I}_k$ (resp., $\overline{\mathcal{I}}_k$), we determined how many authors of $\mathcal{I}_1$ (resp., $\overline{\mathcal{I}}_1$) were connected to at least one author of $\mathcal{I}_k$ (resp., $\overline{\mathcal{I}}_k$). The results obtained are shown in Figure 5.11(a) (resp., 5.11(c)). Next, we computed the percentage of the authors of $\mathcal{I}_k$ (resp., $\overline{\mathcal{I}}_k$),

[2] Actually, at the end of this section, for a further evidence of the results obtained, we also considered eigenvector centrality, beside degree centrality.

[3] Actually, the last interval had a slightly smaller size equal to 1,472 (resp., 893) authors.

who were connected to at least one author of $\mathcal{I}_1$ (resp., $\overline{\mathcal{I}}_1$). The results obtained are shown in Figure 5.11(e) (resp., 5.11(g)).

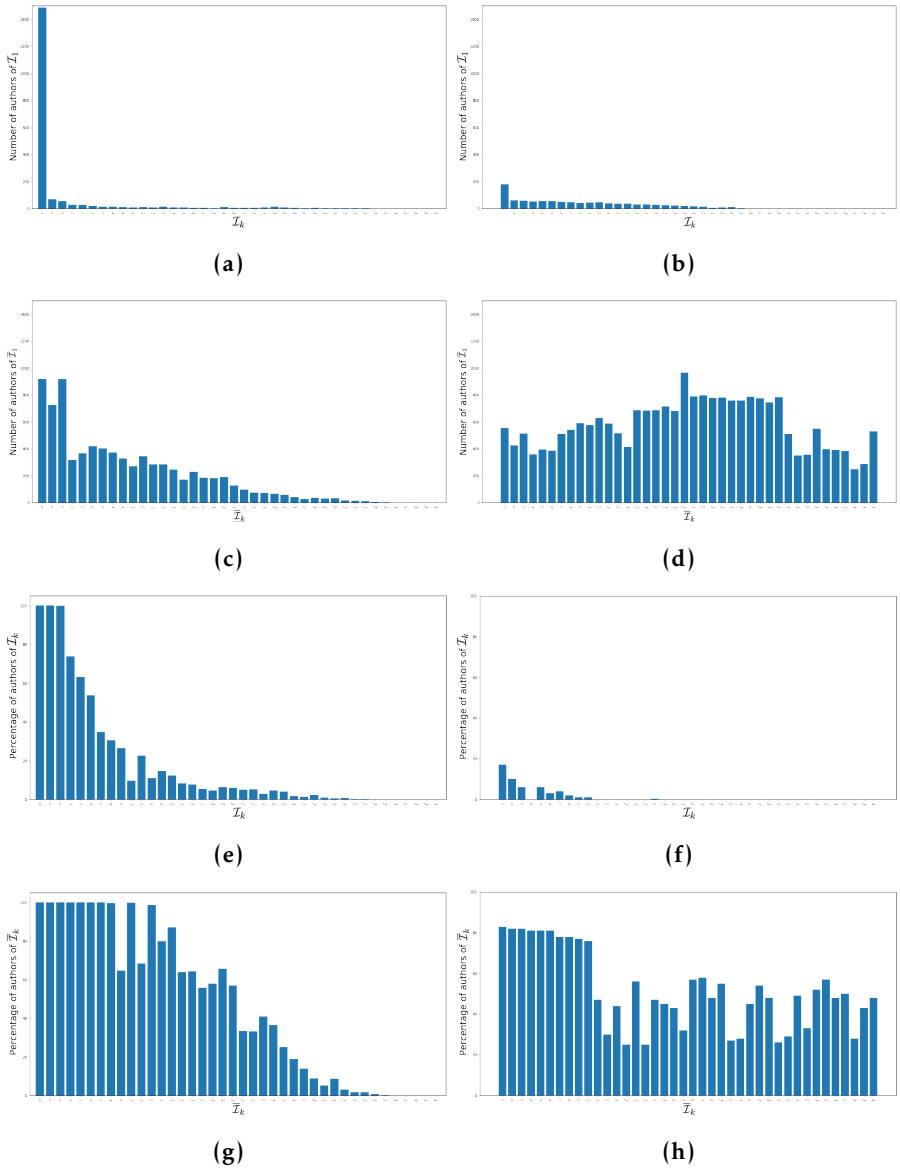


Fig. 5.11: Degree Assortativity of the authors of NSFW and SFW posts (high degree authors)

The analysis of Figures 5.11(a) and 5.11(e) shows a close correlation (i.e., a sort of backbone) between the authors of SFW posts with the highest degree centrality. On the contrary, the analysis of Figures 5.11(c) and 5.11(g) shows that this phenomenon does not occur for the authors of NSFW posts.

In order to evaluate the statistical significance of this result, we generated a null model to compare our outcomes with those of an unbiasedly random scenario. In particular, we built our null model shuffling the arcs of $\mathcal{P}$ (resp. $\overline{\mathcal{P}}$) among the nodes

of this network. In this way, we left the original characteristics of $\mathcal{P}$ (resp. $\overline{\mathcal{P}}$) unchanged, except for the distribution of co-posting activities, which became unbiasedly random in the null model. The results obtained are shown in Figures 5.11(b), 5.11(d), 5.11(f) and 5.11(h).

Comparing Figures 5.11(b) and 5.11(f) with Figures 5.11(a) and 5.11(e) we can see that the represented distributions are similar. Indeed, many of the ranges with the highest values of Figures 5.11(a) and 5.11(e) continue to reach the highest values in Figures 5.11(b) and 5.11(f), too. However, these values are much smaller in the latter case. Therefore, we can conclude that the behavior observed in Figures 5.11(a) and 5.11(e) is not random, but intrinsic to $\mathcal{P}$ (and, therefore, to the authors of SFW posts in Reddit). On the contrary, if we consider Figures 5.11(c) and 5.11(g) (regarding the authors of NSFW posts in Reddit) and compare them with Figures 5.11(d) and 5.11(h), we can see that this phenomenon does not occur for the authors of $\overline{\mathcal{P}}$.

The above analysis suggests that there is a degree assortativity among the authors of SFW posts but not among the authors of NSFW posts. However, in order to confirm the assortativity of the authors of SFW posts, we need to verify whether this trend is still valid for the authors with an intermediate degree centrality and for those with a low degree centrality. If we want to make an exhaustive analysis, we should repeat the tasks previously performed for $\mathcal{I}_1$ (resp., $\overline{\mathcal{I}}_1$) for all the 40 intervals. For lack of space, we will limit our analysis to the intervals $\mathcal{I}_{20}$ (resp., $\overline{\mathcal{I}}_{20}$), as the representative of those with intermediate degree centrality, and $\mathcal{I}_{30}$ (resp., $\overline{\mathcal{I}}_{30}$), as the representative of those with low degree centrality[4].

Figure 5.12(a) (resp., 5.12(c)) shows the number of authors of $\mathcal{I}_{20}$ (resp., $\overline{\mathcal{I}}_{20}$) connected with at least one author of $\mathcal{I}_k$ (resp., $\overline{\mathcal{I}}_k$), while Figure 5.12(e) (resp., 5.12(g)) shows the percentage of authors of $\mathcal{I}_k$ (resp., $\overline{\mathcal{I}}_k$) connected with at least one author of $\mathcal{I}_{20}$ (resp., $\overline{\mathcal{I}}_{20}$). The analysis of these figures suggests the existence of a close correlation among the authors of SFW posts with an intermediate degree of centrality; this correlation does not exist for the authors of NSFW posts.

Even in this case, we compared these findings with those obtained in the null model. The latter are shown in Figures 5.12(b), 5.12(d), 5.12(f) and 5.12(h). Looking at all the diagrams reported in Figure 5.12, once again we can conclude that the observed behavior is not random, but it is a property of Reddit.

[4] We did not choose the intervals $\mathcal{I}_k$ (resp., $\overline{\mathcal{I}}_k$), $k > 30$, because, during the analysis of the connected components, we saw that there is a high number of isolated nodes in $\mathcal{P}$ (resp., $\overline{\mathcal{P}}$) - see Section 5.1.2. Clearly, these nodes belong to the highest intervals and, if considered, could represent a bias in our analysis. To avoid this bias, we chose to not consider the intervals where they reside, and to select $\mathcal{I}_{30}$ (resp., $\overline{\mathcal{I}}_{30}$) as the representative of the intervals with low degree centrality.

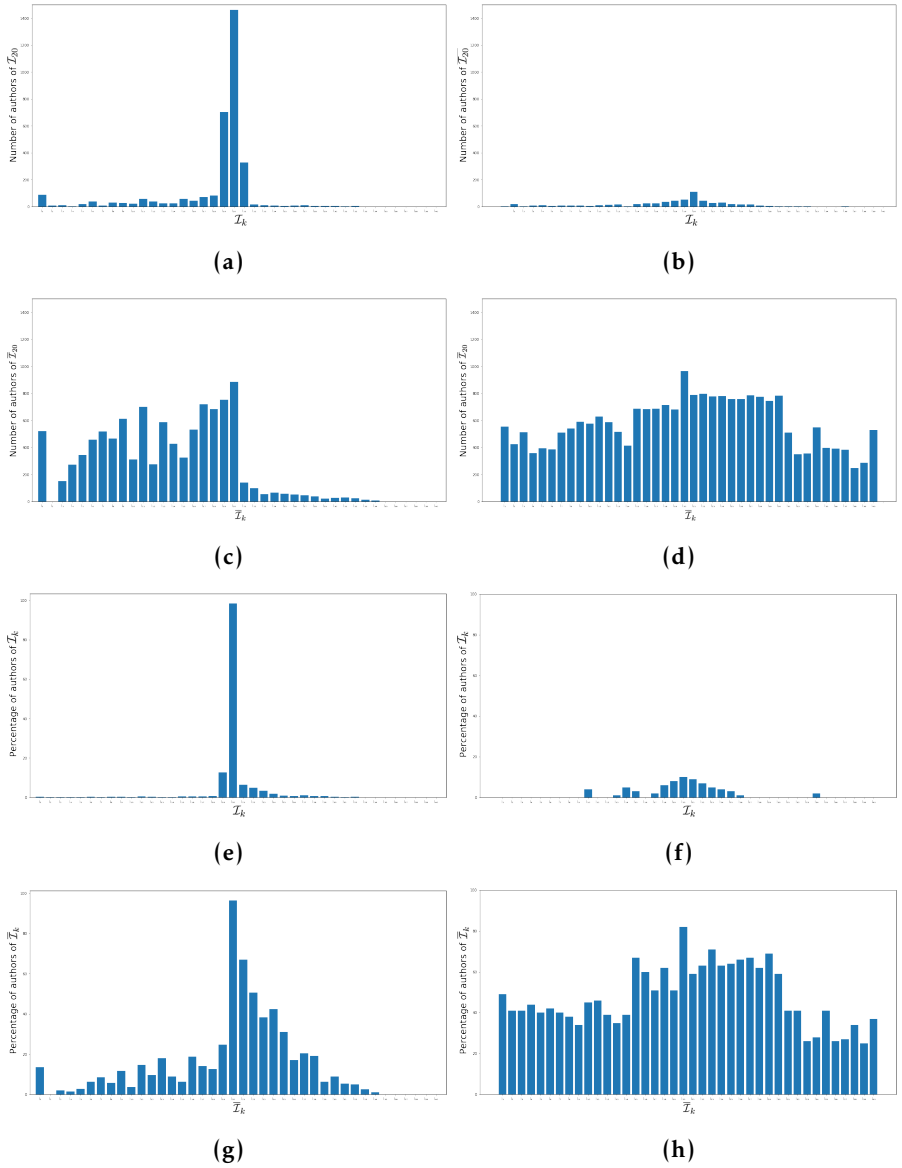


Fig. 5.12: Degree Assortativity of the authors of NSFW and SFW posts (medium degree authors)

In the light of the last observation and of the previous conclusions on authors with an intermediate and a high degree centrality, we can certainly assert that there is no degree assortativity for the authors of NSFW posts. Instead, the possibility that such assortativity exists for the authors of SFW posts remains open.

In order to verify this last possibility, we carried out a study on the authors of $\mathcal{I}_{30}$. Figure 5.13(a) shows the number of authors of $\mathcal{I}_{30}$ connected to at least one author of $\mathcal{I}_k$, while Figure 5.13(c) shows the percentage of authors of $\mathcal{I}_k$ connected to at least one author of $\mathcal{I}_{30}$. These figures reveal the presence of a close correlation between the authors of SFW posts with a low degree centrality.

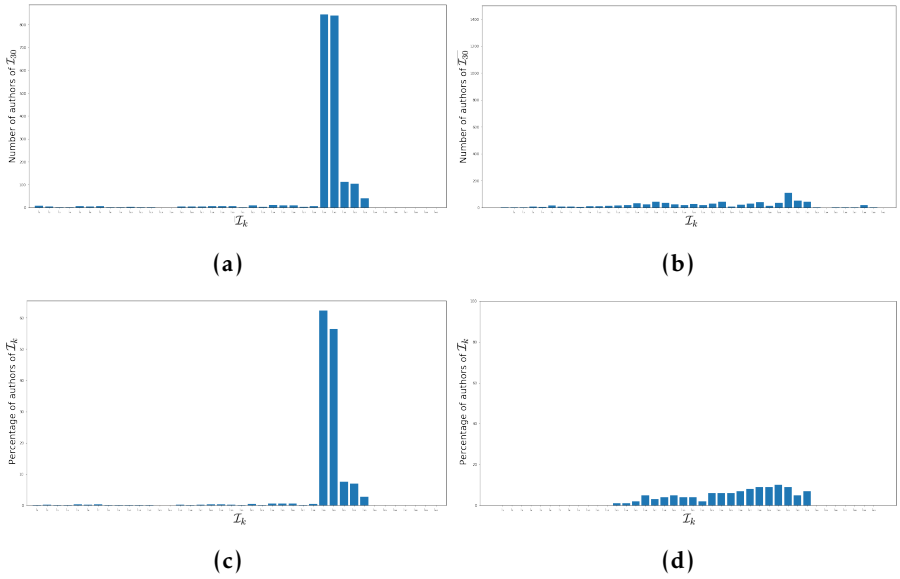


Fig. 5.13: Degree Assortativity of the authors of SFW posts (low degree authors)

Even in this case, we compared the results obtained with those returned using the null model. We report the latter in Figures 5.13(b) and 5.13(d). The comparison of these figures with Figures 5.13(a) and 5.13(c) confirms that the behavior observed for these authors is an intrinsic property of Reddit.

Having verified that there is a sort of backbone among the authors of SFW posts with high (resp., medium, low) degree centrality, we can conclude that there is a degree assortativity for the authors of SFW posts in Reddit. Instead, this property is absent for the authors of NSFW posts in Reddit.

A further interesting analysis is to check if the tendency of the authors of SFW posts to be assortative and the tendency of the authors of NSFW posts to be not assortative is general or strongly depends on the type of assortativity that is being considered (in this case, degree assortativity).

As a premise to this discussion, it should be pointed out that every form of assortativity is independent, so it is impossible to come to a *general rule*. However, the analysis previously mentioned could surely lead us to discover some *trends*.

Therefore, we chose a second form of centrality (in particular, the eigenvector centrality) and we repeated all the steps previously taken for degree centrality with this second one.

The results obtained are very similar to those we have seen for degree centrality, i.e., we found the existence of a strong eigenvector assortativity for the authors of SFW posts and a lack of eigenvector assortativity for the authors of NSFW posts. For space reasons, we cannot show all the results. However, in order to give an idea of them, in Figure 5.14, we report what happens for authors with high eigenvector

centrality. Comparing this figure with Figure 5.11, we can observe a strong similarity in the authors behavior in the two cases. As a consequence, we can say that SFW authors *tend* to be assortative, while NSFW authors *tend* to be not assortative.

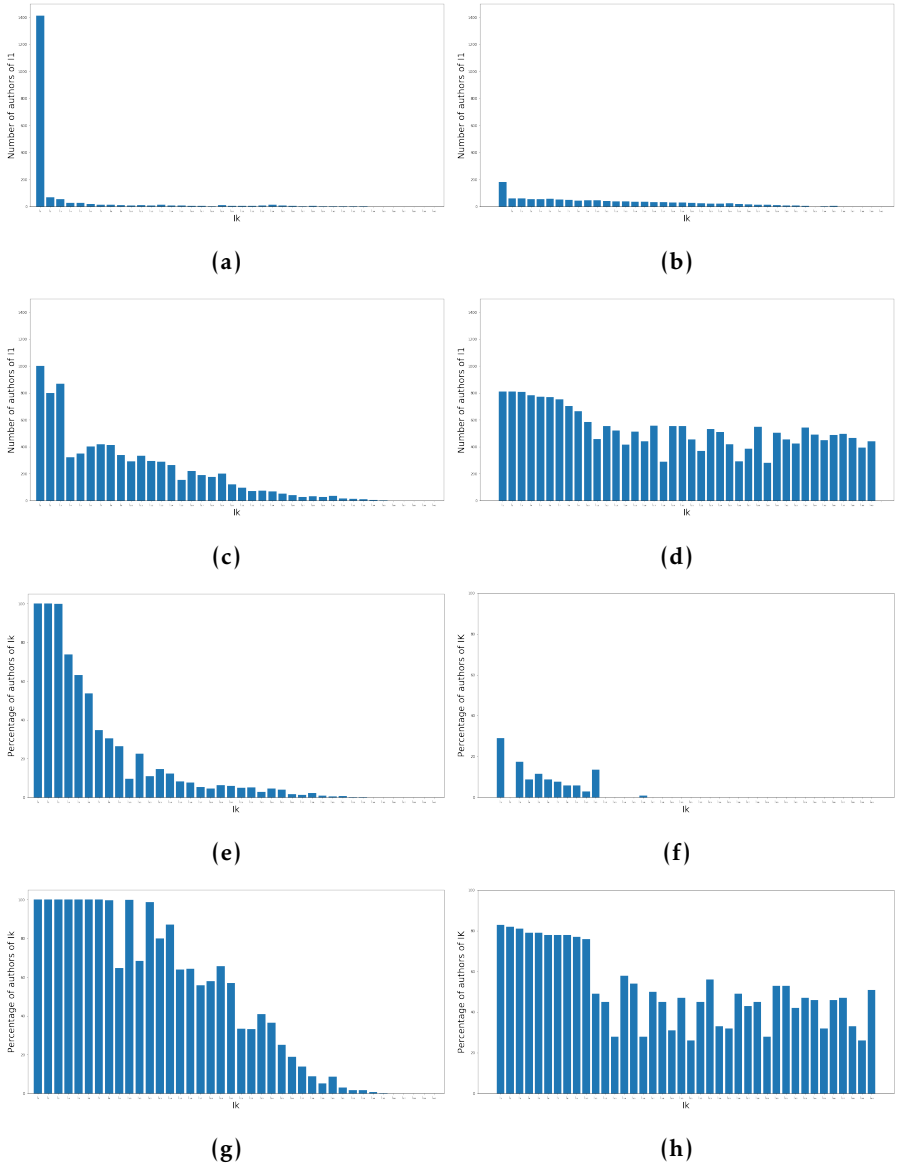


Fig. 5.14: Eigenvector Assortativity of the authors of NSFW and SFW posts (high degree authors)

This result can be explained by the strong community sense of the authors of NSFW posts. They are so cohesive that they do not feel the need to split into groups of peers. The most active people are still willing to interact with everyone else and not only with other equally active people.

Knowledge findings on posts, authors and subreddits. Combining together all the previous results, we can define three main findings related to posts, authors and subreddits, respectively. Some of these findings are made up of several sub-findings.

The three findings are the following:

PF (Finding on NSFW posts)

1. NSFW posts are generally published in much fewer subreddits, have much lower scores and are much less commented than SFW posts.
2. The scores of comments to NSFW posts are much lower than the ones to SFW posts.

AF (Finding on NSFW authors)

1. NSFW authors tend: (i) to publish more posts, *(ii)* to publish in a fewer subreddits, *(iii)* to have a lower number of co-posting authors, *(iv)* to be more interconnected, active and "teamed" than SFW authors.
2. The maximum number of negative posts published by a single NSFW author is much higher than the corresponding one of a single SFW author.
3. Differently from what happens to SFW authors, there is no degree assortativity and no eigenvector assortativity among NSFW authors.

SF (Finding on NSFW subreddits)

1. NSFW subreddits receive much fewer comments than SFW subreddits.

Now, we examine the previous findings in order to identify their correlations. This allows us to have a general view of the phenomenon of NSFW posts in Reddit.

The finding PF.1 tells us that an NSFW post is published in a limited number of subreddits. The finding AF.1 states that NSFW authors publish more than SFW ones. Now, since NSFW posts are fewer than SFW ones, we can conclude that NSFW posts have a much more limited number of authors. In addition, the combination of PF.1 and AF.1 is also a justification to the claim that NSFW authors publish in fewer subreddits than SFW authors.

Combining the findings PF.1 and AF.1 we can conclude that the phenomenon of NSFW posts is a niche one.

The finding PF.1 also tells us that the NSFW posts are little appreciated; actually, this information was quite expected. The results expressed by the finding PF.1 are reinforced by the finding AF.2, which tells us that the maximum number of negative posts published by a single NSFW author is greater than the corresponding number of an SFW author. The finding AF.2 is also, in part, a direct consequence of the finding AF.1.

The finding SF.1, stating that the NSFW subreddits receive fewer comments than SFW ones, represents a further confirmation of what the findings AF.1 and PF.1 say about the fact that NSFW posts are a niche phenomenon.

The poor consideration for NSFW posts, expressed by the finding PF.1, is further confirmed by the finding PF.2, which tells us that not only NSFW posts, but even comments to these posts, receive a much lower score than the comments to SFW posts.

The finding AF.1 (which tells us that the number of co-posting NSFW authors is fewer than SFW authors and that NSFW authors are more interconnected, active and "teamed" than SFW ones) represents a further confirmation that the NSFW post phenomenon is a niche one, carried out by few authors. However, it also tells us that these authors are very active and very well interconnected, ready to play "teamwork".

The last finding extracted, i.e., the finding AF.3, specifies that there is no degree or eigenvector assortativity for NSFW authors. In other words, the strong connection existing among NSFW authors is so widespread and compact that it does not let authors group into "narrow circles". In fact, the sense of cooperation between these authors is so high that the most active ones still collaborate with everyone else and do not limit their interactions to only those with their direct peers, as often happens in many other contexts.

5.2 Content investigation

5.2.1 Methods

Dataset Description. We downloaded the dataset for our analyses from the website `pushshift.io` [65] that represents one of the main data repositories for Reddit. In particular, we focused on 449 NSFW adult subreddits listed at the address `https://www.reddit.com/r/ListOfSubreddits/wiki/nsfw`. We extracted all the posts, along with the corresponding comments, published on these subreddits from January 1^{st}, 2020 to March 31^{st}, 2020. The number of posts composing our dataset is 3,064,758 while the overall number of comments is 11,627,372.

We performed a preliminary ETL (Extraction, Transformation and Loading) activity them. It aims at cleaning available data in such a way as to remove errors and inconsistencies, to integrate data coming from different sources, to transform cleaned and integrated data into a single format and to load transformed data into a unique data source [179].

During the ETL activity, we observed that some of the available posts were published by authors who had left Reddit. We decided to remove these posts and the

corresponding comments from our dataset. Furthermore, a little number of the posts contained in the 449 subreddits of interest were not NSFW ones. Therefore, we had to remove them. To perform such a task, we started from the observation that an NSFW post must be marked as such by its author. If this happens, Reddit puts a red label when displaying it. Moreover, in its database, it sets to `true` the value of the `over_18` field related to this post. As a consequence, we decided to use this value to detect, and then discard from the dataset, the not NSFW posts and the corresponding comments. After the ETL activity, the final number of available NSFW adult posts is 2,981,601, equivalent to 97% of the initial ones. Instead, the final number of NSFW adult comments is 8,383,499, equivalent to 72.10% of the initial ones.

Table 8.1 reports some of the main properties of the authors of posts and comments in our dataset. It highlights some interesting information. In fact, it shows that the number of authors writing comments is much higher (i.e., more than three times) than the number of authors publishing posts. Moreover, only about half of the authors who publish posts also publish comments.

Parameter	*January 2020*	*February 2020*	*March 2020*	*Total*
Authors publishing posts	91,894	92,530	110,873	218,433
Authors publishing comments	369,014	351,967	392,871	738,216
Authors publishing both posts and comments	46,427	44,733	53,063	115,686

Table 5.12: Some parameters regarding authors in the dataset

Figure 8.1 shows the distributions of posts against subreddits (at left) and comments against posts (at right). Figure 8.2 reports the distribution of scores against posts (at left) and comments (at right). As can be seen from these figures, the distribution of posts against subreddits is a Zipf one, whereas all the other distributions are power law. In Table 8.2, we report the values of the coefficients α and δ, along with the minimum and maximum values, relative to these distributions.

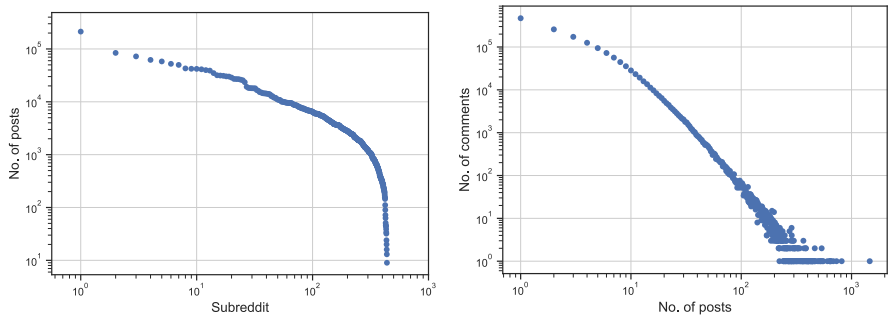


Fig. 5.15: Distributions of posts against subreddits (at left, log-log scale) and comments against posts (at right, log-log scale)

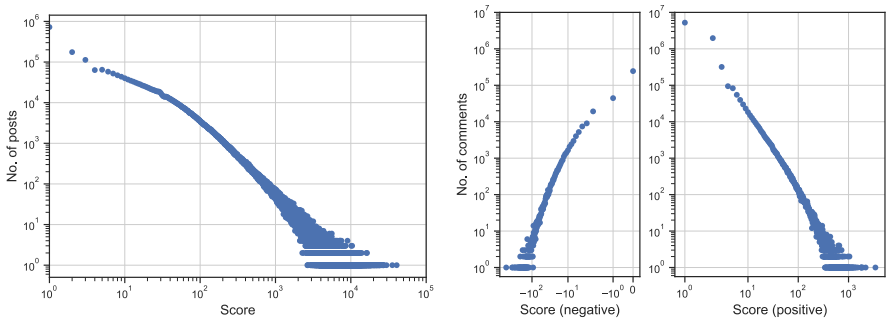


Fig. 5.16: Distributions of scores against posts (at left, log-log scale) and comments (at right, log-log scale)

Field	α	δ	*Minimum Value*	*Maximum Value*
Posts against subreddits	2.1551	0.0487	9	290,746
Comments against posts	3.0821	0.0159	1	1,462
Scores against posts	2.8438	0.0212	0	40,612
Scores against comments (left*)	3.8485	0.0255	-521	0
Scores against comments (right)	2.1456	0.0158	1	3,425

Table 5.13: Values of α and δ, minimum and maximum values of the distributions of interests for the dataset - *The values of α and δ for the left part of the distribution of scores against comments were computed considering the absolute values of scores

General overview of our approach. The general workflow of our approach is illustrated in Figure 5.17. It shows that our approach consists of three main phases, namely: *(i)* Data Cleaning and Annotation, *(ii)* Pattern Extraction and Enrichment, and *(iii)* Network-based Pattern Analysis.

During the Data Cleaning and Annotation phase, we remove irrelevant content and standardize text representation. We also annotate NSFW posts and comments with some additional properties. In particular, we perform lexical (e.g., part-of-speech and named entities) and sentiment annotations. The latter highlight the polarity of sentiments expressed in the texts.

During the Pattern Extraction and Enrichment phase, we extract a set of patterns from posts and comments. These form the basis for the next analysis of NSFW adult posts of Reddit and the corresponding users. In our context, a pattern is a set of words present in posts and comments that satisfy certain properties. During this phase, we first extract frequent patterns. Then, we associate each pattern with a rich set of features concerning the posts and comments from which it derives, as well as the users publishing the texts it represents. Next, for each pattern, we compute some utility measures defined by ourselves. Finally, we select only those patterns

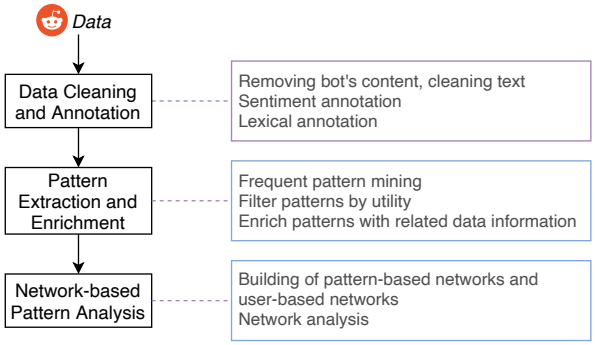


Fig. 5.17: The general workflow of our approach

with high frequency and high utility. Since different utility concepts and measures can be defined, different sets of frequent and useful patterns can be selected. Such sets allow us to analyze the underlying patterns from very different perspectives, but with a uniform methodology.

During the Network-based Pattern Analysis phase, we apply the concepts and approaches of Social Network Analysis on the patterns extracted and enriched during the previous phase. The ultimate goal is the extraction of information and knowledge related to them. In particular, we build and use the following social networks:

- *User Interaction Network*: In it, nodes represent users who published at least one post or comment. An arc (n_i, n_j, w_{ij}) indicates that the user corresponding to n_i commented a post published by the user corresponding to n_j; w_{ij} specifies how many times this happened.
- *Pattern Network*: In it, nodes denote patterns extracted during the previous phase. An arc (n_i, n_j, w_{ij}) indicates that the patterns corresponding to n_i and n_j have been adopted by at least one user in common; w_{ij} represents the number of users who adopted both the pattern corresponding to n_i and the one corresponding to n_j.
- *User Content Network*: In it, nodes represent users who published at least one post or comment. An arc (n_i, n_j, w_{ij}) indicates that there is at least one comment published by the user corresponding to n_i and at least one comment published by the user corresponding to n_j that contain the same pattern; w_{ij} denotes the number of times this happened.

Once these networks are built, we proceed with the application of Social Network Analysis approaches and tools on them to extract information and knowledge on Reddit users, who publish, comment and read NSFW posts and on the corresponding content exchanged among them.

In the next sections we will look at each of these phases in detail.

Data Cleaning and Annotation. This phase is devoted to cleaning up the data of the dataset and annotating it with additional information.

The first step of this phase is the removal of bot-generated content. To identify bots we use a hand-crafted list of bots found in BotWatch[5]. This is a crowdsourced resource available on Reddit to support the investigation of Reddit bots.

The second step is the cleaning of the textual content present in posts and comments. For this purpose, the text of each comment and post is processed through the Natural Language Processing (NLP) pipeline implemented in the Python's spaCy library[6]. During this task, texts are tokenized and lemmatized. Moreover, both English stop words and URLs are removed.

The third step is data annotation. In it, posts and comments are pre-processed to perform a *sentiment annotation* aiming at associating each post and comment with a sentiment value extracted from the text. For this purpose, we use the *compound score* derived by a lexicon and rule-based model for social media text [317]. It is a value in the interval $[-1, 1]$ computed by summing the valence scores of each word in the lexicon, adjusted according to suitable rules, and then normalized to be between -1 (most extreme negative) and +1 (most extreme positive). It is currently recognized as one of the most useful metrics when a single unidimensional sentiment measure is needed for a given sentence. Even more interesting, it has been shown to operate well on Reddit content and social media [308, 342].

In Table 5.14, we report some examples of texts in our dataset. In Table 5.15, we show the results obtained by applying the previous three steps on them. In particular, in the second column of this table, we show the lemmas corresponding to each text, while in the third one we report the corresponding sentiment, computed by applying the compound score.

Id	*Text*
c_1	Serious answer, hell yes! No doubt, so sexy
c_2	Most intense sexy girl posting on reddit, keep them coming.
c_3	Just give me the chance and I would give you a great time.
c_4	Stop being such a h***y b***h and telling me when to f**k you
c_5	You are nothin but a f*****g worthless w***e

Table 5.14: Some input texts from the dataset (swear words are partially masked)

The fourth step regards comment enrichment. During this step, each comment is enriched with features regarding itself, its user and the post it refers to. These features are extremely useful for the Pattern Extraction and Enrichment activities described in Section 5.2.1. To improve readability, we grouped them into two sets

[5] https://www.reddit.com/r/botwatch

[6] https://spacy.io/

Id	*Lemmas*	*Sentiment*
c_1	answer, hell, yes, doubt, sexy	0.4395
c_2	intense, sexy, girl, post, reddit, come	0.6453
c_3	chance, great, time	0.7269
c_4	stop, h***y, b***h, tell, f**k	-0.8591
c_5	nothing, f*****g, worthless, w***e	-0.9128

Table 5.15: Results obtained by applying the Data Cleaning and Annotation tasks on the texts of Table 5.14 (swear words are partially masked)

$\mathcal{F}_U$ and $\mathcal{F}_C$. $\mathcal{F}_U$ contains features related to the user u publishing the comment. They are:

- n_post: it indicates the number of posts published by u;
- n_comm: it denotes the number of comments in the dataset published by u;
- avg_score_post: it represents the average score of the posts published by u;
- avg_score_comm: it indicates the average score of the comments published by u;
- perc_nsfw: it denotes the percentage of NSFW posts published by u;
- perc_sfw: it represents the percentage of SFW posts published by u;
- avg_crosspost: it indicates the average number of crossposts of the posts published by u;
- avg_award: it denotes the average number of awards received by the posts published by u;
- n_sub: it represents the number of subreddits in which u published at least one post;
- perc_top_comm: it indicates the number of the top level comments (i.e., comments on a post and, therefore, not comments on other comments) published by u.

$\mathcal{F}_C$ comprises features related to the comment c itself and the post p it refers to. They are:

- score_post: it indicates the score of p;
- score_comm: it denotes the score of c;
- len_post: it represents the length of the text associated with p;
- len_title: it indicates the length of the title of p;
- len_comm: it denotes the length of the text of c;
- compound: it represents the value of the compound score of c.

We point out that the features of $\mathcal{F}_U$ regarding a certain user u are computed considering all the posts and comments (i.e., not only those concerning NSFW adult content) that u published on Reddit in the time period of the dataset. This choice is

motivated by the fact that we want to have a general characterization of the overall behavior showing by u on Reddit during the period which the dataset refers to.

Pattern extraction and enrichment. This phase is devoted to extracting patterns from the texts of posts and comments in the dataset and, then, enriching them. During this phase an important role is played by pattern mining. This task is well known in the literature. It aims at extracting knowledge from a dataset that can be understood by humans. In particular, it examines the posts and comments returned by the previous phase and extracts interesting and/or unexpected information from them.

Many pattern mining approaches are based on the concept of *pattern frequency*. They aim at identifying the most frequent patterns in a given context. The fundamental assumption, which they are based on, is that frequent patterns are interesting [**?**, 11, 448, 259]. In many application contexts this assumption is true. However, there are cases in which it does not hold. For example, think of the analysis of a purchase transaction database. Here, a pattern such as $\{flour, yeast\}$ might be frequent but uninteresting, since it is fairly obvious that those who buy flour also buy yeast. In light of this consideration, pattern mining researchers have begun to consider that there may be patterns characterized by a low frequency but an extremely high utility (given a certain notion of it). In order to handle such a situation, several *utility functions* to associate with patterns have been introduced [247, 260]. For example, in a sales database, a pattern may have a low co-occurrence frequency but may provide a higher profit than more frequent patterns (think, for instance, of the pattern $\langle$*car*, *car alarm*$\rangle$ against the pattern $\langle$*windshield washer fluid*, *new windshield wipers*$\rangle$).

The introduction of the notion of pattern utility, besides the one of pattern frequency, shifts the focus from frequent pattern mining to high utility pattern mining (hereafter, HUPM) [247, 260, 657]. In this task, the patterns of interest are those characterized by a high utility value, depending on the utility function adopted. We recall that a utility function denotes a user preference ordering over a set of choices [269]. As a consequence, it is a subjective measure. Therefore, it is fair assuming that the utility of an item or a pattern can be defined from several points of view, depending on the preferences of the user exploiting it. This is especially true in our reference scenario where users, posts and comments can be considered from multiple perspectives, even when we focus on a specific issue, such as NSFW adult content. To best address this issue, in this paper, we extend the standard notion of HUPM that considers only one utility function. Specifically, instead of having a one-dimensional view of the utility concept, we have a multi-dimensional view of it. In such a vision, several utility measures can coexist simultaneously and interact with each other, and

the values they assume for an item or a pattern can be suitably combined to obtain an overall value for it.

Having introduced the notions of frequency and utility, we are now able to illustrate our pattern extraction setting and approach. Let $\mathcal{C} = \{c_1, c_2, \cdots, c_n\}$ be a set of lemmatized comments, obtained at the end of the Data Cleaning and Annotation phase, described in Section 5.2.1. Each comment $c_i \in \mathcal{C}$ is associated with a post and is written by an author who is a Reddit user. c_i can be represented by a set of lemmas $c_i = \{l_1, l_2, \cdots, l_m\}$. It can also be seen as a subset of $\mathcal{L} = \{l_1, l_2, \cdots, l_q\}$, i.e., the set of all possible lemmas. Formally, we can write that $c_i \subseteq \mathcal{L}$. Each lemma is an item from the HUPM viewpoint.

A pattern P_j is a set of items; more specifically, $P_j \subseteq \mathcal{L}$. P_j can occur in zero, one or more comments in our dataset. We define the frequency of P_j as the cardinality of the set $\mathcal{C}_j \subseteq \mathcal{C}$ of comments in which it occurs. P_j inherits the set of features characterizing the comments of $\mathcal{C}_j$. Therefore, the utility of P_j can be defined as a suitable function applied on the features of P_j or on a subset of them. The choice of the features and the utility function determines the point of view to be adopted in the pattern analysis. For example, if we focus on the feature compound and the function avg, the utility of a pattern P_j is the average value of the compound scores of the comments relative to the set $\mathcal{C}_j$ in which it appears. It can help us selecting those patterns whose presence in the comments leads to a positive (resp., negative) sentiment. As a more complex example, consider the features score_comm and compound and the function computing the Pearson's correlation [495] between them[7]. In this case, the utility of P_j represents the Pearson's correlation between the compound score and the score of the comments in which P_j appears. It helps us selecting those patterns whose presence in the comments with high (resp., low) score is coupled by a positive (resp., negative) sentiment. This correlation between score and sentiment is not obvious for comments, as there might be comments with high scores and a null or negative sentiment or comments with low score and a null or positive sentiment.

Once the features of interest and the suitable utility functions have been defined, our approach can proceed with the selection of patterns having frequency and utility values higher than a certain threshold. In particular, the frequency threshold may also be low if the goal is to filter out only very rare patterns.

[7] Also known as Pearson's r, the Pearson correlation represents a measure of linear correlation between two sets of data. It measures the ratio of the covariance of two variables to the product of their standard deviations. Its value belongs to the real interval $[-1, 1]$, where 1 (resp., -1) indicates a strong positive (resp., negative) linear correlation, while 0 denotes no correlation.

Our approach for pattern extraction operates as follows. First, it extracts the set of patterns having a frequency higher than a minimum threshold. For this purpose, it uses one of the classical approaches for frequent pattern mining, such as FPGrowth [295]. Then, it associates each extracted pattern with the features appearing in the posts and comments it is present in. These features will be used for the next analyses. After that, it applies the selected utility function to each pattern for computing the pattern's utility value. Finally, it selects and returns only those patterns whose utility value is greater than a minimum threshold.

As an example of how our approach works, suppose we want to consider the features score_comm and compound. Assume we choose as utility function the Pearson's correlation between the two features. Finally, suppose that the minimum frequency threshold is equal to 3. Table 5.16 shows a set of candidate patterns. For each of them, it reports the identifiers of the comments where it appears and, for each comment, the values of the features score_comm and compound, along with the value of the Pearson's correlation between the two features. As can be seen from this table, the three patterns exceed the minimum frequency because the first and the third have a frequency equal to 3 (which means that they are present in 3 comments), while the second has a frequency equal to 4. Suppose we set the minimum utility threshold equal to 0.7, which is a good compromise between the option of selecting many patterns, with the risk of having many useless ones, and that of selecting very few patterns, with the risk of losing useful ones. With this value of the minimum utility threshold, only the pattern $\{great, time\}$ would be selected for the next analysis phase.

Pattern	*Comments*	*Features:* [score_comm, compound]	*Utility:* Pearson
{*answer*, *yes*}	c_{10}	[-2, -0.15]	0.60
	c_{12}	[10, 0.21]	
	c_{16}	[5, 0.54]	
{*great*, *time*}	c_{11}	[13, 0.15]	0.92
	c_{13}	[50, 0.89]	
	c_{21}	[-20, -0.75]	
	c_{22}	[110, 0.99]	
{*intense*, *sexy*}	c_{13}	[50, 0.89]	0.59
	c_{24}	[1, 0.66]	
	c_{25}	[5, -0.32]	

Table 5.16: Example of the pattern extraction phase

If we choose to filter out only very rare patterns, in such a way as not to risk losing little frequent but significant ones, the utility function plays a crucial role. In fact, depending on it, the pattern selection will be directed towards a strategy rather than another. Therefore, in the following of this section, we present an in-depth study of the utility functions of interest.

Pattern utility functions. At the beginning of Section 5.2.1, we saw that it is possible to define several utility functions in order to perform a multi-dimensional analysis on available data. In the following, we consider some of them and conduct several investigations on the patterns extracted using them. In any case, we again point out that, in order to be considered in our analysis, a pattern not only must be extracted by a utility function but also must have a frequency greater than a minimum (possibly very low) threshold. This condition allows us to discard from the analysis rare patterns, whose study could be of little significance[8].

Naive utility function. In order to have a baseline for our analysis, we consider a naive utility function, which assumes that the utility of a pattern P_j is given only by its frequency. This is equivalent to say that each occurrence of a pattern has the same weight regardless of the comment in which it appears. Formally speaking, given a pattern P_j and the set $\mathcal{C}_j \subseteq \mathcal{C}$ of comments in which it occurs, the naive utility function f_n is defined as:

$$f_n(P_j) = |\mathcal{C}_j|$$

The set $\mathcal{P}_n$ of useful patterns consists of those ones whose utility function is greater than a certain threshold th_n:

$$\mathcal{P}_n = \{P_j \mid f_n(P_j) \geq th_n\}$$

After defining the naive utility function, we now analyze its impact on the patterns extracted for the next phase of network-based analysis. In this activity, we refer to the dataset described in Section 5.2.1. As mentioned above, we use a very low value of the frequency threshold to discard only very rare patterns. To this end, we set the frequency threshold equal to 0.01% of the comments available in the dataset.

Figure 5.18 reports the variations in the number of extracted patterns as the utility threshold th_n increases. Patterns are also grouped based on their length. In this figure, we used the semi-log scale because the number of patterns involved is very high. From the analysis of this figure, it is clear that the value 0.07% represents an important "watershed" for the utility threshold. In fact, lower values return a huge number of patterns, difficult to manage in the next network-based analysis phase. Instead, higher values return a limited number of very short patterns. The presence

[8] In this paper, we do not consider the case of anomalies and outliers, i.e., situations where a very rare, but extremely significant, pattern might exist. In fact, outlier and anomaly detection represents a distinct problem, which requires a specific study that we cannot address here.

of only short patterns could be a problem because the most semantically significant information could come from patterns of medium-high length. Therefore, in light of this reasoning, for the naive utility function, we set the threshold th_n equal to 0.07%.

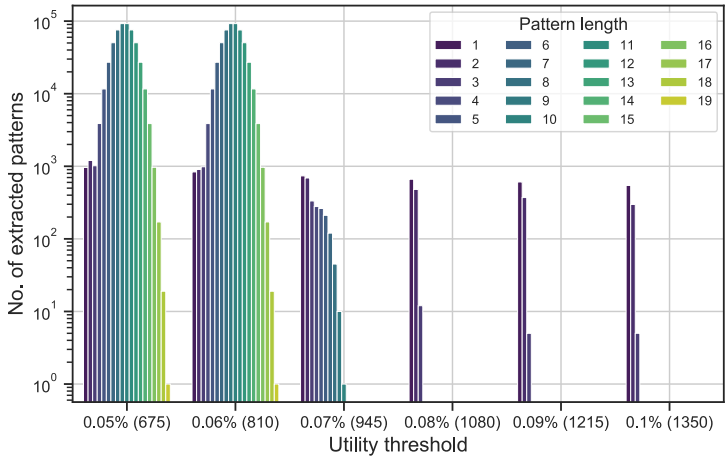


Fig. 5.18: Number of extracted patterns against values of th_n

Compound utility function. As a second utility function, we focus on the compound score (see Section 5.2.1) of the comments where a pattern occurs. In particular, given a pattern P_j and the set $\mathcal{C}_j \subseteq \mathcal{C}$ of comments in which it occurs, and given a comment $c_j \in \mathcal{C}$, we first introduce a function $\gamma(\cdot)$ that receives a comment c_j and returns its compound score. Then, we define the compound utility function f_c as:

$$f_c(P_j) = avg_{c_{j_k} \in \mathcal{C}_j}\{\gamma(c_{j_k})\}$$

where avg computes the average of the set of values received in input.

In this case, it is interesting to analyze both the patterns showing a positive value of f_c and the ones having a negative value of this parameter. As a consequence, we define the following two sets of useful patterns that can be derived using f_c:

$$\mathcal{P}_{f_c^+} = \{P_j \mid f_c(P_j) \geq th_c^+\} \qquad \mathcal{P}_{f_c^-} = \{P_j \mid f_c(P_j) \leq th_c^-\}$$

where th_c^+ and th_c^- are suitable thresholds.

Figure 5.19 (resp., 5.20) reports the variations of the number of patterns extracted as the utility threshold th_c^+ (resp., th_c^-) increases (resp., decreases). Patterns are also grouped with respect to their lengths. The examination of the two figures shows that, unlike the naive utility function, in this case there is no threshold that acts as a watershed. This consideration, together with the fact that the number of patterns extracted is much lower than what happened for the naive utility function, leads

us to choose a low value for th_c^+ and th_c^-. Therefore, based on this reasoning, we set $th_c^+ = 0.2$ and $th_c^- = -0.2$.

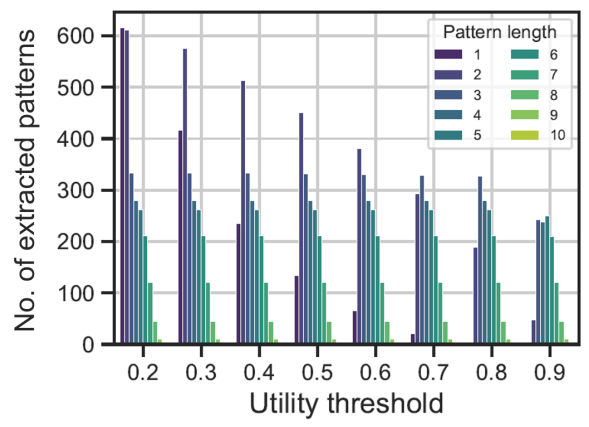


Fig. 5.19: Number of extracted patterns against values of th_c^+

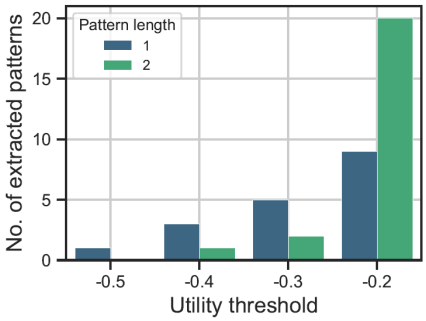


Fig. 5.20: Number of extracted patterns against values of th_c^-

Pearson's correlation utility function. As a final utility function, we consider the Pearson's correlation between the features compound and score_comm (see Section 5.2.1) of the comments in which a pattern P_j occurs. In particular, given a pattern P_j and the set $\mathcal{C}_j \subseteq \mathcal{C}$ of the comments in which it occurs, and given a comment $c_j \in \mathcal{C}_j$, let X (resp., Y) be the set of the values of compound (resp., score_comm) associated with the comments of $\mathcal{C}_j$. The utility function f_p, which computes the Pearson's correlation between the features compound and score_comm of a comment P_j, is defined as:

$$f_p(P_j) = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})^2}}$$

Here, n is the cardinality of $\mathcal{C}_j$ and, therefore, also of X and Y, while x_i (resp., y_i) denotes the i^{th} element of X (resp., Y), and $\overline{x}$ (resp., $\overline{y}$) indicates the mean of the values of X (resp., Y).

Again, it is interesting to analyze both the patterns having a positive value of f_p and those showing a negative value of this parameter. Recall that a positive (resp., negative) value of f_p indicates that there is a direct (resp., inverse) correlation between the sentiment aroused by a comment and the score it obtains. The value of f_p ranges in the real interval $[-1, 1]$; the higher (resp., lower) this value, the greater the direct (resp., inverse) correlation between sentiment and score.

As a consequence, again, we define two sets of useful patterns that can be derived from f_p:

$$\mathcal{P}_{f_p^+} = \{P_j \mid f_p(P_j) \geq th_p^+\} \qquad \mathcal{P}_{f_p^-} = \{P_j \mid f_p(P_j) \leq th_p^-\}$$

where th_p^+ and th_p^- are suitable thresholds.

Figure 5.21 (resp., 5.22) shows the variations of the number of patterns extracted as the utility threshold th_p^+ (resp., th_p^-) increases (resp., decreases). Patterns are also grouped based on their lengths. The examination of the two figures provides us with a not obvious and extremely interesting knowledge pattern. In fact, the number of patterns extracted with a negative value of f_p is much larger (i.e., more than three times) than one with the same positive value of f_p. This allows us to state that a positive (resp., negative) sentiment in a comment is not necessarily accompanied by a high (resp., low) score of it. On the contrary, it appears that a negative sentiment is accompanied by higher scores. This is very evident for moderately positive or negative values of f_p, while this phenomenon is greatly reduced for very negative values of f_p. This can be explained considering that, due to the nature of textual content, NSFW posts and comments tend to be categorized with negative sentiment by any sentiment analysis tool. This happens even when such terms are used in goliardic comments, which are actually appreciated by this type of audience. Think, for example, of a pattern like $\{hot, fuck\}$, possibly accompanied by an emoticon with two little hearts, instead of eyes. In our dataset this pattern reaches a sentiment of -0.1280 but a very high score and, therefore, a negative value of f_p.

This allows us to draw a first important conclusion in this paper, namely that the traditional approaches for sentiment computation do not work well in the case of NSFW posts and comments.

As far as the threshold values are concerned, all the reasoning above, combined with the fact that the number of patterns extracted is very low, leads us to choose low values of th_p^+ and th_p^-. In particular, we set $th_p^+ = 0.1$ and $th_p^- = -0.1$.

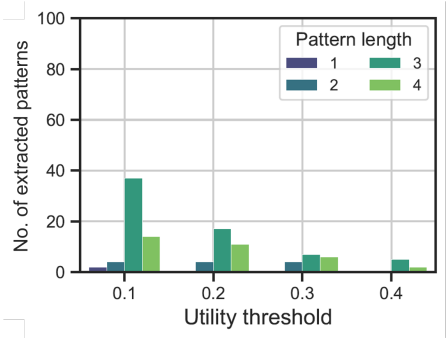


Fig. 5.21: Number of extracted patterns against values of th_p^+

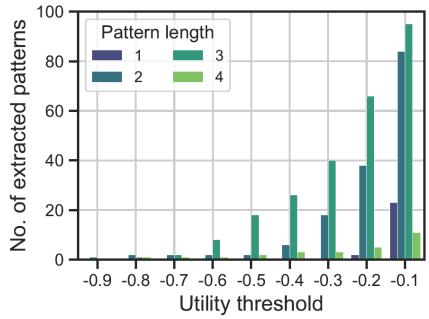


Fig. 5.22: Number of extracted patterns against values of th_p^-

We point out that many other utility functions could be defined over the features presented in Section 5.2.1. In this paper, we decided to focus on these three that we considered particularly significant for the analyses described in the next section.

Network-based Pattern Analysis. In this section, we present the tasks of our approach performed during the Network-based Pattern Analysis phase. They involve the application of Social Network Analysis concepts and techniques to the patterns extracted during the previous phases and to the corresponding users. We recall that, in our approach, different sets of patterns are extracted according to the utility functions adopted. For each set of patterns, three different types of networks are constructed, as we have seen in Section 5.2.1. In the following, we describe the analyses we conducted and the knowledge we obtained by operating on these network types. In particular, we devote a subsection to each of them.

Analysis of User Interaction Networks. In this section, we first formally introduce the concept of User Interaction Network and then extract information on it.

Let $\mathcal{P}_f$ be the set of patterns extracted by applying the utility function f and let $\mathcal{U}_f$ be the set of users who published at least one comment or post containing at least one pattern of $\mathcal{P}_f$. A User Interaction Network $\mathcal{N}^{ui}$ is defined as:

$$\mathcal{N}^{ui} = \langle N^{ui}, A^{ui} \rangle$$

N^{ui} is the set of nodes of $\mathcal{N}^{ui}$. There is a node $n_i \in N^{ui}$ for each user $u_i \in \mathcal{U}_f$. A^{ui} is the set of arcs of $\mathcal{N}^{ui}$. An arc $(n_i, n_j, w_{ij}) \in A^{ui}$ indicates that the user corresponding to n_i commented a post published by the user corresponding to n_j; w_{ij} specifies how many times this fact happened. This network allows us to study the behavior and relationships of users who interact with each other by publishing and commenting NSFW adult content.

Table 5.17 shows the values of some basic parameters for the User Interaction Networks constructed by applying, to the dataset described in Section 5.2.1, the utility functions, as well as the frequency and utility thresholds, described in Section 5.2.1. From the analysis of this table we can see that the information derived in Section 5.2.1 is fully confirmed. For example, the number of nodes and arcs in $\mathcal{N}^{ui}_{f_c^-}$ is much less than the number of nodes and arcs in $\mathcal{N}^{ui}_{f_c^+}$. Conversely, the number of nodes and arcs in $\mathcal{N}^{ui}_{f_p^-}$ is much greater than the number of nodes and arcs in $\mathcal{N}^{ui}_{f_p^+}$. These two trends can be explained considering the cardinalities of $\mathcal{P}_{f_c^-}$, $\mathcal{P}_{f_c^+}$, $\mathcal{P}_{f_p^-}$ and $\mathcal{P}_{f_p^+}$, although the latter refer to patterns while the nodes of User Interaction Networks refer to the corresponding users.

A surprising aspect is the high density of $\mathcal{N}^{ui}_{f_c^-}$ and $\mathcal{N}^{ui}_{f_p^+}$ coupled with the high value of the clustering coefficient for these networks. This leads us to say that, in them, users tend to form very cohesive communities, consisting of many triads and with overall structures very close to those of cliques. This does not happen in the case of $\mathcal{N}^{ui}_{f_p^-}$, where the simultaneous presence of a high density and a low clustering coefficient leads us to think that there are very strong power users, i.e., users receiving comments from many other ones, who do not communicate with each other. For all the networks, the maximum connected components comprise a high fraction of nodes, i.e., about 60% of the overall number of nodes. The only exception regards $\mathcal{N}^{ui}_{f_c^-}$, where the maximum connected component comprises 46.78% of the nodes.

Parameter	$\mathcal{N}^{ui}_{f_n}$	$\mathcal{N}^{ui}_{f_c^-}$	$\mathcal{N}^{ui}_{f_c^+}$	$\mathcal{N}^{ui}_{f_p^-}$	$\mathcal{N}^{ui}_{f_p^+}$
Nodes	272,062	27,083	258,759	27,160	1,452
Arcs	515,471	39,407	496,197	60,662	7,925
Density	0.694e-05	5.373e-05	0.741e-05	8.224e-05	376.15e-05
Clustering coefficient	0.009	0.069	0.002	0.004	0.129
Number of connected components	93,307	14,193	88,697	10,939	506
Size of the maximum connected component	176,140	12,670	167,634	16,030	891
Average weight of arcs	1.205	1.315	1.093	1.205	1.935

Table 5.17: Values of some basic parameters for User Interaction Networks

The average weight of the arcs is very low for all the networks. This tells us that the average number of times a user commented the content of another one is

only slightly more than 1. This leads us to think that the distribution of arcs against weights could follow a power law. The visualization of this distribution in log-log scale, reported in Figure 5.23, allows us to conclude that this conjecture was right. An analogous conclusion was drawn for all the other four User Interaction Networks. We do not report the corresponding graphs for space constraints. We also computed the parameters α and δ of these power law distributions. They are shown in Table 5.18.

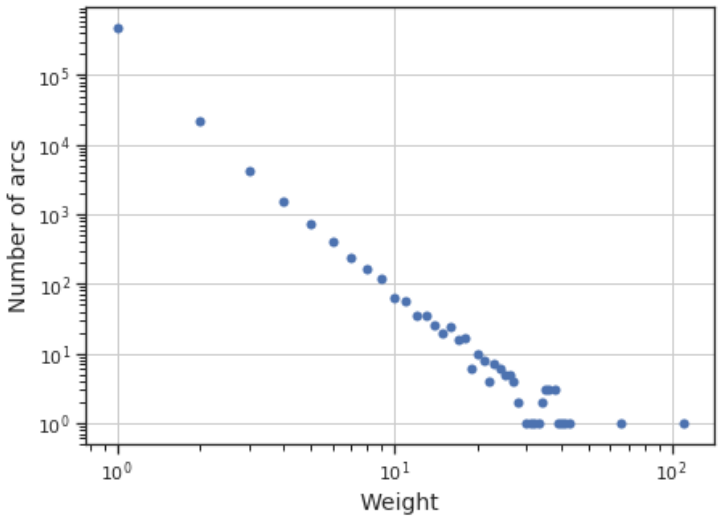


Fig. 5.23: Distribution of arcs against weights for $\mathcal{N}_{f_n}^{ui}$

Parameter	$\mathcal{N}_{f_n}^{ui}$	$\mathcal{N}_{f_c^-}^{ui}$	$\mathcal{N}_{f_c^+}^{ui}$	$\mathcal{N}_{f_p^-}^{ui}$	$\mathcal{N}_{f_p^+}^{ui}$
α	1.368	1.419	1.369	1.371	1.507
δ	0.046	0.056	0.042	0.062	0.063

Table 5.18: Values of the parameters α and δ for the power law distributions of arcs against weights in User Interaction Networks

The previous analysis suggests that there is a very small number of pairs of users such that one of them wrote at least two comments to a post published by the other (we call them *interacting users* in the following). This is a minimum condition for us to talk about a non-casual relationship between the two. We found interesting to compute the average indegree, the average outdegree and the average clustering coefficient for interacting users and to compare them with the corresponding ones concerning all users. In Table 5.19, we report this comparison for the various networks. From the analysis of this table we can see that, independently of the network (and, therefore, of the way in which text patterns are selected), the interacting users

present much greater indegrees and outdegrees than the other ones. Therefore, they are power users in the corresponding networks. Moreover, their clustering coefficient is high. This indicates that they are able to build communities around them. Integrating these two properties, we can say that they are community leaders in the distribution of NSFW adult content in Reddit.

Parameter	$\mathcal{N}_{f_n}^{ui}$	$\mathcal{N}_{f_c^-}^{ui}$	$\mathcal{N}_{f_c^+}^{ui}$	$\mathcal{N}_{f_p^-}^{ui}$	$\mathcal{N}_{f_p^+}^{ui}$
Average Indegree (weight ≥ 2)	19.687	14.836	19.59	16.634	14.393
Average Outdegree (weight ≥ 2)	11.014	5.363	10.835	6.03	7.473
Average Clustering coefficient (weight ≥ 2)	0.031	0.065	0.0180	0.011	0.139
Average Indegree (All)	2.721	1.705	2.695	1.921	1.973
Average Outdegree (All)	2.701	1.695	2.715	1.912	1.987
Average Clustering coefficient (All)	0.003	0.069	0.003	0.003	0.039

Table 5.19: Comparison between interacting users and the overall set of users in the User Interaction Networks

At this point, we thought interesting to investigate whether there was a reciprocal relationship between interacting users. In other words, we wanted to see for how many pairs (u_i, u_j) of interacting users it happened that u_i comments posts of u_j, and vice versa. The fraction of users for whom this happened for each of the networks is reported in Table 5.20. This table shows that it is low for $\mathcal{N}_{f_n}^{ui}$, $\mathcal{N}_{f_c^-}^{ui}$, $\mathcal{N}_{f_c^+}^{ui}$ and $\mathcal{N}_{f_p^-}^{ui}$, while it is higher for $\mathcal{N}_{f_p^+}^{ui}$. As for this last network, we observe that, although its number of nodes is much smaller than the number of nodes of $\mathcal{N}_{f_p^-}^{ui}$, the average indegree and the average outdegree of the nodes of the two networks (for both normal users and interacting ones) are comparable. Furthermore, the clustering coefficient and the fraction of interacting users are much higher for $\mathcal{N}_{f_p^+}^{ui}$ than for $\mathcal{N}_{f_p^-}^{ui}$. All these results tell us that, although the users of $\mathcal{N}_{f_p^-}^{ui}$ are many more than the ones of $\mathcal{N}_{f_p^+}^{ui}$, the latter are certainly more interactive and have a much greater ability to be opinion leaders than the former. Based on the utility function associated with $\mathcal{N}_{f_p^+}^{ui}$, we can say that these users are the only ones capable of maintaining a positive correlation between the compound of their comments and the corresponding scores.

Parameter	$\mathcal{N}_{f_n}^{ui}$	$\mathcal{N}_{f_c^-}^{ui}$	$\mathcal{N}_{f_c^+}^{ui}$	$\mathcal{N}_{f_p^-}^{ui}$	$\mathcal{N}_{f_p^+}^{ui}$
Fraction of proactive users	0.227	0.122	0.227	0.141	0.433

Table 5.20: Fraction of interacting users who comment on each other's posts

We can assume that the users represented in Table 5.20 are the most active ones, able both to publish posts inspiring comments by others (and, therefore, attracting their interest) and, in turn, to comment the posts of others. This last feature is important because it makes them not only content sources but also active entities, who

read and comment the content of others, acting as real opinion leaders. In the following, we call *proactive* such users.

As a last analysis, we have seen how many proactive users belong simultaneously to more than one network. In our opinion, this analysis is important because it identifies those users that, independently of the utility function considered, are always selected as proactive. The results obtained are reported in Table 5.21. From the analysis of this table we can see that the number of such users is very low. This means that the opinion leaders in the field of NSFW adult content are very few, several orders of magnitude smaller than the overall number of users. This implies that, by acting on them (which requires a not exaggerated effort considering their low number), it is possible to reach and influence a really huge number of users.

Parameter	*Value*
Users Proactive in $\mathcal{N}_{f_n}^{ui}$, $\mathcal{N}_{f_c^-}^{ui}$ and $\mathcal{N}_{f_p^-}^{ui}$	139
Users Proactive in $\mathcal{N}_{f_n}^{ui}$, $\mathcal{N}_{f_c^-}^{ui}$ and $\mathcal{N}_{f_p^+}^{ui}$	64
Users Proactive in $\mathcal{N}_{f_n}^{ui}$, $\mathcal{N}_{f_c^+}^{ui}$ and $\mathcal{N}_{f_p^-}^{ui}$	385
Users Proactive in $\mathcal{N}_{f_n}^{ui}$, $\mathcal{N}_{f_c^+}^{ui}$ and $\mathcal{N}_{f_p^+}^{ui}$	59
Users Proactive in $\mathcal{N}_{f_n}^{ui}$, $\mathcal{N}_{f_c^-}^{ui}$, $\mathcal{N}_{f_c^+}^{ui}$ and $\mathcal{N}_{f_p^-}^{ui}$	139
Users Proactive in $\mathcal{N}_{f_n}^{ui}$, $\mathcal{N}_{f_c^-}^{ui}$, $\mathcal{N}_{f_c^+}^{ui}$ and $\mathcal{N}_{f_p^+}^{ui}$	57
Users Proactive in $\mathcal{N}_{f_n}^{ui}$, $\mathcal{N}_{f_c^-}^{ui}$, $\mathcal{N}_{f_p^-}^{ui}$ and $\mathcal{N}_{f_p^+}^{ui}$	12
Users Proactive in $\mathcal{N}_{f_n}^{ui}$, $\mathcal{N}_{f_c^+}^{ui}$, $\mathcal{N}_{f_p^-}^{ui}$ and $\mathcal{N}_{f_p^+}^{ui}$	12
Users Proactive in $\mathcal{N}_{f_n}^{ui}$, $\mathcal{N}_{f_c^-}^{ui}$, $\mathcal{N}_{f_c^+}^{ui}$, $\mathcal{N}_{f_p^-}^{ui}$ and $\mathcal{N}_{f_p^+}^{ui}$	12

Table 5.21: Number of proactive users belonging to more networks

Analysis of Pattern Networks. In this section, we formally introduce the concept of Pattern Network and, then, exploit it to extract information of interest from our dataset.

Let $\mathcal{P}_f$ be the set of patterns extracted by applying the utility function f and let $\mathcal{U}_f$ be the set of users who published at least one comment or post containing at least one pattern of $\mathcal{P}_f$. A Pattern Network $\mathcal{N}^p$ is defined as:

$$\mathcal{N}^p = \langle N^p, A^p \rangle$$

N^p is the set of nodes of $\mathcal{N}^p$. There is a node $n_i \in N^p$ for each pattern of $\mathcal{P}_f$. A^p is the set of arcs of $\mathcal{N}^p$. An arc $(n_i, n_j, w_{ij}) \in A^p$ denotes that the patterns corresponding to n_i and n_j were adopted by at least one user of $\mathcal{U}_f$ in common; w_{ij} specifies how many users adopted it.

This network allows us to study correlations between different NSFW patterns. For example, through it, we can investigate if there exist NSFW patterns that often

appear together, and if this happens indiscriminately or for certain categories of users.

In Table 5.22, we show the values of some basic parameters for the Pattern Networks constructed by applying, to the dataset described in Section 5.2.1, the utility functions and the frequency and utility thresholds described in Section 5.2.1. Similarly to what we observed for Table 5.17 introduced in the previous section, this table represents a confirmation of the knowledge extracted in Section 5.2.1.

Parameter	$\mathcal{N}^p_{f_n}$	$\mathcal{N}^p_{f_c^-}$	$\mathcal{N}^p_{f_c^+}$	$\mathcal{N}^p_{f_p^-}$	$\mathcal{N}^p_{f_p^+}$
Nodes	2,688	29	2,487	213	57
Arcs	771,239	346	595,777	17,749	1,486
Density	0.214	0.852	0.193	0.786	0.931
Clustering coefficient	0.524	0.907	0.504	0.917	0.964
Number of connected components	949	1	948	1	2
Size of the maximum connected component	1,740	29	1,540	213	56
Average weight of arcs	7.159	30.763	7.614	5.297	6.284

Table 5.22: Values of some basic parameters for Pattern Networks

As a first in-depth analysis of this network, we consider the distribution of its arcs against weights. In Figure 5.24, we report this distribution in log-log scale for $\mathcal{N}^p_{f_n}$. From the analysis of this figure we observe that it follows a power law. An analogous result was obtained for all the other Pattern Networks.

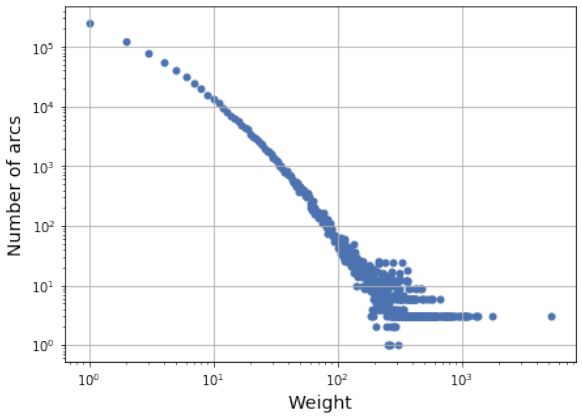


Fig. 5.24: Distribution of arcs against weights for $\mathcal{N}^p_{f_n}$

We also computed the parameters α and δ of these power law distributions. They are shown in Table 5.23. As in the case of the User Interaction Networks, also for the Pattern Networks the most interesting arcs are those with the highest weights. In this case, they indicate patterns employed very often jointly by users. For this reason, we considered the arcs whose weight is higher than or equal to a certain threshold X, which represents the minimum number of times two patterns must coexist to be

selected. In this analysis, we considered several values of X. Some of them are low and more suitable for the networks with a small number of nodes, i.e., $\mathcal{N}^p_{f_c^-}$, $\mathcal{N}^p_{f_p^-}$ and $\mathcal{N}^p_{f_p^+}$. Others are high and well suited for the largest networks and, thus, for $\mathcal{N}^p_{f_n}$ and $\mathcal{N}^p_{f_c^+}$. In Table 5.24, we report the cardinality of the sets of patterns involved in these arcs for each Pattern Network.

Parameter	$\mathcal{N}^p_{f_n}$	$\mathcal{N}^p_{f_c^-}$	$\mathcal{N}^p_{f_c^+}$	$\mathcal{N}^p_{f_p^-}$	$\mathcal{N}^p_{f_p^+}$
α	1.453	2.366	1.430	1.678	1.468
δ	0.045	0.105	0.047	0.088	0.129

Table 5.23: Values of some basic parameters for Pattern Networks

The analysis of this table shows that:

- As for the largest networks, i.e., $\mathcal{N}^p_{f_n}$ and $\mathcal{N}^p_{f_c^+}$, the decrease of the number of coexisting patterns is very gradual against the increase of X.
- As for $\mathcal{N}^p_{f_c^-}$, most of the patterns coexist even with high values of X (e.g., $X = 20$ and $X = 50$). Interestingly, about half of the available patterns continue to coexist even for very high values of X (e.g., $X = 100$).
- As for $\mathcal{N}^p_{f_p^-}$, the decrease of the number of coexisting patterns is rapid when $6 < X < 14$. Then it slows down. Interestingly, there are some patterns that coexist even with high values of X (i.e., $X = 50$ or $X = 100$), despite the low starting number of patterns.
- As for $\mathcal{N}^p_{f_p^+}$, already for $X = 10$ most of the patterns no longer coexist. Moreover, the number of coexisting patterns becomes 0 already for $X = 20$.

The previous reasoning leads us to conclude that, in the network $\mathcal{N}^p_{f_c^-}$, the same patterns tend to occur repeatedly. Such trend is also observed for the network $\mathcal{N}^p_{f_p^-}$, although in a smaller measure. It becomes even less evident for the networks $\mathcal{N}^p_{f_n}$ and $\mathcal{N}^p_{f_c^+}$, where the repetitiveness and the variety of patterns are balanced. Finally, the repetitiveness of patterns is not observed for the network $\mathcal{N}^p_{f_p^+}$, where pattern variety prevails over pattern repetitiveness.

Parameter	$\mathcal{N}^p_{f_n}$	$\mathcal{N}^p_{f_c^-}$	$\mathcal{N}^p_{f_c^+}$	$\mathcal{N}^p_{f_p^-}$	$\mathcal{N}^p_{f_p^+}$
$X = 2$	1,636	29	1,435	213	55
$X = 6$	1,457	28	1,262	201	55
$X = 10$	1,396	28	1,201	143	19
$X = 14$	1,359	28	1,165	87	5
$X = 20$	1,324	26	1,132	55	0
$X = 50$	1,145	20	982	19	0
$X = 100$	855	14	744	9	0

Table 5.24: Cardinality of the sets of patterns exploited very often jointly by users

The patterns belonging to the sets defined above represent the ones appearing most frequently together. We considered interesting to identify triads and, more generally, cliques consisting of three or more of these patterns. In fact, these cliques denote sets of three or more patterns that tend to be always together in posts and comments related to NSFW adult content. We set $X = 6$ for the smallest networks (i.e., $\mathcal{N}^p_{f_c^-}$, $\mathcal{N}^p_{f_p^-}$ and $\mathcal{N}^p_{f_p^+}$) and $X = 20$ for the largest ones (i.e., $\mathcal{N}^p_{f_n}$ and $\mathcal{N}^p_{f_c^+}$). The different values of X for small and large networks were motivated by the fact that the minimum frequency with which patterns must be jointly used, in order to be considered coexisting, must take into account the size of the network, and therefore the difficulty of finding such a property. This difficulty is greater in small-medium networks and smaller in large ones. The choice of a lower value of X for the smallest networks and a higher value for the largest ones derives from this reasoning.

Figure 5.25 shows the distribution of the cliques thus obtained. The results reported in it substantially confirm the information we had derived by analyzing Table 5.24. We call *coexisting* patterns those ones belonging to one of these cliques. They represent patterns tending to appear together more than the others in NSFW posts and comments. In the graph of Figure 5.25, we have shown the cliques associated with each Pattern Network. In that figure, we reported the maximum cliques. However, it is clear that the presence of a clique of dimension q involves the presence of more cliques of dimensions $q-1, q-2, \ldots, 3$. This property is important to be considered when it is necessary to make intersections among cliques of networks with very different sizes, like those examined in this section.

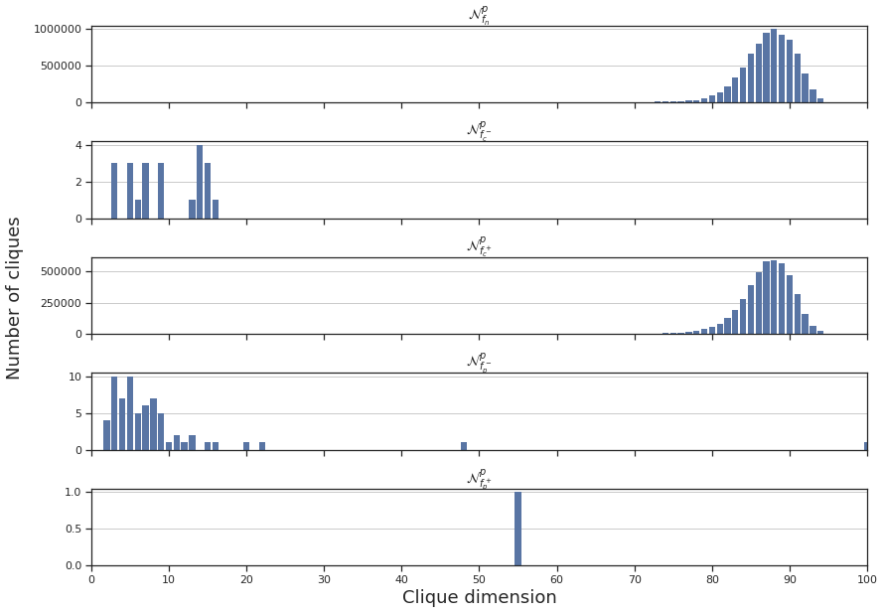


Fig. 5.25: Distribution of cliques of *coexisting* patterns in our Pattern Networks

As a last analysis, we verified if there are patterns coexisting in all our Pattern Networks, regardless the utility functions adopted to construct them. The objective is looking for those patterns that not only tend to appear together in the NSFW posts and comments, but also behave in this way regardless of the utility function used to extract them. Clearly, the request we are making is very strong. In Table 5.25, we report the number of patterns that satisfy this requirement for each possible combination of Pattern Networks.

Parameter	*Value*
Number of *coexisting* patterns simultaneously belonging to $\mathcal{N}^p_{f_n}$, $\mathcal{N}^p_{f_c^-}$ and $\mathcal{N}^p_{f_p^-}$	0
Number of *coexisting* patterns simultaneously belonging to $\mathcal{N}^p_{f_n}$, $\mathcal{N}^p_{f_c^-}$ and $\mathcal{N}^p_{f_p^+}$	0
Number of *coexisting* patterns simultaneously belonging to $\mathcal{N}^p_{f_n}$, $\mathcal{N}^p_{f_c^+}$ and $\mathcal{N}^p_{f_p^-}$	61
Number of *coexisting* patterns simultaneously belonging to $\mathcal{N}^p_{f_n}$, $\mathcal{N}^p_{f_c^+}$ and $\mathcal{N}^p_{f_p^+}$	1

Table 5.25: Number of *coexisting* patterns simultaneously belonging to more Pattern Networks

From the analysis of this table we can see that only in one case there are coexisting patterns that simultaneously belong to three Pattern Networks (in particular, to $\mathcal{N}^p_{f_n}$, $\mathcal{N}^p_{f_c^+}$ and $\mathcal{N}^p_{f_p^-}$). These patterns can be seen as the main "building blocks" of posts and comments with NSFW contents occurring in Reddit. To give an idea of what these patterns look like, in Table 5.26, we report some of them.

Patterns
{*fuck*}
{*hot*}
{*ass, beautiful*}
{*fantastic, look*}
{*tit, holy*}
{*profile, tip*}
{*absolutely, amazing, body*}
{*meet, share, thank*}
{*face, like, need*}
{*hot, look, super*}
{*sexy, pussy, tight*}
{*ask, body, face, post*}
{*face, hi, need, pic*}
{*ass, nice, fuck, hard, cock*}
{*body, amazing, look, love, nude*}

Table 5.26: Examples of *coexisting* patterns simultaneously belonging to $\mathcal{N}^p_{f_n}$, $\mathcal{N}^p_{f_c^+}$ and $\mathcal{N}^p_{f_p^-}$

Analysis of User Content Networks. In this section, we first formalize the concept of User Content Network and then extract knowledge from the User Content Networks obtained from our dataset.

Let $\mathcal{P}_f$ be the set of patterns extracted by applying the utility function f and let $\mathcal{U}_f$ be the set of users who published at least one comment or post containing at least one pattern of $\mathcal{P}_f$. A User Content Network $\mathcal{N}^{uc}$ is defined as:

$$\mathcal{N}^{uc} = \langle N^{uc}, A^{uc} \rangle$$

N^{uc} is the set of nodes of $\mathcal{N}^{uc}$. There is a node $n_i \in N^{uc}$ for each user $u_i \in \mathcal{U}_f$. A^{uc} is the set of arcs of $\mathcal{N}^{uc}$. An arc $(n_i, n_j, w_{ij}) \in A^{uc}$ indicates that at least one post or comment published by n_i and at least one post or comment published by n_j contain the same pattern; w_{ij} denotes the number of times this event occurred.

If the User Interaction Network introduced in Section 5.2.1 allows us to study currently existing user communities, the User Content Network allows us to go one step further. In fact, it allows the identification of virtual communities of users exploiting the same patterns and, de facto, similar languages and contents. These communities may already exist in the User Interaction Network, in which case they are also real communities. Alternatively, they could involve users who never interacted with each other, in which case they are virtual. In this last scenario, our approach could represent the engine of a recommender system aimed at building new real communities of users with similar languages and interests.

In Table 5.27, we show the values of some basic parameters for the User Content Networks constructed by applying to our dataset the utility functions, as well as the frequency and utility thresholds, described in Section 5.2.1. This table presents the same trends as Tables 5.17 and 5.22 and, therefore, also represents a confirmation of the knowledge extracted in Section 5.2.1.

Parameter	$\mathcal{N}^{uc}_{f_n}$	$\mathcal{N}^{uc}_{f_c^-}$	$\mathcal{N}^{uc}_{f_c^+}$	$\mathcal{N}^{uc}_{f_p^-}$	$\mathcal{N}^{uc}_{f_p^+}$
Nodes	272,062	27,083	258,759	27,160	1,452
Arcs	51,579,252	8,743,774	48,546,980	4,609,563	8,296
Density	0.0139e-2	2.384e-2	0.145e-2	1.249e-2	0.788e-2
Clustering coefficient	0.222	0.494	0.212	0.416	0.211
Number of connected components	195,334	12,745	189,539	15,439	1,130
Size of the maximum connected component	76,728	14,339	69,220	11,722	323
Average weight of arcs	1.121	1.019	1.121	1.016	1.247

Table 5.27: Values of some basic parameters for User Content Networks

Similarly to the User Interaction Networks, also for the User Content Networks the average weight of the arcs is very low. Also for these networks, we computed the distribution of arcs against weights. In Figure 5.26, we report the results obtained for the $\mathcal{N}^{uc}_{f_n}$ network in log-log scale. From the analysis of this figure it is clear that this distribution follows a power law. A similar result was obtained for all the other

User Content Networks. We also computed the parameters α and δ of these power law distributions. They are reported in Table 5.28.

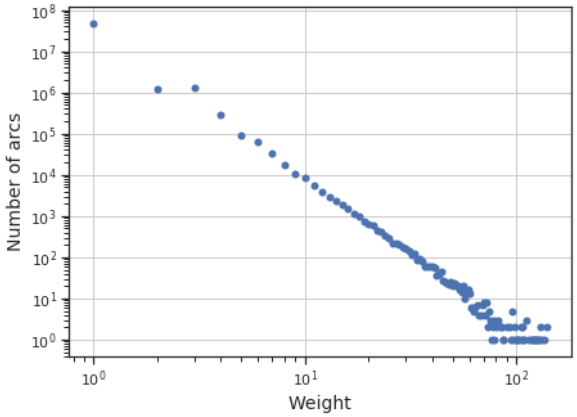


Fig. 5.26: Distribution of arcs against weights for $\mathcal{N}_{f_n}^{uc}$

Parameter	$\mathcal{N}_{f_n}^{uc}$	$\mathcal{N}_{f_c^-}^{uc}$	$\mathcal{N}_{f_c^+}^{uc}$	$\mathcal{N}_{f_p^-}^{uc}$	$\mathcal{N}_{f_p^+}^{uc}$
α	1.300	1.174	1.297	1.289	1.537
δ	0.032	0.146	0.033	0.069	0.098

Table 5.28: Values of the parameters α and δ for the power law distributions of the arcs against the weights in User Content Networks

Analogously to what happened for User Interaction Networks in Section 5.2.1, also for User Content Networks, we can observe that there is a very small number of pairs of users who adopted the same pattern in their comments two or more times (we call them *common content* users in the following). These pairs are the seeds from which we can start to identify virtual communities. We computed the average indegree, outdegree and clustering coefficient for common content users and we compared them with the corresponding ones for all the users involved in the User Content Networks. In Table 5.29 we report the results obtained. From the analysis of this table we can observe that, regardless of the User Content Network we consider, common content users have a much higher degree than the other users and, therefore, are power users. Since their clustering coefficient is also higher than that of the other users, we can conclude that they are able to build communities around them. Similarly to the interacting users in Section 5.2.1, we can conclude that they are community leaders. In particular, they are real community leaders if the corresponding community already exists, or potential community leaders, if this community is currently only virtual.

Parameter	$\mathcal{N}_{fn}^{uc}$	$\mathcal{N}_{fc^-}^{uc}$	$\mathcal{N}_{fc^+}^{uc}$	$\mathcal{N}_{fp^-}^{uc}$	$\mathcal{N}_{fp^+}^{uc}$
Average Degree (weight ≥ 2)	1,988.39	3,412.28	1,961.02	1,663.42	43.210
Average Clustering coefficient (weight ≥ 2)	0.434	0.765	0.396	0.649	0.412
Average Degree (All)	274.82	392.15	270.2	194.10	6.41
Average Clustering coefficient (All)	0.222	0.494	0.212	0.416	0.211

Table 5.29: Comparison between common content users and the overall set of users in the User Content Networks

The User Content Network is intrinsically bidirectional; so, if there is a link from a user u_i to a user u_j, that link is also to be intended in the other sense. As a consequence, using the language adopted in Section 5.2.1 for real communities, we can say that all content users are to be intended as *proactive*.

As a further analysis, analogously to what we did for real communities in Section 5.2.1, we determined the fraction of common content users who simultaneously belong to more than one User Content Network. The results obtained are shown in Table 5.30. From the analysis of this table we can observe that this number of users is very small. Such a result is not surprising because it is in line with the one shown in Table 5.21. It indicates that also for virtual communities, as already seen for real ones, opinion leaders in the field of NSFW content are very few.

Parameter	*Value*
Common content users in $\mathcal{N}_{fn}^{uc}$, $\mathcal{N}_{fc^-}^{uc}$ and $\mathcal{N}_{fp^-}^{uc}$	298
Common content users in $\mathcal{N}_{fn}^{uc}$, $\mathcal{N}_{fc^-}^{uc}$ and $\mathcal{N}_{fp^+}^{uc}$	155
Common content users in $\mathcal{N}_{fn}^{uc}$, $\mathcal{N}_{fc^+}^{uc}$ and $\mathcal{N}_{fp^-}^{uc}$	1,020
Common content users in $\mathcal{N}_{fn}^{uc}$, $\mathcal{N}_{fc^+}^{uc}$ and $\mathcal{N}_{fp^+}^{uc}$	144
Common content users in $\mathcal{N}_{fn}^{uc}$, $\mathcal{N}_{fc^-}^{uc}$, $\mathcal{N}_{fc^+}^{uc}$ and $\mathcal{N}_{fp^-}^{uc}$	298
Common content users in $\mathcal{N}_{fn}^{uc}$, $\mathcal{N}_{fc^-}^{uc}$, $\mathcal{N}_{fc^+}^{uc}$ and $\mathcal{N}_{fp^+}^{uc}$	144
Common content users in $\mathcal{N}_{fn}^{uc}$, $\mathcal{N}_{fc^-}^{uc}$, $\mathcal{N}_{fp^-}^{uc}$ and $\mathcal{N}_{fp^+}^{uc}$	32
Common content users in $\mathcal{N}_{fn}^{uc}$, $\mathcal{N}_{fc^+}^{uc}$, $\mathcal{N}_{fp^-}^{uc}$ and $\mathcal{N}_{fp^+}^{uc}$	29
Common content users in $\mathcal{N}_{fn}^{uc}$, $\mathcal{N}_{fc^-}^{uc}$, $\mathcal{N}_{fc^+}^{uc}$, $\mathcal{N}_{fp^-}^{uc}$ and $\mathcal{N}_{fp^+}^{uc}$	28

Table 5.30: Number of common content proactive users belonging to more networks

Finally, we found it interesting to see how many of the virtual opinion leaders are already real and how many, instead, are potential, and therefore not already known and discovered thanks to our approach. For this purpose, for each row in Table 5.30, we compared the corresponding users (who, recall, are virtual opinion leaders) with those associated with the same row in Table 5.21 (who, instead, are real ones). The results obtained are shown in Table 5.31. In particular, in the third column of this table, we report the fraction of real opinion leaders that are also identified as virtual ones by our approach. Instead, in the fourth column, we report the fraction of virtual opinion leaders who are already real ones. From the analysis

of this table, we can observe that our approach for finding virtual opinion leaders is almost complete, since the fraction of real opinion leaders that are recognized by it is greater than 0.90. Furthermore, we can observe that it is also very useful and significant, because it is able to propose a considerable number of new potential opinion leaders whose existence was not known and who can be used as seeds for building new communities.

Real opinion leaders	*Virtual opinion leaders*	*Fraction of real opinion leaders who are also virtual ones*	*Fraction of virtual opinion leaders who are also real ones*
Users belonging to $\mathcal{N}^{ui}_{f_n}, \mathcal{N}^{ui}_{f_c^-}$ and $\mathcal{N}^{ui}_{f_p^-}$	Users belonging to $\mathcal{N}^{uc}_{f_n}, \mathcal{N}^{uc}_{f_c^-}$ and $\mathcal{N}^{uc}_{f_p^-}$	0.97	0.44
Users belonging to $\mathcal{N}^{ui}_{f_n}, \mathcal{N}^{ui}_{f_c^-}$ and $\mathcal{N}^{ui}_{f_p^+}$	Users belonging to $\mathcal{N}^{uc}_{f_n}, \mathcal{N}^{uc}_{f_c^-}$ and $\mathcal{N}^{uc}_{f_p^+}$	0.94	0.39
Users belonging to $\mathcal{N}^{ui}_{f_n}, \mathcal{N}^{ui}_{f_c^+}$ and $\mathcal{N}^{ui}_{f_p^-}$	Users belonging to $\mathcal{N}^{uc}_{f_n}, \mathcal{N}^{uc}_{f_c^+}$ and $\mathcal{N}^{uc}_{f_p^-}$	0.92	0.35
Users belonging to $\mathcal{N}^{ui}_{f_n}, \mathcal{N}^{ui}_{f_c^+}$ and $\mathcal{N}^{ui}_{f_p^+}$	Users belonging to $\mathcal{N}^{uc}_{f_n}, \mathcal{N}^{uc}_{f_c^+}$ and $\mathcal{N}^{uc}_{f_p^+}$	0.93	0.38
Users belonging to $\mathcal{N}^{ui}_{f_n}, \mathcal{N}^{ui}_{f_c^-}, \mathcal{N}^{ui}_{f_c^+}$ and $\mathcal{N}^{ui}_{f_p^-}$	Users belonging to $\mathcal{N}^{uc}_{f_n}, \mathcal{N}^{uc}_{f_c^-}, \mathcal{N}^{uc}_{f_c^+}$ and $\mathcal{N}^{uc}_{f_p^-}$	0.94	0.44
Users belonging to $\mathcal{N}^{ui}_{f_n}, \mathcal{N}^{ui}_{f_c^-}, \mathcal{N}^{ui}_{f_c^+}$ and $\mathcal{N}^{ui}_{f_p^+}$	Users belonging to $\mathcal{N}^{uc}_{f_n}, \mathcal{N}^{uc}_{f_c^-}, \mathcal{N}^{uc}_{f_c^+}$ and $\mathcal{N}^{uc}_{f_p^+}$	0.92	0.37
Users belonging to $\mathcal{N}^{ui}_{f_n}, \mathcal{N}^{ui}_{f_c^-}, \mathcal{N}^{ui}_{f_p^-}$ and $\mathcal{N}^{ui}_{f_p^+}$	Users belonging to $\mathcal{N}^{uc}_{f_n}, \mathcal{N}^{uc}_{f_c^-}, \mathcal{N}^{uc}_{f_p^-}$ and $\mathcal{N}^{uc}_{f_p^+}$	0.96	0.35
Users belonging to $\mathcal{N}^{ui}_{f_n}, \mathcal{N}^{ui}_{f_c^+}, \mathcal{N}^{ui}_{f_p^-}$ and $\mathcal{N}^{ui}_{f_p^+}$	Users belonging to $\mathcal{N}^{uc}_{f_n}, \mathcal{N}^{uc}_{f_c^+}, \mathcal{N}^{uc}_{f_p^-}$ and $\mathcal{N}^{uc}_{f_p^+}$	0.97	0.39
Users belonging to $\mathcal{N}^{ui}_{f_n}, \mathcal{N}^{ui}_{f_c^-}, \mathcal{N}^{ui}_{f_c^+}, \mathcal{N}^{ui}_{f_p^-}$ and $\mathcal{N}^{ui}_{f_p^+}$	Users belonging to $\mathcal{N}^{uc}_{f_n}, \mathcal{N}^{uc}_{f_c^-}, \mathcal{N}^{uc}_{f_c^+}, \mathcal{N}^{uc}_{f_p^-}$ and $\mathcal{N}^{uc}_{f_p^+}$	0.95	0.41

Table 5.31: Fraction of real opinion leaders who are also virtual, and vice versa

5.2.2 Results

In this section, we present some considerations regarding the proposed approach and the results obtained. We have seen that our approach involves three main phases, namely: *(i)* Data Cleaning and Annotation; *(ii)* Pattern Extraction and Enrichment; *(iii)* Network-based Pattern Analysis. The first phase has the only objective to prepare data for the next two ones. The second phase still deals with the preparation of data but, at the same time, allows the extraction of information of interest by suitably combining the utility functions and interpreting the extracted patterns and their features. In particular, thanks to the analysis of the features of the patterns extracted through the Pearson's correlation utility function, we discovered an important piece of knowledge. Indeed, we observed that a positive (resp., negative) sentiment in a comment is not necessarily accompanied by a high (resp. low) score of that comment. We also saw that the direct consequence of this knowledge is that

traditional approaches to sentiment computation do not work well in the case of NSFW posts and comments.

The third phase focuses on extracting meaningful information from available data using the concepts and methodologies of Social Network Analysis. Specifically, we defined three support social networks. The first allowed us to study real communities of proactive users sharing a common language and potentially the same interests. The third allowed us to go further and determine virtual communities of users sharing language and interests. The second allowed us to shift the focus from users to patterns and identify the ones appearing together most frequently in user comments and posts having NSFW adult content. Those described above are just the general peculiarities of the three networks. Starting from them and from the tools provided by Social Network Analysis, we defined a uniform approach for the extraction of several interesting information. This approach first examines the distribution of the weights of the network arcs. After having ascertained that they follow a power law, it focuses on the arcs with higher weights. Then, it combines these arcs together in order to identify the possible presence of triads or cliques. Such structures are the basis for the detection of communities of users (in the case of the first and third network) or of "core patterns" (in the case of the second network). These communities express commonality of interests, in the case of users, and commonality of language, in the case of patterns.

In the following, we give a brief overview of the main information extracted through our approach.

Applying our approach to the User Interaction Networks we found that the number of interacting users is very low and that they are also power users and community leaders. Most of them are just a source of information, while a small part (ranging from 12% to 43% of users, depending on the network) is proactive, and therefore interacts with other users in both directions. It is not necessarily the case that proactive users in one User Interaction Network are also proactive in the others. This happens very rarely. Those few users who are proactive in more networks are opinion leaders. Their knowledge can become very valuable because acting on them (which requires a not exaggerated effort, since their number is very small), it is possible to reach and influence a really huge number of users.

Applying our approach to the Pattern Networks, we found that the number of patterns adopted very frequently by users (the so-called coexisting patterns) is very small. The number of these patterns and the variation of their number against the minimum frequency threshold vary for the different available networks, although these numbers are always very low. Considering coexisting patterns connected to each other makes it possible to obtain triads and cliques, which represent the seeds

found in most of the NSFW posts and comments in Reddit. Such seeds are very different in number, and often different in content, from one Pattern Network to another. Only an extremely small number of them are common to all Pattern Networks. They represent the building blocks for NSFW adult posts or comments on Reddit. Some of these patterns have been reported in Table 5.26.

Applying our approach to User Content Networks, we had the opportunity to study virtual communities of users. Some of them are already present on Reddit while others are only virtual and can be used as part of a system which suggests new communities of users adopting similar content and languages. We have seen that the number of proactive users is very small in virtual communities, and the number of them who simultaneously belong to more networks is even smaller. Also for virtual communities, as for real ones, such users represent opinion leaders, and acting on them it is possible to reach a huge number of users. Finally, we have seen that our approach for extracting virtual opinion leaders is complete because it is capable of finding more than 90% of real opinion leaders. At the same time, it is able to identify a large number of virtual opinion leaders whose existence was unknown and who could be used as seeds or building blocks for the creation of new communities.

6

Investigating negative reviews and negative influencers

In this chapter, we propose an investigation of negative reviews and define the profile of negative influencers in Yelp. The methodology adopted to achieve this goal consists of two phases. The first one is theoretical and aims at defining a multi-dimensional social network based model of Yelp, three stereotypes of Yelp users, and a network based model to represent negative reviewers and their relationships. The second phase is experimental and consists in the definition of five hypotheses on negative reviews and reviewers in Yelp and their verification through an extensive data analysis campaign. This was performed on Yelp data represented by means of the models introduced during the first phase. Its most important result is the construction of the profile of negative influencers in Yelp. The main novelties of this approach are: (i) the definition of the two social network based models of Yelp and its users; (ii) the definition of three stereotypes of Yelp users and their characteristics; (iii) the construction of the profile of negative influencers in Yelp.

The material presented in this chapter was derived from [222].

6.1 Methods

6.1.1 Definition of Yelp model

Our multi-dimensional investigation of negative reviews and detection of negative influencers in Yelp is possible thanks to a new multi-dimensional social network-based model of Yelp. This model starts from the observation that, in this social medium, businesses are organized according to a taxonomy consisting of four levels. Level 0 includes 22 macro-categories. Each macro-category has one or more child categories; therefore, level 1 includes 1002 categories. A category may have zero, one or more sub-categories; as a consequence, level 2 comprises 532 sub-categories. Finally, level 3, has only 19 sub-sub-categories; indeed, most sub-categories are not further categorized. Our model represents Yelp as a set of 22 communities, one for each macro-category:

$$\mathcal{Y} = \{\mathcal{C}_1, \mathcal{C}_2, \cdots, \mathcal{C}_{22}\}$$

Given the macro-category $\mathcal{C}_i$, $1 \leq i \leq 22$, a corresponding user network $\mathcal{U}_i = \langle N_i, A_i \rangle$ can be defined. N_i is the set of the nodes of $\mathcal{U}_i$; there is a node n_{i_p} for each user u_{i_p} who reviewed at least one business of $\mathcal{C}_i$. A_i is the set of the arcs of $\mathcal{U}_i$; there is an arc $a_{pq} = (n_{i_p}, n_{i_q}) \in A_i$ if there exists a relationship between the users u_{i_p}, corresponding to n_{i_p}, and u_{i_q}, corresponding to n_{i_q}.

Finally, an overall user network $\mathcal{U} = \langle N, A \rangle$ corresponding to $\mathcal{Y}$ can be defined. There is a node $n_i \in N$ for each Yelp user. There is an arc $a_{pq} = (n_p, n_q) \in A$ if there exists a relationship between the users u_p, corresponding to n_p, and u_q, corresponding to n_q.

In the definition of $\mathcal{U}$ (and, consequently, of $\mathcal{U}_i$), we do not specify the kind of relationship between u_p and u_q. Actually, it is possible to define a specialization of $\mathcal{U}$ for each relationship we want to investigate. Here, we are interested in two relationships existing between Yelp users, namely friendship and co-review. As a consequence, we define two specializations of $\mathcal{U}$, namely $\mathcal{U}^f$ and $\mathcal{U}^{cr}$. $\mathcal{U}^f$ is the specialization of $\mathcal{U}$ when we consider friendship as the relationship between users, whereas $\mathcal{U}^{cr}$ denotes the specialization of $\mathcal{U}$ when co-review (i.e., reviewing the same business) is the relationship between users.

Starting from this model, it is possible to define some Yelp stereotypes, namely: *(i) the k-bridge*, i.e., a person operating in k categories of Yelp; *(ii) the power user*, i.e., a person very active in all the categories that she is interested in; *(iii) the double-life user*, i.e., a person showing different behaviors in the different categories she attends. Her different behaviors can regard the activity level (*access-dl-user*) or the severity of her reviews (*score-dl-user*). These stereotypes can lead to the detection of negative influencers in Yelp.

6.1.2 Definition of negative influencer stereotypes

As we have seen above, our methodology starts from the multi-dimensional social network-based model, formulates some hypotheses and aims at verifying them using an inferential campaign based on social network analysis. This campaign makes use of a number of concepts, stereotypes and definitions that we introduce in this section. Instead, the way they are exploited to prove the hypotheses and, more in general, to extract useful knowledge is described in Section 6.2.

The first concept we introduce is a stereotype, namely the *k-bridge*. Specifically, a *k-bridge* is a Yelp user who reviewed businesses belonging to exactly k different macro-categories of Yelp. A user who reviewed businesses of only one macro-category is a *non-bridge*. Finally, we use the generic term *bridge* to denote a k-bridge

such that $k > 1$. Given a k-bridge u_p of $\mathcal{U}$, where $\mathcal{U}$ is the overall user network corresponding to Yelp, there are k nodes $n_{1_p}, n_{2_p}, \cdots, n_{k_p}$ associated with her, one for each macro-category containing at least one review performed by her.

After having introduced the *k-bridge*, we present some other stereotypes, namely the power user and the double-life user. More specifically, let $\mathcal{C}_i \in \mathcal{Y}$ be one of the macro-categories of Yelp.

Let rn_i be the average number of reviews of $\mathcal{C}_i$. Let b_p be a Yelp bridge and let $CSet_p$ be the set of the macro-categories that received reviews from b_p. Then:

- b_p is defined as a *power user* if, for each macro-category $\mathcal{C}_j \in CSet_p$, the number of her reviews is greater than or equal to $2 \cdot rn_j$.
- b_p is defined as a *(x,y) access double-life user* (*access-dl-user*, for short) if both the following conditions hold:
  - for a subset $CSet_{p_x} \subset CSet_p$ of x macro-categories, the number of reviews of each $\mathcal{C}_j \in CSet_{p_x}$ is greater than or equal to $2 \cdot rn_j$;
  - for a subset $CSet_{p_y} \subset CSet_p$ of y macro-categories, such that $CSet_{p_x} \cap CSet_{p_y} = \emptyset$, the number of reviews of each $\mathcal{C}_k \in CSet_{p_y}$ is less than or equal to $\frac{1}{2} \cdot rn_k$.

Double-life users play an extremely interesting role because they are very rare. Therefore, we deepen our investigation on them and introduce a second kind of double-life users. Specifically, let b_p be a Yelp bridge. Then b_p is defined as a (x, y) *score double-life user* (*score-dl-user*, for short) if both the following conditions hold:

- for a subset $CSet_{p_x} \subset CSet_p$ of x macro-categories, the average number of stars that b_p assigned to the corresponding businesses is higher than or equal to 4;
- for a subset $CSet_{p_y} \subset CSet_p$ of y macro-categories, such that $CSet_{p_x} \cap CSet_{p_y} = \emptyset$, the average number of stars that b_p assigned to the corresponding businesses is lower than or equal to 2.

In order to make our inferential campaign on negative reviews and reviewers complete, we need to introduce a further network that we call *Negative Reviewer Network* $\overline{\mathcal{U}} = \langle \overline{N}, \overline{A} \rangle$. $\overline{N}$ is the set of nodes of $\overline{\mathcal{U}}$. There is a node $n_i \in \overline{N}$ for each Yelp user who made at least one negative review. There is an arc $a_{pq} = (n_p, n_q)$ if there exists a friendship relationship between the user u_p, corresponding to n_p, and the user u_q, corresponding to n_q.

6.1.3 Hypothesis definition

Starting from this theoretical background, we aim at answering the three questions mentioned in the Introduction. In particular, we use the above model and stereotypes to design and perform a social network analysis-based campaign aiming at evaluating some hypotheses that we synthesize in the following:

- First of all, the review mechanism of Yelp is based on a scale from 1 to 5 stars. This is similar to the review mechanisms encountered in several other social media. In this context, we formulate the following:

  Hypothesis 1 (H1) - The star-based review system of Yelp is positively biased.

  In the scale adopted by Yelp, 1 means "absolutely bad" and 5 means "fantastic". A review with 2 stars is still negative, but 3 stars already denote a positive review. In other words, the review mechanism of Yelp makes it more probable that users release positive reviews. Unless the experience was really bad, the review will almost always be positive. This is confirmed by how Yelp itself labels the stars (1 - "Eek! Methinks not"; 2 - "Meh. I've experienced better"; 3 - "A-OK"; 4 - "Yay! I'm a fan"; 5 - "Woohoo! As good as it gets!").

  On the other hand, if we consider this review mechanism from a more formal and theoretical viewpoint, we can observe that it is based on a Likert scale, which was already shown to be asymmetric and positively biased [26, 496, 76].
- We think that the stereotypes introduced above can help very much in evaluating negative reviews and influencers. As for a specific kind of stereotype, i.e., the double-life users, we formulate the following:

  Hypothesis 2 (H2) - access-dl-users and score-dl-users play a key role in negative reviews.

  To understand the reasoning behind this hypothesis, consider score-dl-users. Clearly, they can be partitioned into two sets. The former is made up of users who mainly write positive reviews and few negative reviews. These are basically positive users who, for some reasons, had a bad experience with some businesses. So, what drove them to write negative reviews, considering that they are keen to write positive ones? A user assigns a 1-star score to a business when her expectations were not satisfied. This was already investigated in literature (see, for instance, [305]), where it was proved that a high discrepancy between the others' opinions and the experience of a user is the main driver for her to write a negative review.

  The latter set of access-dl-users is much more peculiar. It comprises those users who generally write negative reviews but, in some cases, release positive ones. These users have probably developed very severe criteria for evaluating businesses, leading them to be satisfied only rarely.
- We have already discussed about the multi-dimensionality of our model. One of its main dimensions is friendship. Actually, it is well known that this relationship plays a key role in social networks [80, 546, 77]. Starting from these results, it is reasonable to formulate the following:

Hypothesis 3 (H3) - A user has a strong influence on her friends when doing negative reviews.

This could seem obvious. In past literature it has been proved that users are influenced by others when writing reviews. In particular, it has been found that users tend to have a positive opinion of a product/service if it has been positively commented by other users [162].

In addition, people generally trust more those users sharing their personal profile on online review platforms [244]. It was found that a personal information disclosure is crucial for the spread of positive comments about a product/service, because the possibility of associating information with a particular person gives a boost in the overall perceived confidence. All of this is amplified when users share a common geographical location. This reasoning can also be applied to relationships like friendship, because personal information is certainly disclosed between friends.

Here, we hypothesize that the influence exerted by friends is valid not only for positive reviews but also for negative ones, possibly leading to a phenomenon of negative influence between friends.

- Another stereotype introduced above that could play an important role as negative influencer is the bridge one. As for it, we formulate the following:

  Hypothesis 4 (H4) - Bridges have a much greater influence power than non-bridges.

  If Yelp can be modeled as a network of different communities, each corresponding to a given business macro-category, it is immediate to think of bridge users as special ones, capable of facilitating information diffusion from a community to another. Bridge users have a position of power in the network, and this power can even be measured [341]. If we look at classical centrality measures in social network analysis, it is easy to argue that bridge users have a high betweenness centrality value. On the other hand, if we look at reviews, it is plausible that a bridge could expand the negative conception of a brand from a category to another which both the bridge and the brand belong to.
- The previous reasoning about the correlation between bridges and betweenness centrality paves the way to think that centralities play a key role in the diffusion of negative reviews. In particular, it is reasonable to make the following hypothesis:

  Hypothesis 5 (H5) - There is a correlation between degree and/or eigenvector centrality and the capability of being negative influencer.

  Degree centrality tells us which nodes have the highest number of relationships in a network. These are probably power users, if we consider our stereotypes.

They certainly are important users, because they are densely connected. On the other hand, eigenvector centrality can help us to identify influential users, who do not like to appear as such (the so called grey eminences or grey cardinals). Those kinds of users are often connected to few nodes, each having a high number of relationships with the other users [418]. These two centrality measures can be useful to find negative influencers in Yelp.

6.1.4 Preliminary analysis of negative influencers stereotypes

We collected the data necessary for the activities connected with our inferential campaign from the Yelp website at the address `https://www.yelp.com/dataset`. In order to extract information of interest from available data, we had to carry out a preliminary analysis. A first result concerns the presence of 10,289 businesses whose category did not belong to any of the Yelp macro-categories, and 482 businesses that did not have any category associated with them (recall that in Yelp a business can belong to one or more categories). Since the total number of businesses was 192,609, we decided to discard these two kinds of businesses, because the amount of data removed was insignificant while their presence would have led to procedural problems.

At this point, we analyzed the distribution of the categories among the macro-categories. We report the result obtained in Figure 6.1. As we can see from this figure, the macro-category "Restaurants" has a much greater number of categories than the other ones.

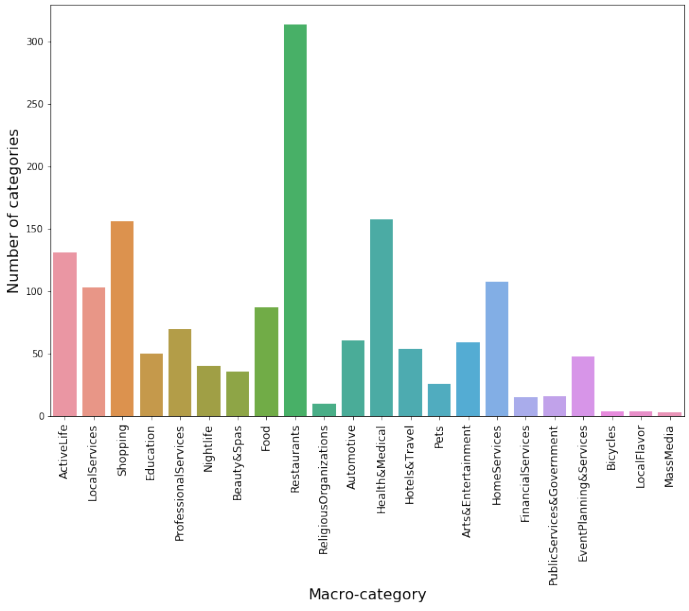


Fig. 6.1: Distribution of the categories inside the Yelp macro-categories

Figure 6.2 shows the average number of reviews per user for each macro-category. As we can see, the three macro-categories with the highest average number of reviews are "Restaurants", "Food" and "Nightlife". Furthermore, in Figure 6.3, we show the same distribution for bridges only. We can see that the three macro-categories with the highest number of reviews are always the same. However, the average number of reviews is generally higher for bridges than for normal users. Therefore, we can conclude that bridges not only tend to review businesses of different macro-categories (and this happens by definition of bridge itself) but also to do more reviews than non-bridges.

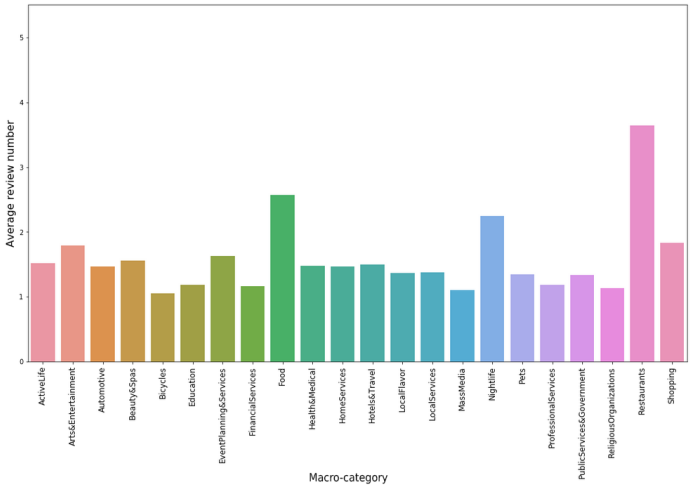


Fig. 6.2: Average number of business reviews made by Yelp *users* for each macro-category

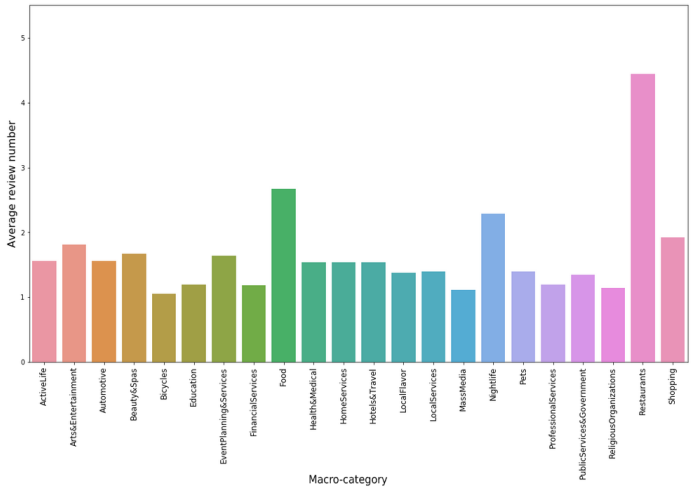


Fig. 6.3: Average number of business reviews made by Yelp *bridges* for each macro-category

In Figure 6.4, we report the distribution of access-dl-users against k. From the analysis of this figure, we observe that the number of access-dl-users is already very high for $k = 2$ and further increases for $k = 3$; then, it decreases very quickly and becomes almost negligible for $k > 4$.

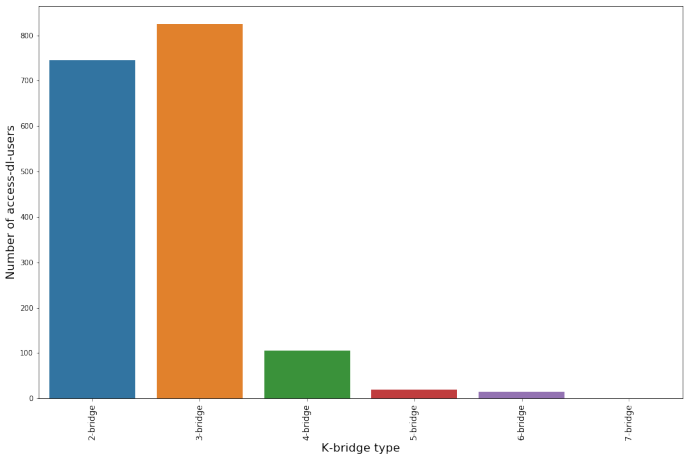


Fig. 6.4: Distribution of access-dl-users against k

We start looking at the access-dl-users corresponding to the simplest case of bridges, namely 2-bridges. Table 6.1 shows the total number of 2-bridges, the number of (1,1) access-dl-users and the number of power users, together with their corresponding percentage of the overall number of 2-bridges. This table shows that (1,1) access-dl-users and power users represent very small fractions of the overall set of 2-bridges.

Type of users	*Number and percentage*
2-bridges	427130 (100%)
(1,1) access-dl-users	745 (0.17%)
power users	375 (0.087%)

Table 6.1: Numbers and percentages of 2-bridges, access-dl-users and power users in Yelp

We continue by examining all the k-bridges as k grows, until at least one of them is an access-dl-user or a power user. We can observe that this condition occurs for $k \leq 6$. The corresponding numbers and percentages are shown in Tables 6.2 - 6.5. From the analysis of these tables, we can see how the number of k-bridges decreases as k increases, but the decrease is not fast. On the other hand, the number of access-dl-users decreases very rapidly, about one order of magnitude at each step. The number of power users decreases more slowly.

Type of users	*Number and percentage*
3-bridges	245123 (100%)
(1,2) access-dl-users	450 (0.18%)
(2,1) access-dl-users	374 (0.15%)
power users	200 (0.081%)

Table 6.2: Numbers and percentages of 3-bridges, access-dl-users and power users in Yelp

Type of users	*Number and percentage*
4-bridges	147101 (100%)
(1,3) access-dl-users	19 (0.013%)
(2,2) access-dl-users	59 (0.040%)
(3,1) access-dl-users	28 (0.019%)
power users	35 (0.023%)

Table 6.3: Numbers and percentages of 4-bridges, access-dl-users and power users in Yelp

Type of users	*Number and percentage*
5-bridges	91680 (100%)
(1,4) access-dl-users	6 (0.007%)
(2,3) access-dl-users	11 (0.012 %)
(3,2) access-dl-users	3 (0.003%)
(4,1) access-dl-users	0 (0%)
power users	14 (0.015%)

Table 6.4: Numbers and percentages of 5-bridges, access-dl-users and power users in Yelp

Type of users	*Number and percentage*
6-bridges	63708 (100%)
(1,5) access-dl-users	0 (0%)
(2,4) access-dl-users	0 (0%)
(3,3) access-dl-users	1 (0.002%)
(4,2) access-dl-users	2 (0.003%)
(5,1) access-dl-users	11 (0.017%)
power users	11 (0.017%)

Table 6.5: Numbers and percentages of 6-bridges, access-dl-users and power users in Yelp

6.2 Results

6.2.1 Investigating the Hypothesis H1

A user can assign a number of stars between 1 and 5 to a business in Yelp. The higher the number of stars, the better her rating is. Therefore, we decided to study the reviews of users focusing on the number of stars that they assigned to businesses.

Figure 6.5 shows the average number of stars that users assigned to the businesses of each macro-category. As we can see from this figure, this number is very high as it is always greater than 3. As previously pointed out, this is actually not very surprising because the mechanism based on stars follows a Likert scale and, in literature, it is well known that this scale is generally positively biased [26, 496, 76].

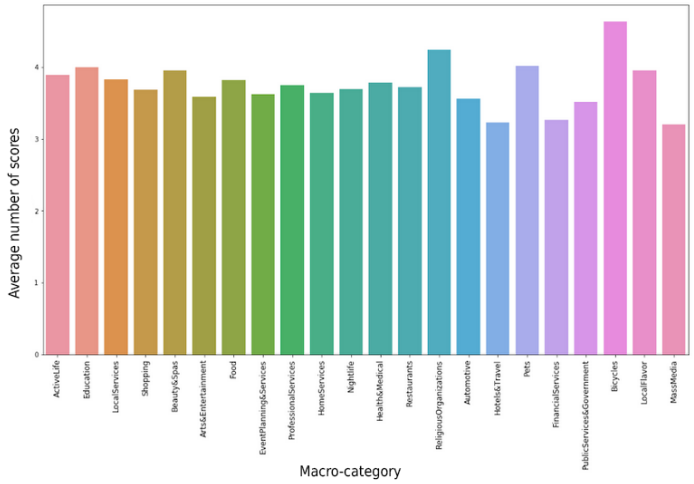


Fig. 6.5: Average number of stars for each macro-category of Yelp

In Table 6.6, we report the mean, standard deviation and mode of the number of stars assigned by bridges and non-bridges to all businesses. As we can see from this table, there is no substantial difference in this type of behavior between bridges and non-bridges.

Statistical Parameter	*Bridges*	*Non-bridges*
Mean	3.73	3.57
Standard Deviation	1.44	1.72
Mode	5	5

Table 6.6: Values of mean, standard deviation and mode of the number of stars assigned by bridges and non-bridges to all businesses

From the results of Table 6.6, it is clear that it makes no sense to talk about power users in the star-based analysis, because almost all users have the same behavior and assign a high number of stars to almost all businesses. All these tests allow us to define the following:

Implication 1: The star-based review system of Yelp is positively biased. Indeed, almost all users assign a high number of stars to almost all businesses.

Implication 1 is clearly a confirmation of the correctness of the Hypothesis H1.

6.2.2 Investigating the Hypothesis H2

In Figure 6.6, we report the distribution of score-dl-users against k. From the analysis of this figure we note that it follows a power law. If we compare this figure with Figure 6.4, we observe that for $k = 2$, the number of score-dl-users is much smaller than the one of access-dl-users. However, the decrease of the number of score-dl-users when k increases is much smaller because they are different from 0 until to $k = 14$.

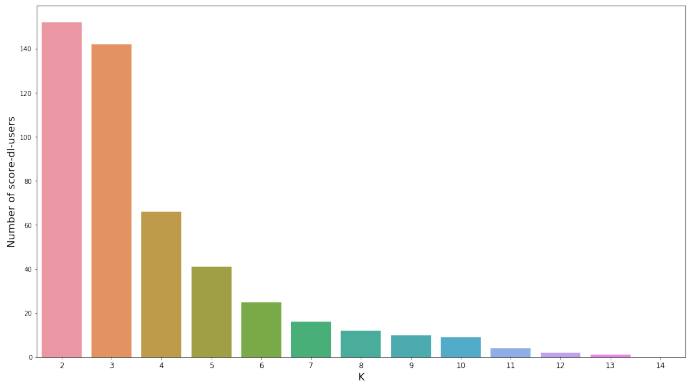


Fig. 6.6: Distribution of score-dl-users against k

We continued our analysis by verifying whether score-dl-users and access-dl-users were the same people or not. We carried out this analysis with $k = 6$, because we had no access-dl-users with higher values of k. In this case, we could see that the intersection of the two sets was empty.

To better understand the main features of score-dl-users we considered those corresponding to 7-bridges. These users were 16 (see Figure 6.6), a number that allowed us to examine in detail each review carried out by them. During this analysis we found several interesting knowledge patterns. More specifically, we observed that (1,6) and (6,1) score-dl-users show a completely different behavior from the other 7-bridges. In fact, in this case, each (1,6) score-dl-user assigned positive scores to all the business of the only macro-category that she positively reviewed. Similarly, each (6,1) score-dl-user assigned negative values to all the businesses of the only macro-category that she negatively reviewed. This can be justified thinking that users have a strong interest in that macro-category and so they developed more accurate and stable evaluation criteria for the businesses belonging to it.

As for the other 7-bridges, we found that (2,5), (3,4), (4,3) and (5,2) score-dl-users show a less extreme behavior, in the sense that they do not tend to give always positive or always negative ratings to all the businesses of a given macro-category.

We then repeated the previous analyses for the last category of access-dl-users that we had available, namely the 6-bridges, to verify if the particular behavior of score-dl-users was typical of this kind of double-life user or if it was something common. Actually, 6-bridge access-dl-users were 13; therefore, we were able to make a detailed analysis of each review performed by each user also in this case. We examined (1,5), (2,4), (3,3), (4,2) and (5,1) access-dl-users and we did not find substantial differences in the behavior of these five categories of users. This appeared as a confirmation of the singularity of the behavior observed for (1,6) and (6,1) score-dl-users. The previous analyses suggest the following:

Implication 2: *(a)* Score-dl-users play a key role in negative reviews. *(b)* They are very keen on negatively judging the macro-category they mostly attend.

Implication *2(a)* confirms the correctness of our Hypothesis H2. But there is much more. In fact, Implication *2(b)* was an unexpected result that prompted us to carry out a further experiment to have a confirmation. In it, we considered k-bridges, with $3 \leq k \leq 8$, and computed the percentage of them who negatively reviewed the macro-category of businesses they attended the most. Afterwards, we computed the same percentage taking into account only k-bridges that were score-dl-users. The results obtained are shown in Table 6.7. They represent an extremely strong confirmation of the previous qualitative analysis.

k	*Percentage of k-bridges*	*Percentage of score-dl-users k-bridges*
3	4.35%	91.5%
4	4.03%	79%
5	3.65%	61%
6	2.40%	63%
7	2.11%	56%
8	1.55%	33%

Table 6.7: Percentages of k-bridges and score-dl-users k-bridges who negatively reviewed the macro-category they mostly attended

As we have seen, the definition and behavior of score-dl-users are based on the number of stars assigned by a user to a business during a review. We have already said that this type of score is based on a Likert scale and, therefore, it is positively biased [26, 496, 76]. In order to overcome this problem, in the literature authors suggest evaluating the text of the reviews and to make a sentiment analysis on it [340, 338]. We carried out this activity using two well-known sentiment analysis

tools. The first is TextBlob[1], which, given a text, specifies if the corresponding polarity is positive, negative or neutral. We applied TextBlob to users' review texts. The results obtained are reported in Table 6.8. From the analysis of this table we can see that the difference between the score based on stars and the polarity based on sentiment analysis is equal to 15%.

Parameters	*Value obtained by applying TextBlob*
Reviews	6,685,902
Reviews with a number of stars less than or equal to 2 (negative reviews)	1,544,553
Reviews classified as negative by TextBlob	847,359
Reviews with a number of stars greater than or equal to 3 (positive reviews)	5,141,347
Reviews classified as positive by TextBlob	5,781,007
Reviews classified as neutral by TextBlob	57,536
Negative reviews classified as positive	823,414
Positive reviews classified as negative	154,176
Positive reviews classified as neutral	30,914
Negative reviews classified as neutral	26,620

Table 6.8: Comparison between the review score based on stars and the review polarity obtained by applying TextBlob

The second sentiment analysis tool we considered is Vader [317]. Also in this case, we applied it to the users' review texts. The results obtained are shown in Table 6.9. The analysis of this table confirms the very low difference between the score of the star-based reviews and the polarity of the review texts (in fact, in this case, this difference is equal to 14%).

Parameter	*Value obtained by applying Vader*
Reviews	6,685,902
Reviews with a number of stars less than or equal to 2 (negative reviews)	1,544,553
Reviews classified as negative by Vader	982,102
Reviews with a number of stars greater than or equal to 3 (positive reviews)	5,141,347
Reviews classified as positive by Vader	5,649,489
Reviews classified as neutral by Vader	54,311
Negative reviews classified as positive	724,241
Positive reviews classified as negative	184,557
Positive reviews classified as neutral	31,542
Negative reviews classified as neutral	22,767

Table 6.9: Comparison between the review score based on stars and the review polarity obtained by applying Vader

This allows us to conclude that score-based evaluations are generally confirmed by the sentiment analysis performed on the corresponding reviews.

[1] `https://textblob.readthedocs.io`

6.2.3 Investigating the Hypothesis H3

At this point, we analyzed how users influence each other with regard to negative reviews. We took into consideration the network of friendships $\mathcal{Y}^f$ since it is easier for a user to have characteristics more similar to her friends than to people she does not know, due to the principle of homophily [435]. Therefore, the ability to influence someone and/or to be influenced by her is presumably greater with friends than with others.

As a first analysis, for each macro-category, we considered the percentage of users such that they, and at least one of their friends, reviewed the same business negatively. The results obtained are shown in Figure 6.7. From the analysis of this figure we can see how the percentages are extremely low. The macro-category with the highest percentage is "Restaurant", followed by "Nightlife" and "Food". This result can be explained taking into account that a person often attends restaurants or nightclubs with her friends. Therefore, it is not unlikely that her negative judgement of a business may lead to (or, on the contrary, may be caused by) a negative judgement of one or more of her friends.

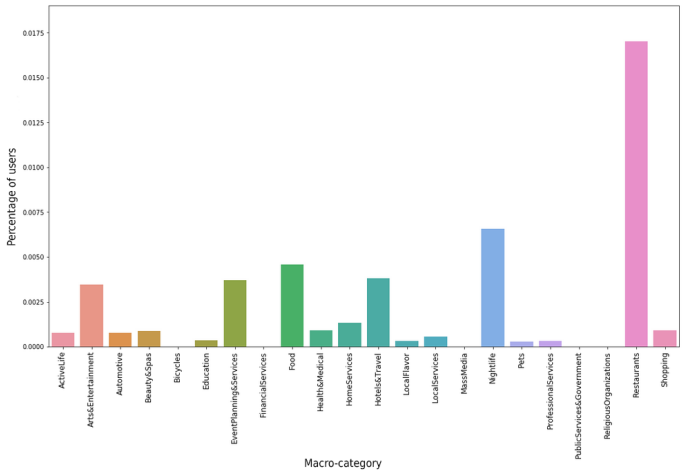


Fig. 6.7: Percentages of *users* such that they, and at least one of their friends, reviewed the same business negatively

We repeated the analysis by distinguishing bridges from non-bridges. The corresponding results are shown in Figures 6.8 and 6.9. From the analysis of these figures we observe higher values for bridges than for non-bridges. For example, the value of "Nightlife" for bridges is more than 4 times the value for non-bridges. Similarly, "Food", in case of bridges, has a percentage more than 7 times higher than for non-bridges.

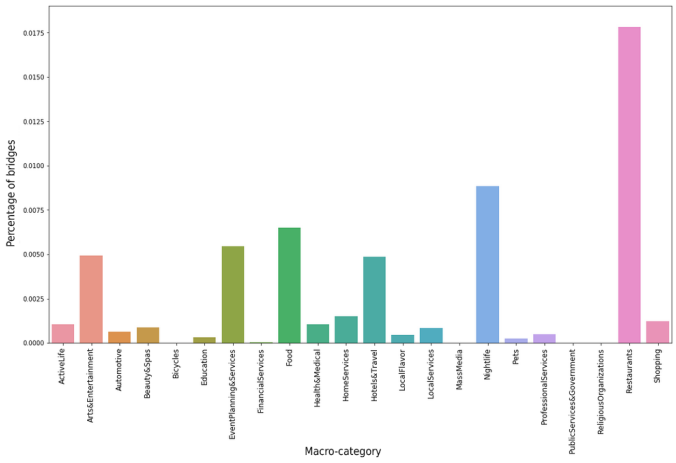


Fig. 6.8: Percentages of *bridges* such that they, and at least one of their friends, reviewed the same business negatively

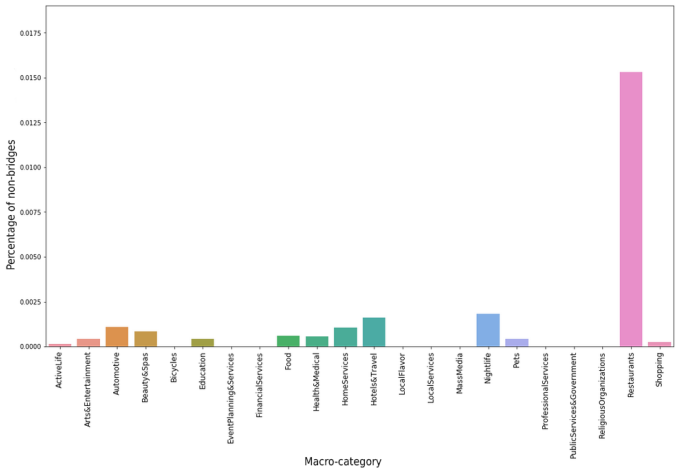


Fig. 6.9: Percentages of *non-bridges* such that they, and at least one of their friends, reviewed the same business negatively

To prove the statistical significance of our results we adopted a null model to compare our findings with those obtained in an unbiasedly random scenario. Specifically, we built our null model by shuffling the negative reviews among users in our dataset. In this way, we left unaltered all the original features with the exception of the distribution of negative reviews, which became unbiasedly random in the null model. After that, we repeated our analysis on the null model. The results obtained are reported in Figure 6.10. Comparing this figure with Figure 6.7, we can see that there is a certain similarity in the distributions; indeed, many of the macro-categories that had the highest values in Figure 6.7 continue to have the highest values in Figure 6.10. However, in this last case, the values of the percentages are

several orders of magnitude smaller. Therefore, we can conclude that the behavior observed in Figure 6.7 is not random but it is the result of the reference context.

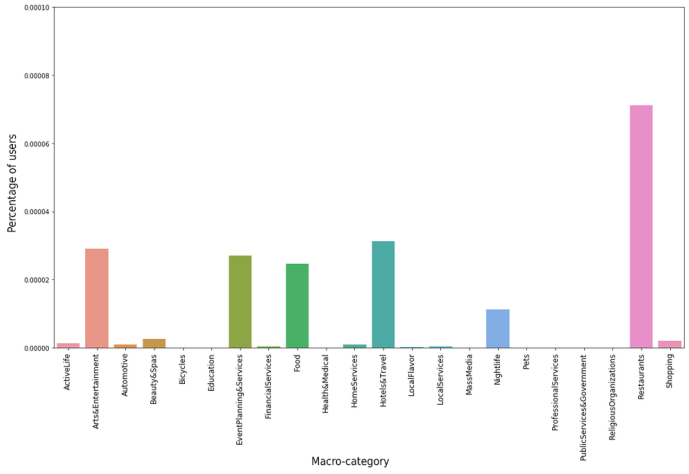


Fig. 6.10: Percentages of *users* in the null model such that they, and at least one of their friends, reviewed the same business negatively

At this point, for each macro-category, for each user who reviewed a given business negatively, we computed the percentage of her friends who, having reviewed the same business, made a negative review. The results obtained are shown in Figure 6.11. As we can see from this figure, the percentage values are very high for almost all macro-categories.

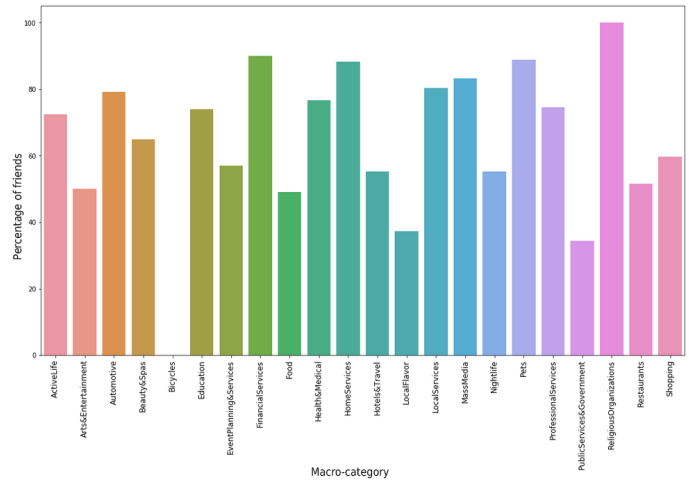


Fig. 6.11: Percentages of friends who, having reviewed the same business as a *user* who reviewed a business negatively, also provided a negative review

Figures 6.12 and 6.13 show the same distributions, but for bridges and non-bridges. From the analysis of these figures, it can be observed that the phenomenon is always strong, regardless of whether or not a user is a bridge. An interesting knowledge pattern to observe is that there is a strong polarization on the macro-categories especially in the case of non-bridges. In fact, the percentages of friends influenced by them are either above 90% or null.

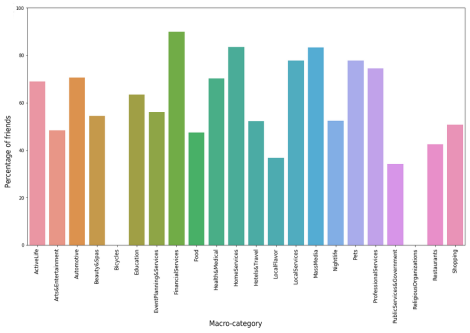


Fig. 6.12: Percentages of friends who, having reviewed the same business as a *bridge* who reviewed a business negatively, also provide a negative review

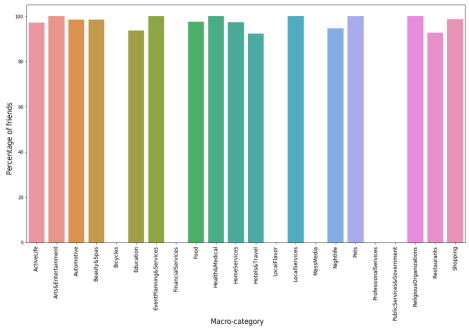


Fig. 6.13: Percentages of friends who, having reviewed the same business as a *non-bridge* who reviewed a business negatively, also provide a negative review

All the results shown above allow us to deduce the following:

Implication 3: A user has a very high influence on her/his friends when doing negative reviews.

This implication represents a confirmation of the correctness of our Hypothesis H3.

6.2.4 Investigating the Hypothesis H4

In order to evaluate the Hypothesis H4, we started with the computation of the average percentage of users who, having made a negative review in a category, have at least $X\%_{oo}$ of their friends who negatively reviewed a business in the same category. The values of X that we considered are 1, 2, 3, 5, 10 and 100. As an example, in Figure 6.14, we report the results obtained in the case of $X = 5$. As we can see from this figure, the percentages are some orders of magnitude greater than the ones of Figure 6.10. The macro-categories with the highest values are the same as before, i.e., “Restaurants”, “Food” and “Nightlife”.

As in the previous case, we distinguished bridges from non-bridges. The results of the corresponding analysis are shown in Figures 6.15 and 6.16. These figures,

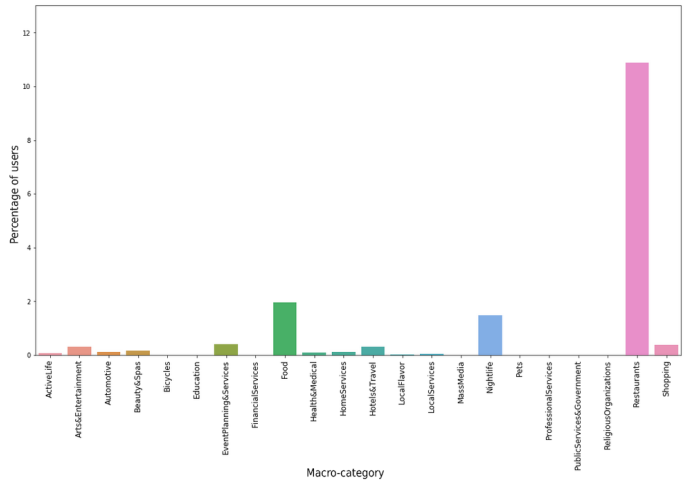


Fig. 6.14: Average percentages of *users* who, having made a negative review in a macro-category, have at least $X\%_{oo}$ of their friends who reviewed a business in the same macro-category negatively

along with the previous ones involving bridges and non bridges, allow us to define the following:

Implication 4: Bridges have a much greater power of influence than non-bridges.

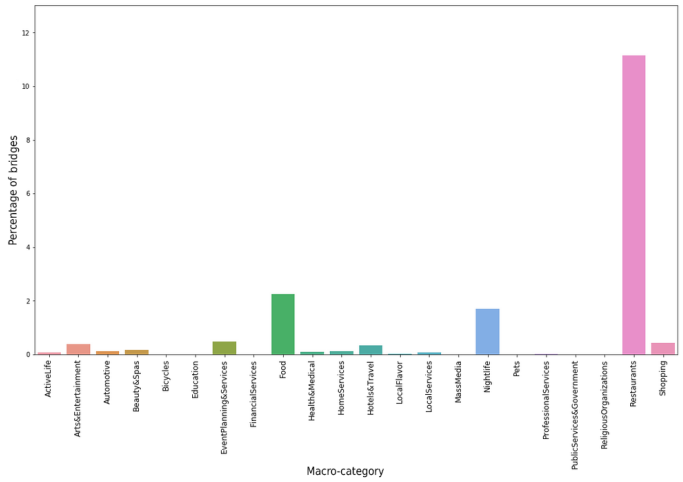


Fig. 6.15: Average percentages of *bridges* who, having made a negative review in a macro-category, have at least $X\%_{oo}$ of their friends who reviewed a business in the same macro-category negatively

Again, we made the comparison with the null model. The results obtained for $X = 5$ are reported in Figures 6.17, 6.18 and 6.19. From the examination of these figures, we can see how results obtained are not random but they are intrinsic to

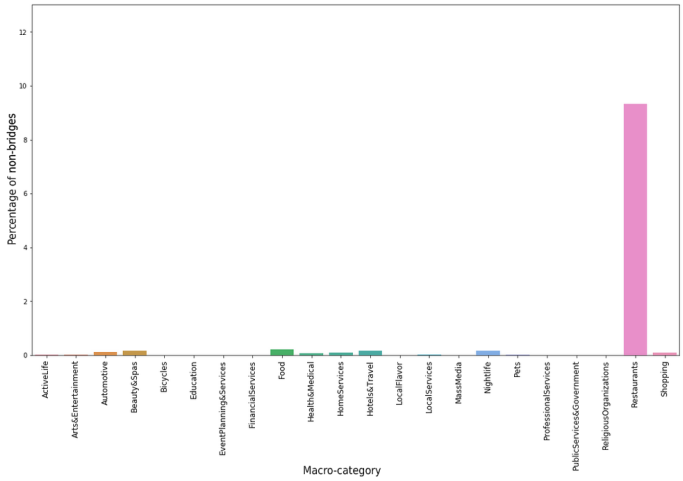


Fig. 6.16: Average percentages of *non-bridges* who, having made a negative review in a macro-category, have at least $X‰$ of their friends who reviewed a business in the same macro-category negatively

Yelp. Note that the non-randomness can be observed for *bridges* but generally not for *non-bridges*; this is important because it allows us to conclude that this property characterizes bridges against non-bridges.

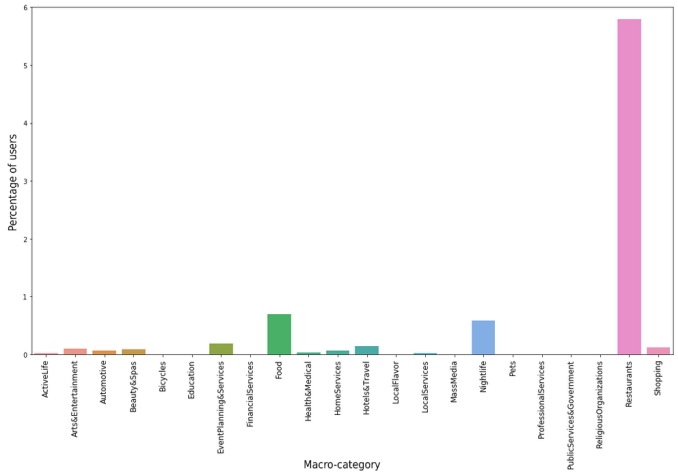


Fig. 6.17: Average percentages of *users* in the null model who, having made a negative review in a macro-category, have at least $X‰$ of their friends who reviewed a business in the same macro-category negatively

Implication 4 represents a confirmation that our Hypothesis H4 was correct.

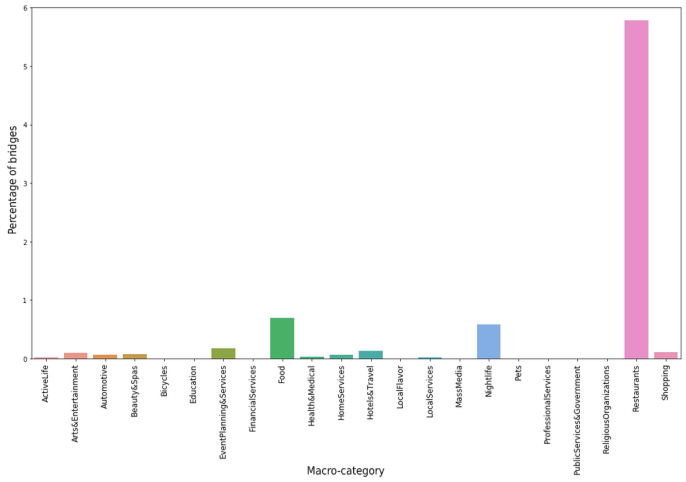


Fig. 6.18: Average percentages of *bridges* in the null model who, having made a negative review in a macro-category, have at least $X‰$ of their friends who reviewed a business in the same macro-category negatively

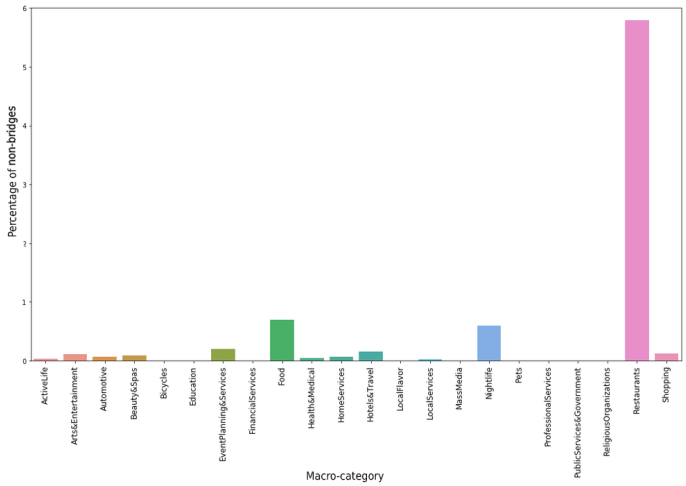


Fig. 6.19: Average percentages of *non-bridges* in the null model who, having made a negative review in a macro-category, have at least $X‰$ of their friends who reviewed a business in the same macro-category negatively

6.2.5 Investigating the Hypothesis H5 and defining a profile of negative influencers in Yelp

To investigate the correctness of the Hypothesis H5 we considered the *Negative Reviewer Network* $\overline{U} = \langle \overline{N}, \overline{A} \rangle$ introduced in Section 6.1.2.

The analysis of this network allowed us to focus on users who reviewed some businesses negatively, because, as we saw in the previous analysis, they are uncommon. Firstly, we computed the number of nodes, the number of edges, the clustering

coefficient and the density of $\overline{\mathcal{U}}$ and we compared them with the same parameters as $\mathcal{U}$. Results are shown in Table 6.10.

	$\mathcal{U}$	$\overline{\mathcal{U}}$
Number of nodes	1637138	743178
Number of edges	7392305	2199987
Average clustering coefficient	0.043	0.039
Density	0.00000551619	0.00000796645

Table 6.10: Characteristics of $\mathcal{U}$ and $\overline{\mathcal{U}}$

From the analysis of this table we can observe that the number of users who made at least one negative review is 45.39% of total users. As for the average clustering coefficient and the density, we found that their values do not present significant differences between $\mathcal{U}$ and $\overline{\mathcal{U}}$.

At this point, we computed the distribution of users for $\overline{\mathcal{U}}$; it is shown in Figure 6.20. As we can see from this figure, it follows a power law.

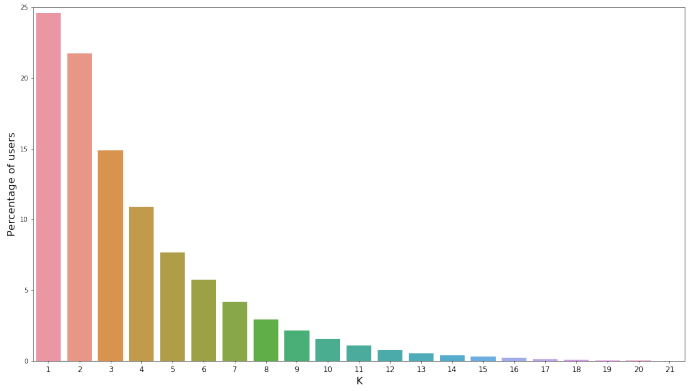


Fig. 6.20: Distribution of users of $\overline{\mathcal{U}}$ against k

After studying the basic parameters of $\overline{\mathcal{U}}$, we computed the degree centrality of the nodes of this network. In particular, we focused on the users with the highest values of degree centrality. More specifically, we considered the top X% users, $X \in \{1, 5, 10, 20\}$. Observe that as X decreases, the corresponding top users are increasingly central, i.e., increasingly strong. In Figure 6.21, we show the distributions against k for the top X% of users with the highest degree centrality. Note that for $X = 20$, the distribution follows a power law, even if it is flatter than the one of Figure 6.20, which referred to all users. As X decreases, we can see how the distribution becomes flatter and flatter, moving to the right and tending to a Gaussian shape. This allows us to conclude that more central users (i.e., those with the highest degree

centrality) tend to be stronger also as k-bridges (i.e., characterized by an increasingly higher value of k).

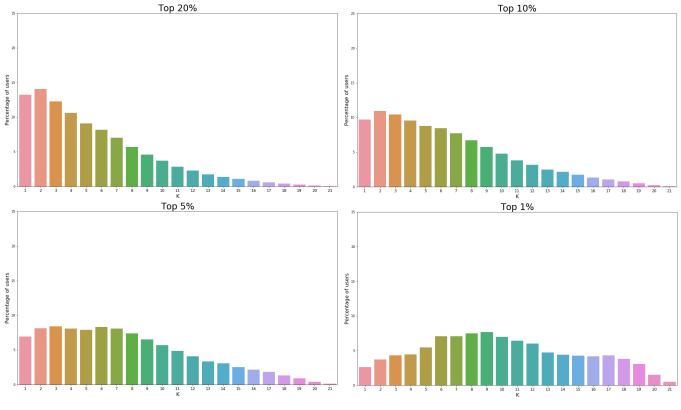


Fig. 6.21: Distributions of the top X% of users with the highest degree centrality against k

Instead, in Figure 6.22, we show the user distributions against k for the top X% of users with the highest eigenvector centrality. The trend of these distributions as X decreases is very similar to (although slightly less marked than) the one of the degree centrality.

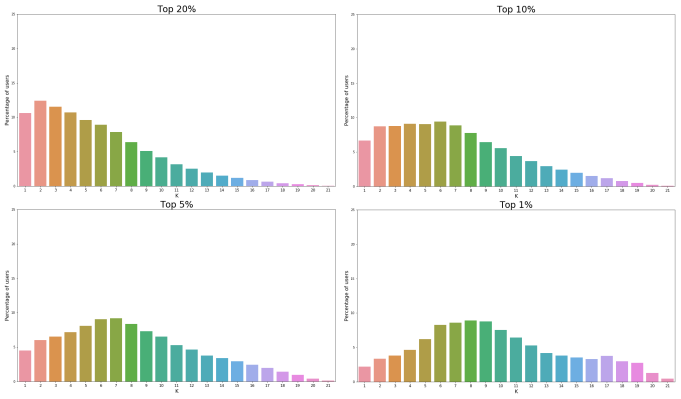


Fig. 6.22: Distributions of the top X% of users with the highest eigenvector centrality against k

Figure 6.23 shows the user distributions against k for the top X% of users with the highest PageRank. Also in this case, we have a similar trend, although the variations of the distributions as X decreases are much more attenuated, compared to the two previous cases. The last three figures allow us to define the following:

Implication 5: There is a correlation between k-bridges and top central users.

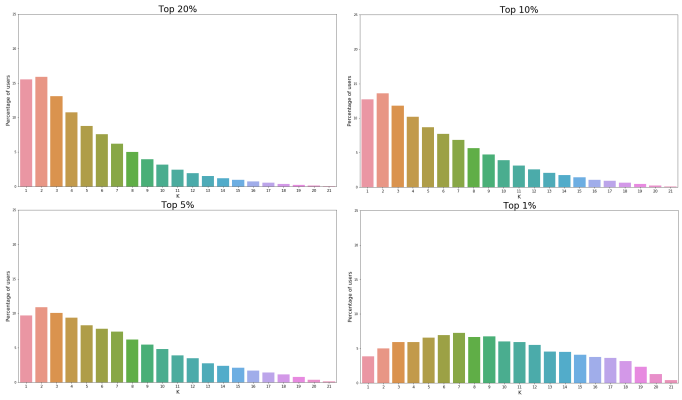


Fig. 6.23: Distributions of the top X% of users with the highest PageRank against k

Implication 5 is valid especially for the top central users based on degree centrality. This result, along with the previous ones, is extremely important because it allows us to determine which are the main negative influencers in Yelp. In fact, we can define the following:

> ***Implication 6: The main negative influencers in Yelp are score-dl-users who simultaneously are top central users (according to degree and/or eigenvector and/or PageRank centrality measures).***

Implication 6 not only confirms the correctness of the Hypothesis H5, but goes much further. In fact, it defines a profile of the negative influencers in Yelp and, consequently, provides a way to detect them.

6.3 Discussion

6.3.1 Reference context

In the previous sections, we have investigated the phenomenon of negative reviews in Yelp and, then, we have characterized negative influencers in this social medium. In the past, different research papers have focused on the consequences that user-written reviews have on businesses and, generally, on the market. As a first step in this scenario, it is interesting to understand what makes customer reviews helpful to a consumer in her process of making a purchase decision. With regard to this, in [550], the authors first collect reviews made on `Amazon.com`. Then, they distinguish between two different product types, namely: *(i)* search goods, for which a consumer can obtain information on their quality before purchasing them; *(ii)* experience goods, which are products requiring a purchase before evaluating their quality. This product categorization plays a key role in understanding what a consumer perceives more from a review. Indeed, moderate reviews are more helpful

than extreme (i.e., strongly positive or negative) ones for experience goods, but not for search goods. Furthermore, longer reviews are generally perceived as more helpful than shorter ones, but this effect is greater for search goods than for experience goods.

Another interesting contribution in this scenario is reported in [673], in which the authors introduce several factors that can influence the decision making process of consumers about their purchases. Indeed, the authors of [673] strive to understand the key elements that guide a user in the purchase of a certain product. They propose a model taking systematic factors (e.g., the quality of online reviews) and heuristic ones (e.g., the quantity of online reviews) into account. They test this model on 191 users and obtain interesting results. In fact, they identify important factors to care about; these are argument quality, source credibility, and perceived quantity of reviews. They empirically prove that consumers receiving reviews from credible sources and perceiving the quantity of reviews as large tend to perceive the topics in online reviews as more informative and persuasive. This means that if consumers find review sources to be credible, their purchase intention is usually higher. Finally, they also show that consumers are more likely to purchase products with many online reviews rather than with few ones.

Several authors have investigated the impact of positive and negative reviews. For instance, the authors of [162] examine how a positive Electronic Word of Mouth (hereafter, eWOM) can affect other users' purchasing decisions. Indeed, eWOM is strictly related to the online reviews phenomenon, which can be regarded as a special case of it. Generally, eWOM is based on an analysis of costs and benefits. The authors investigate the psychological motivations beneath the spread of positive reviews. They take a sample dataset from the `OpenRice.com` platform, one of the most successful review platforms in Hong Kong and Macau. Through a questionnaire, they asked people who wrote reviews on this website their motivations. Starting from the received answers, they build a model based on different features, namely the eWOM intention of consumers, the reputation, the reciprocity, the sense of belonging, the pleasure to help, the moral obligation and the self-efficacy of knowledge. They show that their model is capable of representing the behavior of users when they share (positive) personal experiences on such online platforms.

The influence of positive reviews of businesses has been studied from many other points of view. For example, in [358], the authors analyze celebrity sponsorships in the context of for-profit and non-profit marketing. They actually find that famous people can influence the appreciation one has for a product or service, in a positive or negative direction. This suggests that it makes sense studying who negative influencers are, how they behave and how they can be detected in an online plat-

form. Not limited to celebrities, people are more incline to follow users disclosing their personal information [244]. The members of an online community rate reviews containing descriptive identity information more positively, and the prevalence of identity information disclosure by reviewers is associated with increased subsequent sales of online products. In addition, the shared geographical location increases the relationship between disclosure and product sales.

Wrapping up these important results, we can say that buyers are influenced by positive eWOM, especially if it is performed by nearby identifiable users; even more, celebrities can change the appreciation that people have for a product or a service. But the consequences are not just limited to customers. Even internal decision-making processes of businesses can be influenced by online review systems [12]. The diffusion of personal opinions through the Internet has radically changed the concept of reviewing a product or a service that one has in traditional media. In fact, online review platforms offer to users a space where they can express their *unfiltered* thoughts on products or services. In particular, eWOM encourages a two-way communication between a source and a reader, thus being more engaging. A very important result of [12] is that eWOM helps companies to obtain higher product and service evaluations and, if necessary, higher amounts of funding; furthermore, it influences the decision-making processes of companies, showing that its power is not limited only to buyers. The other important result of [12] is that the effect of negative eWOM is much greater than the one of positive eWOM.

Negative reviews open up many research issues. One of them is finding out what drives users to write negative reviews. Discontent, or "disconfirmation", with a product or service has been studied as a cause of this phenomenon. The authors of [305] define disconfirmation as the discrepancy between the expected evaluation of a product and the evaluation of the same product performed by experts. In particular, they find that a person is more likely to leave a review when the disconfirmation she encounters is great. They also find that the evaluation published by a person may not reflect her post-purchase evaluation in a neutral manner; indeed, the direction of such polarization is in agreement with disconfirmation.

The authors of [660] introduce a theory about the initial beliefs of a consumer when she is looking for a product. According to this theory, a consumer forms an initial judgement about a product based on its summary rating statistics. This initial belief plays a key role in her next evaluation of the review. To prove their conjecture, the authors of [660] collected the application reviews from Apple Store from July 1^{st} to August 31^{st}, 2013. By analyzing these reviews they show the existence of a confirmation bias, which outlines the tendency of consumers to perceive reviews confirming (resp., disconfirming) their initial beliefs as more (resp., less) helpful.

This tendency is moderated by the consumer confidence in their initial beliefs. This bias also leads to a greater perceived helpfulness of positive reviews when the average product rating is high, and of negative reviews when the average product rating is low.

6.3.2 Main findings of the knowledge extraction process

In the Introduction, we specified that the main novelties concern: *(i)* the definition of the two social network-based models of Yelp; *(ii)* the definition of three Yelp user stereotypes and their characteristics; *(iii)* the construction of the profile of negative influencers in Yelp. We also pointed out that we aim at answering three research questions, namely: *(i)* What about the dynamics leading a Yelp user to publish a negative review? *(ii)* How can the interaction of these dynamics increase the "power" of negative reviews and people making them? *(iii)* Who are the negative influencers in Yelp? In order to obtain these results and answer these questions, we conducted a data analytics campaign that allowed us to formulate six implications.

The first tells that "The star-based review system of Yelp is positively biased. Indeed, almost all users assign a high number of stars to almost all businesses.". It can be explained by taking into account that Yelp's review system is based on a Likert scale, and it is well known that this scale is positively biased [26, 496, 76]. This implication does not provide unexpected information, but still represents an important confirmation about the correctness of our knowledge extraction process.

The second implication tells that "Score-dl-users play a key role in negative reviews. They are very keen on negatively judging the macro-category they mostly attend.". Unlike the first one, it was not expected. Its explanation partially comes from the first implication. Indeed, if it is true that the Likert scale is positively biased, then a user must be particularly motivated to give a negative rating. Moreover, if such an evaluation is given by a double life user, then it means that it is provided by a person potentially balanced in her evaluations (indeed, she gave both positive and negative evaluations in the past). If a person with these characteristics gives a negative review, it is reasonable to assume that she did so because she had "something important to say". In that case, she probably provides some well founded justifications for her dissatisfaction. In order to do this, she must be competent in that macro-category, which explains the last part of the implication.

The third implication tells that "A user has a very high influence on her/his friends when doing negative reviews.". The first part of it represents an expected result, and is easily explained by the homophily principle [435]. The second part was unexpected and can be explained by considering that several studies in related literature show that negative reviews and reviewers are stronger than positive ones.

The fourth implication tells that "Bridges have a much greater power of influence than non-bridges.". It represents a partially expected result if we consider that bridges generally have a high betweenness centrality and, thus, have the ability to convey an idea, sentiment or opinion from one macro-category to another.

The fifth implication tells that "There is a correlation between k-bridges and top central users.". At first glance, it may appear an expected result, but actually this is not the case. In fact, in some contexts, for example in a Social Internetworking System, bridges connecting different social networks are not necessarily power users [103]. Actually, the more the communities involved in a (multi-) network scenario are integrated, the more likely a bridge is also a power user. Based on this reasoning, and considering that Yelp's macro-categories are closely related to each other, because both a user and a business can belong to more macro-categories simultaneously, the result obtained is reasonable and motivated.

Finally, the sixth implication tells that "The main negative influencers in Yelp are score-dl-users who simultaneously are top central users (according to degree and/or eigenvector and/or PageRank centrality measures).". It is certainly unexpected and is one of our major findings. It was obtained by appropriately integrating the previous five implications. For this reason, the justifications underlying it are those that allowed us to explain the implications from which it derives.

6.3.3 Theoretical contributions

Here, we provide several theoretical contributions to the literature on online review systems and eWOM. First of all, it introduces a new multi-dimensional social network-based model of Yelp. This model perfectly fits the category-based structure of this social medium. It represents Yelp as a set of 22 communities, one for each macro-category. At the same time, it models this social medium as a user network $\mathcal{U}$ where each node denotes a user and an arc between two nodes represents a generic relationship between the corresponding users. Our model can be used in several different scenarios, depending on the type of relationship one wants to represent. In our study, we have specialized it to two different types of relationships, namely the friendship between users (i.e., $\mathcal{U}^f$) and the co-review of the same business carried out by different users (i.e., $\mathcal{U}^{cr}$).

The usage of our model, together with a set of experiments performed on a Yelp dataset, allowed us to show that the star-based review mechanism of Yelp is positively biased. This fact implies that a user must have a strong motivation to write a negative review. In turn, this implies that all information about negative reviews and negative influencers in Yelp is extremely valuable.

After that, thanks to our multi-dimensional model, we were able to define different stereotypes of users in Yelp. In particular, we considered three different stereotypes, namely the bridges, the power users and the double-life users. Bridges are users connecting different communities in Yelp. They are crucial for the dissemination of information in this social platform. In fact, we have seen that the influence exerted by bridges is greater than the one exerted by non-bridges. Power users are very active in performing reviews in the categories of their interest. The amount of reviews they carry out makes them extremely important in the identification of potential influencers. Double-life users show different behaviors in the different categories in which they operate. They generally show a particular attention and severity in a category in which they are extremely experienced. This means that they can play a valuable role as influencers in this category.

We have defined our multi-dimensional model and these stereotypes with respect to Yelp. However, our model can be easily generalized to other online review platforms, such as TripAdvisor, as well as to other types of social platforms. In case of online review platforms, the extension of our model is immediate. In fact, it is sufficient to know and report in our model the hierarchy of categories underlying the online review platform. In case of other types of social media, the extension is possible and quite simple. In fact, it is sufficient to specify a (possibly hierarchical) mechanism for dividing users into groups, as well as to identify the types of user relationships of interest. It seems quite obvious that friendship is a relationship of interest for any social platform. On the contrary, co-review does not always make sense and could be replaced by other types of relationships.

As for stereotypes, we observe that those considered here are not the only ones possible for an online review platform. In the future, we plan to identify other stereotypes and study their contribution to the extraction of useful knowledge from Yelp. At the same time, the three identified stereotypes can be directly extended to any other online review platform. The concept of power user can be easily extended to any social platform and any online social network too. The concept of bridge and double-life user can be extended only to those cases where users of a social platform can be organized into communities based on some parameters. In this case, a bridge is a user acting as a link between two communities, while a double-life user is a user having different behaviors in different communities.

The last theoretical contribution concerns the definition of the Negative Reviewer Network. This model plays an extremely important role in the study of negative reviews and, above all, in the identification of negative influencers, who correspond to nodes with high degree centrality and/or high eigenvector centrality, as we have seen in Section 6.2.5. Analogously to what happens for the other theoretical tools,

the extension of this model to other online review platforms is immediate. Instead, its extension to other types of social platforms is much less simple than the other models and concepts seen above. In fact, by its nature, the Negative Reviewer Network is specifically designed to model negative reviews and reviewers. Therefore, its extension is only possible by identifying other negative behaviors that one wants to study and by defining a form of co-participation of multiple users to these behaviors.

6.3.4 Practical implications

Starting from the theoretical background, the hypotheses made and the implications confirming them, we can outline different applications of the knowledge here extracted to real life scenarios. In particular, we can identify two different perspectives, i.e., the business and the user ones.

The business perspective concerns all the possible actions that a company can take to expand its customer base, to improve its brand image or to extend the products/services it offers. In this context, the user identified stereotypes and the implications associated with them can be extremely useful. Let us consider, for example, k-bridges. We have seen the extremely important role that they play in disseminating information between different communities. In the previous sections, we have also seen that past literature highlights the strong impact that negative reviews can have. In this context, a k-bridge making a negative review could have a disruptive effect on a business image.

Therefore, the possibility of detecting k-bridges provided by our approach can become a valuable tool for a business, which can adopt a variety of policies aiming at improving their evaluation of its products/services from negative to neutral or, even, positive. Another extremely important policy in this sense could regard the promotion of a business to k-bridges who do not know it. This could favor the knowledge of this business in all the communities which the k-bridges belong to. In fact, a k-bridge belonging to a community where a business is well known and another community where this latter is unknown could become a promoter of the business from the former community to the latter one.

Another important application that could leverage k-bridges is the expansion of products/services offered by a business towards new categories, or even new macro-categories, of Yelp. One way to increase the chance of designing new products/services being of interest to users could be as follows. A business could identify all the k-bridges belonging to the categories in which it is already known and its products/services are highly appreciated. Then, it could determine the other categories of products/services where the identified k-bridges have performed revisions; in fact, the products/services of these last categories could be of interest for the potential

customers of this business. The greater the number of k-bridges that have shown interest in these categories, the more likely customers belonging to them will be attracted by the business if it expands its offers towards these markets.

A further application of k-bridges, collateral to the one seen above, concerns advertising campaigns. In fact, knowing the most promising communities when proposing new products/services also implies being able to carry out advertising campaigns focusing on them. In this way, the effectiveness and efficiency of the advertisement activity in terms of time and costs are increased.

However, k-bridges are not the only identified stereotype having important practical applications. In fact, both power users and double-life users are equally important. Since the latter two stereotypes appear within the definition of negative influencers, we now see some possible applications of this last concept that subsumes the other two ones. Negative influencers have two important characteristics. The first concerns the high value of network centrality measures (degree centrality and/or eigenvector centrality and/or PageRank), which makes them very influential in the communities where they operate. The second concerns their behavior in carrying out reviews. In fact, we have seen that a negative influencer, being a score-dl-user, tends to give positive reviews in the categories of lesser interest, while she is very demanding and severe in the categories in which she is more experienced and that interest her the most. This also assumes that such a user generally has a recognized leadership exactly in the category in which she is most severe. Therefore, it becomes crucial for a business in that category taking all possible actions to ensure that she takes a neutral, or hopefully a positive, attitude towards the products/services it offers. On the other hand, as we have seen for k-bridges, it is possible to think of targeted advertising and marketing actions on these users that, if successful, are characterized by a high level of efficiency and effectiveness.

So far we have seen the possible exploitations of our knowledge patterns from the business viewpoint. Now, we want to see how the same patterns can have practical implications for the user as well. In particular, we want to consider what benefits a user can get by looking at other relevant users (such as k-bridges, power users, influencers) in Yelp.

A first benefit can be obtained from the examination of the reviews of negative influencers in Yelp. Based on the knowledge we have extracted, we can assume that these users are very experienced in a certain category and very severe in exactly that category. Therefore, if these users have issued positive reviews on the products/services of a business in that category, it is very likely that they are of high quality.

A second benefit for a user concerns the knowledge of the features characterizing the profile of an influencer in Yelp. This knowledge becomes extremely useful if she

wants to become an influencer in that social medium. In fact, based on the derived implications, the user knows that she has a better chance to become an influencer if she becomes a k-bridge. As a consequence, she will have to be active in making revisions in multiple categories. In addition, she should be a power user; therefore, she must have many friendship and co-review relationships (which implies she has a high degree centrality). Alternatively, she can have a limited number of friendship and co-review relationships as long as the users connected to her are, in turn, power users (which implies she has a high eigenvector centrality). Finally, she must identify one or more categories in which she wants to be an influencer and develop a high experience in them in order to give severe, but correct, reviews.

The knowledge here extracted can also be useful to define recommender systems for users who want to discover new products/services. This can be done, for example, by leveraging k-bridges. In fact, assume that a user follows some categories. It is possible to identify all the k-bridges of these categories and, for these k-bridges, to consider the categories followed by them. In this way, it is possible to identify which categories are the most followed by these k-bridges. If one of these categories is not already followed by the user, it is possible to recommend it to her. This very general approach could be further refined by examining the proximity, in the Yelp hierarchy, of candidate categories to those already followed by the user. A further refinement could assign different weights to the different k-bridges, based on the similarity of their past evaluation to those of the user of interest on the same products/services, or based on the number of categories already followed by both them and the user of interest.

6.3.5 Limitations and future research directions

Our theoretical tools (i.e., the multi-dimensional social network-based model of Yelp, the stereotypes and the Negative Review Network), together with the hypotheses formulated and the implications confirming them, have allowed us to shed light on the phenomenon of negative reviews and negative influencers in Yelp. The tools proposed and the approach followed are sufficiently general to be extended directly to other online review platforms and, after some generalizations, to any social platform. However, they are to be considered simply as a first step in this direction, because they are not free from limitations, whose knowledge paves the way to new future research investigations.

The first limitation of our approach is that it is exclusively structural and does not take semantics into account. Actually, a more focused study on the contents of negative reviews would be necessary to understand the reasons that led users to formulate them. This would increase the effectiveness and efficiency of the applications

of our approach discussed in Section 6.3.4. In fact, given a service/product receiving many negative reviews, we could strive to understand the main reasons for this fact and, therefore, make the appropriate improvements aimed at satisfying as many users as possible in the shortest time.

An in-depth semantic analysis of reviews would also be extremely useful to define one or more taxonomies of negative influencers. This would allow us to classify them based not only on the products/services they criticize, as in the present approach, but also on the main reasons for negativity (which would give us several indications on where intervening first or mainly). Semantic knowledge would also allow us to better evaluate negative influencers in order to understand who give plausible reasons and who, instead, are prevented, regardless it happens. As a matter of fact, a business could make an effective and efficient recovery work on the former category of influencers, while it could decide not to intervene on the latter one, because the possibility of making them neutral or positive is low.

Another limitation of our approach, which is, at the same time, a potential future development of our research concerns stereotypes. Here, we have presented three of them, namely the k-bridges, the power users and the double-life users. Their identification was driven by our research needs. However, we believe that several other stereotypes could be defined and that it could be even possible to go so far as to define a real taxonomy of stereotypes for both Yelp and other online (review) platforms. These would become a real toolbox available to decision makers when they need to make decisions regarding the products/services provided by their business (for instance, to determine those ones to be removed from catalogues, new ones to be proposed, existing ones to be modified for making them more in line with user needs and desires, etc.).

A third limitation of our approach, which is also linked to current technological limitations expected to become less impacting in the future, concerns the possibility of studying all these phenomena over time. In fact, our current approach is based on a temporal (albeit wide) photograph of the negative reviews of Yelp. It is not incremental and, if we want to study the evolution of a phenomenon over time, we should take more datasets referring to different times and study them separately. However, this does not allow us to have a continuous monitoring of the phenomenon, in order to capture any changes regarding it (for instance, any change of how some products/services are perceived by users) as soon as possible. The weight of this limitation (and, consequently, the relevance of overcoming it) is smaller in substantially stable socio-economic conditions, because user perceptions of products/services change very slowly over time in this scenario. Instead, it becomes crucial in historical periods characterized by sudden and disruptive phenomena (think, for instance, of

the current COVID-19 pandemic), capable of upsetting all previous mental patterns of people's judgement. In this case, having the possibility of immediately understanding the changed perceptions of users about products/services and/or the appearance of new needs, with the consequent demand for new products/services, can allow a business to gain a huge advantage over its competitors. More importantly, this feature would allow the whole ecosystem of public and private product/service providers to be efficient and effective in responding to people demands.

7

Investigating user behavior in a blockchain during a cryptocurrency speculative bubble

In this chapter, we present a complex network-based approach to investigate user behavior during a cryptocurrency speculative bubble. Our approach is general and can be applied to any past, present and future cryptocurrency speculative bubble. To verify its potential, we apply it to investigate the Ethereum speculative bubble happened in the years 2017 and 2018. We also describe several knowledge patterns about the behavior of specific categories of users that we obtained from this investigation. Finally, we define how our approach can support the construction of an identikit of the speculators who operated during the Ethereum speculative bubble.

The material presented in this chapter was derived from [253].

7.1 Methods

7.1.1 Dataset description

The dataset we used for our analysis is based on the Ethereum blockchain. As stated on the platform official website[1] "Ethereum is a technology that lets you send cryptocurrency to anyone for a small fee. It also powers applications that everyone can use and no one can take down". Ethereum is a programmable blockchain and represents the technological framework behind the cryptocurrency Ether (ETH).

Our dataset was downloaded from Google BigQuery[2]. It contains all the transactions made on Ethereum from January 1^{st}, 2017 to December 31^{st}, 2018. After some data cleaning operations, a row of the dataset, which represents a transaction, contains four columns, namely:

- `from_address`, the blockchain address starting the transaction;
- `to_address`, the blockchain address receiving the transaction;
- `timestamp`, the transaction timestamp;

[1] `https://ethereum.org/en/what-is-ethereum/`

[2] `https://www.kaggle.com/bigquery/ethereum-blockchain`

- `value`, the amount of Weis[3] transferred during the transactions.

The dataset is made of 354,107,563 transactions; the total number of user addresses is 43,537,168. We computed some statistics on it, which are reported in Table 7.1.

Parameter	*Value*
Number of transactions	354,107,563
Total number of `from_addresses`	38,881,752
Total number of `to_addresses`	42,457,991
Cardinality of the intersubsection between `from_addresses` and `to_addresses`	37,802,576
Number of null `from_addresses`	2,104,863
Number of null `to_addresses`	0

Table 7.1: Some preliminary statistics performed on our dataset

7.1.2 Defining the user categories of interest

In this subsection, we present some preliminary analyses "depicting" the pre-bubble, bubble and post-bubble periods, as well as the general behavior of users during the two years covered by our dataset and, especially, during the three periods of our interest. At the end of these analyses, we will be able to define the user categories of interest.

A first analysis concerns the distributions of the number of transactions against `from_addresses` and `to_addresses`. They are reported in Figure 7.1. This figure shows that the two distributions follow a power law. We computed some parameters for them; they are reported in Table 7.2.

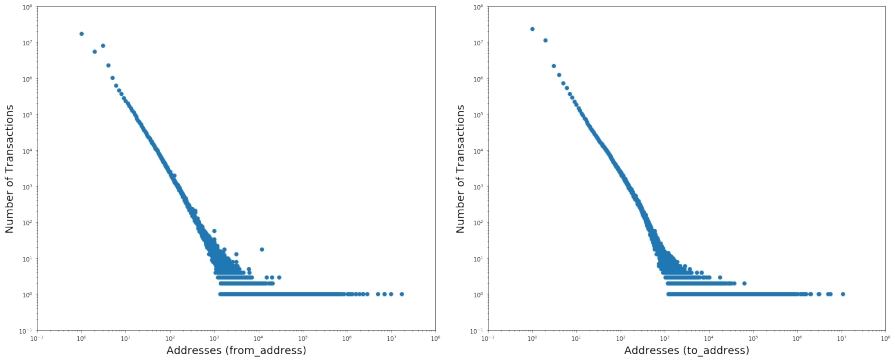


Fig. 7.1: Log-log plots of the distributions of transactions against `from_addresses` (at left) and `to_addresses` (at right)

[3] Wei is the smallest denomination of Ether; it corresponds to 10^{-18} Ethers.

Parameter	`from_addresses`	`to_addresses`
Maximum number of transactions	17,509,218	23,404,261
Average number of transactions	5,640.76	5,913.37
α (power law parameter)	1.477	1.565
δ (power law parameter)	0.013	0.074

Table 7.2: Values of the parameters of transaction distributions against addresses

From the analysis of both Figure 7.1 and Table 7.2 we can observe that the two power law distributions are similar.

The second analysis that we take into consideration concerns the variation of the number of transactions over time. The purpose of this analysis is the identification of the pre-bubble, bubble and post-bubble periods. This trend is shown in Figure 7.2. From the analysis of this figure we can see that from January 2017 to October 2017 there is a substantially linear growth of the number of transactions. From November 2017 to March 2018 there is first an impressive increase and then an impressive decrease of the same variable. Finally, from April 2018 to December 2018 the number of transactions has an irregular trend, but on average its values are lightly higher than the ones observed before November 2017. Based on these observations, in the following, we assume as pre-bubble period the time interval January - October 2017, as bubble period the time interval November 2017 - March 2018, and as post-bubble period the time interval April - December 2018.

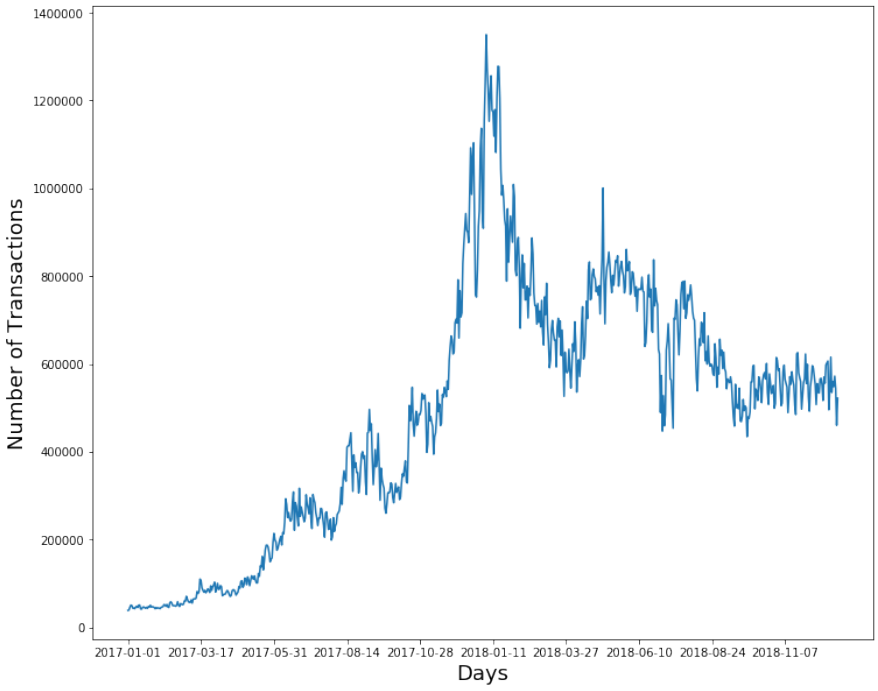


Fig. 7.2: Number of transactions over time

The next analysis focuses on power addresses, i.e., those addresses that have made the most transactions. The analysis of these addresses is extremely relevant for two reasons. First, since the distributions of transactions against addresses follow a power law, the analysis of power addresses covers most of the phenomenon we want to examine. Second, since the number of power addresses is very small, compared to the total number of addresses, it is possible to make very precise and detailed analyses on them, which would be impossible to conduct on all addresses or on a very high fraction of them.

In particular, for each period (pre-bubble, bubble and post-bubble) and for each type of addresses (`from` and `to`), we decided to take the top 1000 addresses as the power ones. For each set thus selected, Table 7.3 shows: *(i)* what percentage of the total number of addresses operating in the reference period the top 1000 addresses correspond to; *(ii)* what percentage of the total number of transactions performed in the reference period the transactions carried out by the top 1000 addresses correspond to. From the analysis of this table, we can deduce that the previous conjectures on the opportunity to carry out the power address analyses were correct.

Set	*Percentage of addresses*	*Percentage of transactions*
Pre-bubble, top 1000 `from_addresses`	0.01549%	89.81%
Bubble, top 1000 `from_addresses`	0.00599%	78.48%
Post-bubble, top 1000 `from_addresses`	0.00534%	77.87%
Pre-bubble, top 1000 `to_addresses`	0.01325%	86.02%
Bubble, top 1000 `to_addresses`	0.00495%	82.29%
Post-bubble, top 1000 `to_addresses`	0.00548%	86.34%

Table 7.3: Percentage of the addresses and transactions covered by each set of power addresses

A first analysis of power addresses concerned the possible overlap between `from_addresses` and `to_addresses`. For this purpose, for each period, we computed the intersubsection between the top 1000 `from_addresses` and the top 1000 `to_addresses`. The result obtained is reported in Table 7.4. This table shows that only a small fraction of power addresses is simultaneously present in the top 1000 `from_addresses` and in the top 1000 `to_addresses`. Another information emerging from Table 7.4 is that this fraction significantly decreases in the transition from pre-bubble to bubble and from bubble to post-bubble periods.

A further analysis on power addresses led us to compute the possible intersubsections of the top 1000 addresses during the pre-bubble, bubble and post-bubble periods. The results obtained are reported in Table 7.5. Here, T^F_{Pre} (resp., T^F_B, T^F_{Post}) is the set of the top 1000 `from_addresses` during the pre-bubble (resp., bubble,

Pre-bubble	*Bubble*	*Post-Bubble*
173	115	81

Table 7.4: Number of power addresses simultaneously belonging to the set of the top 1000 `from_addresses` and to the set of the top 1000 `to_addresses` in the three periods of interest

post-bubble) period. Analogously, T^T_{Pre}, T^T_B and T^T_{Post} are the corresponding sets for `to_addresses`. From the analysis of this table we can see that:

Set	*Cardinality*
$\lvert T^F_{Pre} \cap T^F_B \rvert$	267
$\lvert T^F_B \cap T^F_{Post} \rvert$	268
$\lvert T^F_{Pre} \cap T^F_{Post} \rvert$	107
$\lvert T^T Pre \cap T^T_B \rvert$	288
$\lvert T^T_B \cap T^T_{Post} \rvert$	309
$\lvert T^T_{Pre} \cap T^T_{Post} \rvert$	114
$\lvert T^F_{Pre} \cap T^F_B \cap T^F_{Post} \rvert$	102
$\lvert T^T_{Pre} \cap T^T_B \cap T^T_{Post} \rvert$	112

Table 7.5: Cardinalities of the possible intersubsections of the top 1000 addresses during the pre-bubble, bubble and post-bubble periods

- The trends of `from_addresses` and `to_addresses` are very similar.
- The bubble has changed the power address scenario considerably. In fact, while the cardinality of the sets $\lvert T^F_{Pre} \cap T^F_B \rvert$, $\lvert T^F_B \cap T^F_{Post} \rvert$, $\lvert T^T Pre \cap T^T_B \rvert$ and $\lvert T^T_B \cap T^T_{Post} \rvert$ is quite large, the one of the sets $\lvert T^F_{Pre} \cap T^F_{Post} \rvert$ and $\lvert T^T_{Pre} \cap T^T_{Post} \rvert$ is much smaller. This tells us that, during the bubble period, most of the power addresses present in the pre-bubble period disappeared and new power addresses appeared; these last continued to exist during the post-bubble period. Finally, we observe that there are some power addresses, which we call "Survivors", that are present in the pre-bubble, bubble and post-bubble periods.

Based on the intersubsections introduced in Table 7.5, we can define three categories of addresses whose analysis appears extremely interesting for the extraction of knowledge on the bubble of Ethereum (and, presumably, of other cryptocurrencies). These categories are:

- *the Survivors*, which are the power addresses present in the pre-bubble, bubble and post-bubble periods;
- *the Missings*, which are the power addresses present in the pre-bubble period, but absent in the bubble and post-bubble ones;

- *the Entrants*, which are the power addresses absent in the pre-bubble period, but present in the bubble and post-bubble ones.

In the following, we aim at extracting knowledge patterns about these categories of addresses (and, ultimately, of users).

The next analysis aims at identifying how many power addresses are present in each category. We conducted this analysis for `from_addresses`, `to_addresses` and the intersubsection of these two sets. The results obtained are shown in Table 7.6.

Addresses	*Survivors*	*Entrants*	*Missings*
`from_addresses`	102	166	728
`to_addresses`	112	197	710
Intersubsection of `from_addresses` and `to_addresses`	21	17	114

Table 7.6: Number of power addresses belonging to the Survivors, Entrants and Missings categories

To fully understand the knowledge that can be extracted from this table, we must recall that: *(i)* the maximum number of power addresses for each category is equal to 1000; *(ii)* the Survivors, the Entrants and the Missings are obtained carrying out intersubsection operations. According to this reasoning, we can observe that the Survivors are very few; this result was expected because this category of addresses is obtained performing the intersubsection of three sets. The Entrants are also few while the Missings are many. This confirms that the bubble completely revolutionized the power address scenario in Ethereum, making the previous "main actors" (i.e., power addresses) disappear while introducing new ones.

Observe that, for all categories, the intersubsections between `from_addresses` and `to_addresses` are very small. This is totally in line with Table 7.4, where we have seen that only a few addresses are `from_addresses` and `to_addresses` simultaneously.

7.1.3 Detecting the main features of the user categories of interest

Given a period (pre-bubble, bubble and post-bubble) and the set of the corresponding power addresses, we build a support social network. More specifically, let

$$\mathcal{N}_{Pre} = \langle NS_{Pre}, AS_{Pre} \rangle \qquad \mathcal{N}_{B} = \langle NS_{B}, AS_{B} \rangle \qquad \mathcal{N}_{Post} = \langle NS_{Post}, AS_{Post} \rangle$$

be the social networks associated with the pre-bubble, bubble and post-bubble periods.

NS_{Pre} (resp., NS_B, NS_{Post}) represents the set of the nodes of $\mathcal{N}_{Pre}$ (resp., $\mathcal{N}_B$, $\mathcal{N}_{Post}$). In this set, there is a node n_i for each power address. A label is associated

with n_i; it allows us to specify if the corresponding address belongs to one of the categories of interest (Survivors, Entrants, Missings) or to none of them. Since there is a biunivocal correspondence between power addresses and nodes, in the following we will use these two terms interchangeably.

AS_{Pre} (resp., AS_B, AS_{Post}) denotes the set of the arcs of $\mathcal{N}_{Pre}$ (resp., $\mathcal{N}_B$, $\mathcal{N}_{Post}$). There is an arc $(n_i, n_j, TS_{ij}) \in AS_{Pre}$ (resp., AS_B, AS_{Post}) if there was at least one transaction from n_i to n_j. TS_{ij} represents the set of transactions from n_i to n_j made during the pre-bubble (resp., bubble, post-bubble) period. It consists of a set of pairs (t_{ij_k}, τ_{ij_k}), where t_{ij_k} represents the k^{th} transaction and τ_{ij_k} indicates the corresponding timestamp.

Having defined the support social networks, we can start our analyses on the address categories of interest. Below, we use the following notations:

- $\mathcal{S}^F$ (resp., $\mathcal{S}^T$), to indicate the Survivors `from_addresses` (resp., `to_addresses`).
- $\mathcal{E}^F$ (resp., $\mathcal{E}^T$), to denote the Entrants `from_addresses` (resp., `to_addresses`).
- $\mathcal{M}^F$ (resp., $\mathcal{M}^T$), to represent the Missings `from_addresses` (resp., `to_addresses`).

In order to conduct our analyses on the address categories, we have considered the adoption of ego networks extremely useful. We recall that the ego network of a node n_i (called, precisely, "ego") consists of n_i, the nodes (called "alters") to which n_i is directly connected, the arcs connecting the ego to the alters and the arcs connecting the alters to each other. An ego network provides a clear indication of the relationships the corresponding ego is involved in, the nodes it interacts with, and the relationships existing between these last ones. In our analysis, which aims at detecting the features of each address category, ego network can play an important role because, due to the principle of homophily characterizing social networks [435], the features of a node are strongly influenced by the nodes belonging to its neighborhood.

As a first task, we computed the average number of nodes, the average number of arcs and the average density of the ego networks of the nodes belonging to each address category of interest. First, we examined the pre-bubble period. The results obtained are reported in Table 7.7.

From the analysis of this table we can see that the ego networks of the Survivors nodes have an average number of nodes and arcs significantly higher than the ego networks of the nodes belonging to the other two categories. If such a result was expected for the Entrants (because the corresponding nodes were not power addresses during the pre-bubble period), it is instead surprising for the Missings. In fact, the latter, like the Survivors, were power addresses during the pre-bubble period. This allows us to conclude that having a very large ego-network during the pre-bubble

Parameter	$\mathcal{S}^F$	$\mathcal{S}^T$	$\mathcal{M}^F$	$\mathcal{M}^T$	$\mathcal{E}^F$	$\mathcal{E}^T$
Average number of nodes	36,177.84	27,335.21	1,710.52	2,864.44	537.69	886.02
Average number of arcs	115,290.27	68,051.82	4,561.86	7,342.89	795.53	1,718.39
Average density	0.1120	0.0639	0.3852	0.2423	0.2125	0.1568

Table 7.7: Average number of nodes, average number of arcs and average density of the ego networks of the nodes belonging to the address categories of interest - Pre-bubble period

period increases the possibility of remaining power addresses during the bubble and post-bubble periods. As far as density is concerned, there are no particular observations to make taking into account that the low density of Survivor's ego networks can be explained simply by the large number of nodes characterizing them.

After this, we examined the bubble period. The results obtained are reported in Table 7.8.

Parameter	$\mathcal{S}^F$	$\mathcal{S}^T$	$\mathcal{M}^F$	$\mathcal{M}^T$	$\mathcal{E}^F$	$\mathcal{E}^T$
Average number of nodes	82,832.51	59,339.83	366.58	798.29	17,180.69	18,945.69
Average number of arcs	325,179.44	172,713.37	587.84	2563.00	59,733.11	67,956.61
Average density	0.074	0.019	0.401	0.282	0.211	0.031

Table 7.8: Average number of nodes, average number of arcs and average density of the ego networks of the nodes belonging to the address categories of interest - Bubble period

From the analysis of this table we can observe that both the Survivors and the Entrants have much larger ego networks than the Missings. Actually, this result was expected since, in the bubble period, the nodes belonging to the Survivors and the Entrants are power addresses. Instead, it is unexpected that the Survivors have much larger ego networks than the Entrants. In fact, the addresses of both categories are power addresses during the bubble period. However, it seems that the Survivors tend to include the strongest power addresses. Note also that the one of the Survivors' ego networks during the bubble period is about twice the size of the Survivors' ego networks during the pre-bubble period. Also, the Survivors' ego networks have by far the largest size during the bubble period. This allows us to conclude that it is exactly the activity of the Survivors that could have caused the bubble; this activity has led to the exit of the Missings from the power addresses and to the arrival of the Entrants among them. However, these last ones enter into the power addresses "on tiptoe"; in fact, they are not the ones who dictate the line and cause the bubble; this task is carried out by the Survivors.

Finally, we considered the post-bubble period. The results obtained are reported in Table 7.9.

Parameter	$\mathcal{S}^F$	$\mathcal{S}^T$	$\mathcal{M}^F$	$\mathcal{M}^T$	$\mathcal{E}^F$	$\mathcal{E}^T$
Average number of nodes	47,237.20	46,661.02	162.10	572.93	19,686.75	22,373.64
Average number of arcs	174,537.78	148,359.25	425.70	1,360.52	93,099.84	70,518.77
Average density	0.1045	0.039	0.411	0.233	0.178	0.0157

Table 7.9: Average number of nodes, average number of arcs and average density of the ego networks of the nodes belonging to the address categories of interest - Post-bubble period

The analysis of this table confirms the trends we observed in Table 7.8 for the bubble period. This is not surprising because also during the post-bubble period both the Survivors and the Entrants are power addresses. Note that, during this period, the size of the Survivors' ego networks is much smaller than the one of the Survivors' ego networks during the bubble period, although it is slightly larger than the size of the Survivors' ego networks during the pre-bubble period. This trend perfectly reflects the one of the number of transactions reported in Figure 7.2. This is a further confirmation that the trend shown by Ethereum in the years 2017 and 2018, which led to a bubble, was mainly caused by the Survivors. We note that the size of the Entrants' ego network during the post-bubble period shows a slight growth compared to the bubble period. This is an indication that, during the post-bubble period, the Entrants consolidate their presence among the power addresses, even though they are not dictating the line yet: this is still a responsibility of the Survivors.

The analysis of Tables 7.7 - 7.9, along with the previous reasoning, indicates that having very large ego networks seems to be an intrinsic feature of the Survivors, regardless of the pre-bubble, bubble or post-bubble period.

7.1.4 Generalizability of the proposed analyses

In subsection 7.1.1, we saw that our dataset was derived from Ethereum. Furthermore, we saw that each record in it corresponds to a transaction and stores only four fields related to it, namely: *(i)* the blockchain address starting it; *(ii)* the blockchain address receiving it; *(iii)* its timestamp; *(iv)* the amount of money transferred during it. These four fields are very general and available for any cryptocurrency blockchain. Therefore, although our analysis was performed on Ethereum, our approach can be extended to any cryptocurrency blockchain. To facilitate this extension, in the following we abstract the analyses described here into a well-structured

algorithm of which they represent single steps. The pseudo-code of this algorithm is shown in Algorithms 3 and 4.

Input

- $\mathcal{B}$: the cryptocurrency blockchain of interest
- I: the time interval to investigate

Output

- PA^F_{Pre}, PA^F_B, PA^F_{Post}, PA^T_{Pre}, PA^T_B, PA^T_{Post}: power addresses of the dataset
- SPA_{Pre}, SPA_B, SPA_{Post}, PA^F_{Pre-B}, PA^F_{B-Post}, PA^T_{Pre-B}, PA^T_{B-Post}: power addresses of the dataset
- $\mathcal{S}^F$, $\mathcal{S}^T$: the Survivors; $\mathcal{M}^F$, $\mathcal{M}^T$: the Missings; $\mathcal{E}^F$, $\mathcal{E}^T$: the Entrants
- $EgoKPSet$: a set of knowledge patterns derived from the ego network analyses
- $BackboneKPSet$: a set of knowledge patterns on the possible presence of backbones
- $BSurvivorsSet$: a set of potential Survivors
- $PBSurvivorsSet$: a set of potential Survivors
- $PBEntrantsSet$: a set of potential Entrants

Require:

- D: a dataset of transactions;
- I_{Pre}, I_B, I_{Post}: time intervals;
- $\mathcal{N}_{Pre}$, $\mathcal{N}_B$, $\mathcal{N}_{Post}$: social networks;
- $ENSet^{\mathcal{S},F}_{Pre}$, $ENSet^{\mathcal{S},T}_{Pre}$, $ENSet^{\mathcal{M},F}_{Pre}$, $ENSet^{\mathcal{M},T}_{Pre}$, $ENSet^{\mathcal{E},F}_{Pre}$, $ENSet^{\mathcal{E},T}_{Pre}$: a set of ego networks;
- $ENSet^{\mathcal{S},F}_{B}$, $ENSet^{\mathcal{S},T}_{B}$, $ENSet^{\mathcal{M},F}_{B}$, $ENSet^{\mathcal{M},T}_{B}$, $ENSet^{\mathcal{E},F}_{B}$, $ENSet^{\mathcal{E},T}_{B}$: a set of ego networks;
- $ENSet^{\mathcal{S},F}_{Post}$, $ENSet^{\mathcal{S},T}_{Post}$, $ENSet^{\mathcal{M},F}_{Post}$, $ENSet^{\mathcal{M},T}_{Post}$, $ENSet^{\mathcal{E},F}_{Post}$, $ENSet^{\mathcal{E},T}_{Post}$: a set of ego networks;
- T^F_{Pre}, T^T_{Pre}, T^F_B, T^T_B, T^F_{Post}, T^T_{Post}: top power addresses of the dataset;

D = Extract_Dataset($\mathcal{B}$,I)
$\langle I_{Pre}, I_B, I_{Post}\rangle$ = Determine_Intervals(D)
$\langle PA^F_{Pre}, PA^F_B, PA^F_{Post}\rangle$ = Detect_From_Power_Addresses($I_{Pre}, I_B, I_{Post}, D$)
$\langle PA^T_{Pre}, PA^T_B, PA^T_{Post}\rangle$ = Detect_To_Power_Addresses($I_{Pre}, I_B, I_{Post}, D$)
$\langle SPA_{Pre}, SPA_B, SPA_{Post}\rangle$ = Detect_Super_Power_Addresses($PA^F_{Pre}, PA^F_B, PA^F_{Post}, PA^T_{Pre}, PA^T_B, PA^T_{Post}$)
$\langle SPA^F_{Pre-B}, SPA^F_{B-Post}\rangle$ = Detect_Multi_Interval_From_Power_Addresses($PA^F_{Pre}, PA^F_B, PA^F_{Post}$)
$\langle SPA^T_{Pre-B}, SPA^T_{B-Post}\rangle$ = Detect_Multi_Interval_To_Power_Addresses($PA^T_{Pre}, PA^T_B, PA^T_{Post}$)
$\langle \mathcal{S}^F, \mathcal{S}^T\rangle$ = Detect_Survivors($PA^F_{Pre}, PA^F_B, PA^F_{Post}, PA^T_{Pre}, PA^T_B, PA^T_{Post}$)
$\langle \mathcal{M}^F, \mathcal{M}^T\rangle$ = Detect_Missings($PA^F_{Pre}, PA^F_B, PA^F_{Post}, PA^T_{Pre}, PA^T_B, PA^T_{Post}$)
$\langle \mathcal{E}^F, \mathcal{E}^T\rangle$ = Detect_Entrants($PA^F_{Pre}, PA^F_B, PA^F_{Post}, PA^T_{Pre}, PA^T_B, PA^T_{Post}$)
$\langle \mathcal{N}_{Pre}, \mathcal{N}_B, \mathcal{N}_{Post}\rangle$ = Construct_Social_Networks($I_{Pre}, I_B, I_{Post}, D$)
$\langle ENSet^{\mathcal{S},F}_{Pre}, ENSet^{\mathcal{S},T}_{Pre}, \rangle$ = Construct_Survivors_Ego_Networks_Pre($I_{Pre}, \mathcal{N}_{Pre}, \mathcal{S}^F, \mathcal{S}^T$)
$\langle ENSet^{\mathcal{S},F}_{B}, ENSet^{\mathcal{S},T}_{B}, \rangle$ = Construct_Survivors_Ego_Networks_Bubble($I_B, \mathcal{N}_B, \mathcal{S}^F, \mathcal{S}^T$)
$\langle ENSet^{\mathcal{S},F}_{Post}, ENSet^{\mathcal{S},T}_{Post}, \rangle$ = Construct_Survivors_Ego_Networks_Post($I_{Post}, \mathcal{N}_{Post}, \mathcal{S}^F, \mathcal{S}^T$)

Algorithm 3: Investigating user behavior during a cryptocurrency speculative bubble (first part)

Our algorithm receives the cryptocurrency blockchain $\mathcal{B}$ of interest and the time interval I during which there was a speculative bubble involving $\mathcal{B}$.

It first calls the function *Extract_Dataset* that returns the dataset D of the transactions of $\mathcal{B}$ during I. Next, it calls the function *Determine_Intervals* to partition I into three sub-intervals I_{Pre}, I_B and I_{Post}, relating to the pre-bubble, bubble and post-bubble periods, respectively. After that, it calls the functions

Require:

$\langle ENSet^{\mathcal{M},F}_{Pre}, ENSet^{\mathcal{M},T}_{Pre}, \rangle$ = Construct_Missings_Ego_Networks_Pre($I_{Pre}, \mathcal{N}_{Pre}, \mathcal{M}^F, \mathcal{M}^T$)
$\langle ENSet^{\mathcal{M},F}_{B}, ENSet^{\mathcal{M},T}_{B}, \rangle$ = Construct_Missings_Ego_Networks_Bubble($I_B, \mathcal{N}_B, \mathcal{M}^F, \mathcal{M}^T$)
$\langle ENSet^{\mathcal{M},F}_{Post}, ENSet^{\mathcal{M},T}_{Post}, \rangle$ = Construct_Missings_Ego_Networks_Post($I_{Post}, \mathcal{N}_{Post}, \mathcal{M}^F, \mathcal{M}^T$)
$\langle ENSet^{\mathcal{E},F}_{Pre}, ENSet^{\mathcal{E},T}_{Pre}, \rangle$ = Construct_Entrants_Ego_Networks_Pre($I_{Pre}, \mathcal{N}_{Pre}, \mathcal{E}^F, \mathcal{E}^T$)
$\langle ENSet^{\mathcal{E},F}_{B}, ENSet^{\mathcal{E},T}_{B}, \rangle$ = Construct_Entrants_Ego_Networks_Bubble($I_B, \mathcal{N}_B, \mathcal{E}^F, \mathcal{E}^T$)
$\langle ENSet^{\mathcal{E},F}_{Post}, ENSet^{\mathcal{E},T}_{Post}, \rangle$ = Construct_Entrants_Ego_Networks_Post($I_{Post}, \mathcal{N}_{Post}, \mathcal{E}^F, \mathcal{E}^T$)
$EgoKPSet$ = Analyze_Ego_Pre($ENSet^{\mathcal{S},F}_{Pre}, ENSet^{\mathcal{S},T}_{Pre}, ENSet^{\mathcal{M},F}_{Pre}, ENSet^{\mathcal{M},T}_{Pre}, ENSet^{\mathcal{E},F}_{Pre}, ENSet^{\mathcal{E},T}_{Pre}$)
$EgoKPSet = EgoKPset \cup$ Analyze_Ego_Bubble($ENSet^{\mathcal{S},F}_{B}, ENSet^{\mathcal{S},T}_{B}, ENSet^{\mathcal{M},F}_{B}, ENSet^{\mathcal{M},T}_{B}, ENSet^{\mathcal{E},F}_{B}, ENSet^{\mathcal{E},T}_{B}$)
$EgoKPSet = EgoKPset \cup$ Analyze_Ego_Post($ENSet^{\mathcal{S},F}_{Post}, ENSet^{\mathcal{S},T}_{Post}, ENSet^{\mathcal{M},F}_{Post}, ENSet^{\mathcal{M},T}_{Post}, ENSet^{\mathcal{E},F}_{Post}, ENSet^{\mathcal{E},T}_{Post}$)
$BackboneKPSet$ = Detect_Backbones_Survivor_Pre($ENSet^{\mathcal{S},F}_{Pre}, ENSet^{\mathcal{S},T}_{Pre}, \mathcal{S}^F, \mathcal{S}^T, \mathcal{M}^F, \mathcal{M}^T, \mathcal{E}^F, \mathcal{E}^T$)
$BackboneKPSet = BackboneKPSet \cup$ Detect_Backbones_Survivor_Bubble($ENSet^{\mathcal{S},F}_{B}, ENSet^{\mathcal{S},T}_{B}, \mathcal{S}^F, \mathcal{S}^T, \mathcal{M}^F, \mathcal{M}^T, \mathcal{E}^F, \mathcal{E}^T$)
$BackboneKPSet = BackboneKPSet \cup$ Detect_Backbones_Survivor_Post($ENSet^{\mathcal{S},F}_{Post}, ENSet^{\mathcal{S},T}_{Post}, \mathcal{S}^F, \mathcal{S}^T, \mathcal{M}^F, \mathcal{M}^T, \mathcal{E}^F, \mathcal{E}^T$)
$BackboneKPSet = BackboneKPSet \cup$ Detect_Backbones_Missing_Pre($ENSet^{\mathcal{M},F}_{Pre}, ENSet^{\mathcal{M},T}_{Pre}, \mathcal{S}^F, \mathcal{S}^T, \mathcal{M}^F, \mathcal{M}^T, \mathcal{E}^F, \mathcal{E}^T$)
$BackboneKPSet = BackboneKPSet \cup$ Detect_Backbones_Missing_Bubble($ENSet^{\mathcal{M},F}_{B}, ENSet^{\mathcal{M},T}_{B}, \mathcal{S}^F, \mathcal{S}^T, \mathcal{M}^F, \mathcal{M}^T, \mathcal{E}^F, \mathcal{E}^T$)
$BackboneKPSet = BackboneKPSet \cup$ Detect_Backbones_Missing_Post($ENSet^{\mathcal{M},F}_{Post}, ENSet^{\mathcal{M},T}_{Post}, \mathcal{S}^F, \mathcal{S}^T, \mathcal{M}^F, \mathcal{M}^T, \mathcal{E}^F, \mathcal{E}^T$)
$BackboneKPSet = BackboneKPSet \cup$ Detect_Backbones_Entrants_Pre($ENSet^{\mathcal{E},F}_{Pre}, ENSet^{\mathcal{E},T}_{Pre}, \mathcal{S}^F, \mathcal{S}^T, \mathcal{M}^F, \mathcal{M}^T, \mathcal{E}^F, \mathcal{E}^T$)
$BackboneKPSet = BackboneKPSet \cup$ Detect_Backbones_Entrants_Bubble($ENSet^{\mathcal{E},F}_{B}, ENSet^{\mathcal{E},T}_{B}, \mathcal{S}^F, \mathcal{S}^T, \mathcal{M}^F, \mathcal{M}^T, \mathcal{E}^F, \mathcal{E}^T$)
$BackboneKPSet = BackboneKPSet \cup$ Detect_Backbones_Entrants_Post($ENSet^{\mathcal{E},F}_{Post}, ENSet^{\mathcal{E},T}_{Post}, \mathcal{S}^F, \mathcal{S}^T, \mathcal{M}^F, \mathcal{M}^T, \mathcal{E}^F, \mathcal{E}^T$)
$\langle T^F_{Pre}, T^T_{Pre}, T^F_B, T^T_B, T^F_{Post}, T^T_{Post}, \rangle$ = Detect_Top_Power_Addresses($I_{Pre}, I_B, I_{Post}, D$)
$BSurvivorsSet$ = Predict_Bubble_Survivors($T^F_{Pre}, T^T_{Pre}, T^F_B, T^T_B, \mathcal{S}^F, \mathcal{S}^T, \mathcal{M}^F, \mathcal{M}^T, \mathcal{E}^F, \mathcal{E}^T, I_{Pre}, I_B, D$)
$PBSurvivorsSet$ = Predict_Post_Survivors($T^F_B, T^T_B, T^F_{Post}, T^T_{Post}, \mathcal{S}^F, \mathcal{S}^T, \mathcal{M}^F, \mathcal{M}^T, \mathcal{E}^F, \mathcal{E}^T, I_B, I_{Post}, D$)
$PBEntrantsSet$ = Predict_Post_Entrants($T^F_B, T^T_B, T^F_{Post}, T^T_{Post}, \mathcal{S}^F, \mathcal{S}^T, \mathcal{M}^F, \mathcal{M}^T, \mathcal{E}^F, \mathcal{E}^T, I_B, I_{Post}, D$)

return all outputs

Algorithm 4: Investigating user behavior during a cryptocurrency speculative bubble (second part)

Detect_From_Power_Addresses, *Detect_To_Power_Addresses* and *Detect_Super_Power_Addresses* to determine the power addresses with the largest number of incoming arcs, outgoing arcs and both. Finally, it calls the functions *Detect_Multi_Interval_From_Power_Addresses* and *Detect_Multi_Interval_To_Power_Addresses* to determine the addresses that remain From_Power_Addresses and To_Power_Addresses when passing from the pre-bubble period to the bubble one and from the bubble period to the post-bubble one.

At this point, our algorithm has all the data it needs to activate *Detect_Survivors*, *Detect_Missings* and *Detect_Entrants*, which aim at determining the Survivors $\mathcal{S}^F$ and $\mathcal{S}^T$, the Missings $\mathcal{M}^F$ and $\mathcal{M}^T$ and the Entrants $\mathcal{E}^F$ and $\mathcal{E}^T$. Next, it calls the function *Construct_Social_Network* that returns the social networks $\mathcal{N}_{Pre}$, $\mathcal{N}_B$ and $\mathcal{N}_{Post}$ relative to the pre-bubble, bubble and post-bubble period. After that, it calls the functions *Construct_Survivors_Ego_Network_Pre*, *Construct_Survivors_Ego_Network_Bubble* and *Construct_Survivors_Ego_Network_Post* to construct the ego networks of the Survivors of the social networks $\mathcal{N}_{Pre}$, $\mathcal{N}_B$ and $\mathcal{N}_{Post}$. Similarly, it proceeds to call the suitable functions for constructing the ego networks of the Missings and the Entrants for the same social networks mentioned above.

The ego networks thus constructed represent the basis for the next analyses aimed at extracting a set *EgoKPSet* of knowledge patterns on the characteristics of the Survivors, the Missings and the Entrants in the pre-bubble, bubble and post-bubble periods. Our algorithm performs this extraction by calling the functions *Analyze_Ego_Pre*, *Analyze_Ego_Bubble* and *Analyze_Ego_Post*. The next analysis performed by it concerns the possible existence of backbones linking Survivors, Missings or Entrants in the pre-bubble, bubble and post-bubble periods. To this end, it calls some functions having the objective of extracting the set BackboneKPSet of knowledge patterns concerning the possible existence of backbones among the various kinds of address of interest.

Once the backbone analysis is finished, our algorithm proceeds with the last analysis which, unlike the previous ones, is predictive. In fact, it aims at predicting, during a certain period, the nodes that will become protagonists in the next period. To this end, it calls the functions *Predict_Bubble_Survivors*, *Predict_Post_Survivors* and *Predict_Post_Entrants*. The first examines nodes during the pre-bubble period and predicts which of them constitute the set $BSurvivorsSet$ of potential Survivors during the bubble period. The second and the third examine the nodes during the bubble period and predict which of them will form the set $PBSurvivorsSet$ and $PBEntrantsSet$ of potential Survivors and Entrants during the post-bubble period.

The algorithm terminates returning in output all the information extracted through the calls of the functions mentioned above.

A more abstract and simplified graphical representation of it is shown in Figure 7.3.

7.2 Results

In this subsection, we provide some considerations regarding the proposed analyses, the results obtained and their applicability for future cryptocurrency speculative bubbles. In particular, we aim at answering the following questions:

- Are there backbones linking users of a certain category? Can we apply the concept of ego networks and k-cores to detect them?
- The graphical evaluation of the existence of a backbone should have been based on the concept of clique. However, due to computational complexity issues, our experiments were performed on k-cores, which represent a relaxation of the clique concept. Could the results obtained have been affected by the adoption of k-cores instead of cliques?

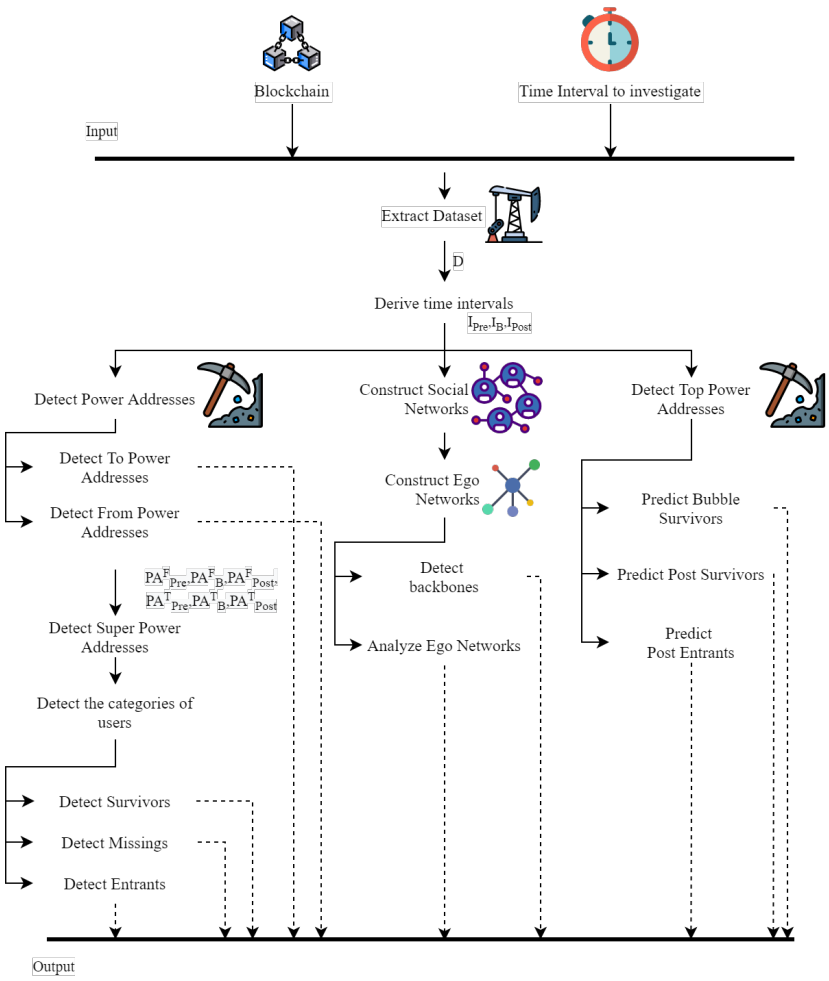


Fig. 7.3: A graphical abstract representation of our algorithm

- Do the described outcomes allow us to infer that there was a group of speculators who managed the Ethereum bubble in the years 2017-2018? If so, what can be said about their profile?
- Can we predict the characteristics of the main future users for the next periods?

In the following, we devote a subsubsection to each of the four issues mentioned above.

7.2.1 Evaluating the existence of backbones linking users of a certain category

The ego networks introduced previously represent a considerable tool to also estimate the possible existence of backbones linking addresses of the same category. In fact, a way to do this consists in verifying, given an address category, the fraction of the corresponding ego networks having, among the alters, at least k addresses belonging to it. Clearly, the higher the value of k and the fraction of the ego networks

satisfying this property, the stronger the hypothesis that a backbone exists among the addresses of the category into examination.

To better clarify this idea, let us consider Table 7.10 that refers to the Survivors' ego networks during the pre-bubble period. In the left part of this table, we examine the set $\mathcal{S}^F$ of the Survivors `from_addresses`. The fifth row of this table tells us that 19.6% of the ego networks of the nodes of $\mathcal{S}^F$ contains at least 5 nodes of $\mathcal{S}^F$ among the alters. This percentage decreases to 0.9% if we consider the presence of at least 5 nodes of $\mathcal{E}^F$ and increases to 33.3% if we take into account the presence of at least 5 nodes of $\mathcal{M}^F$.

	Ego networks of $\mathcal{S}^F$			*Ego networks of* $\mathcal{S}^T$		
	Nodes of $\mathcal{S}^F$	*Nodes of* $\mathcal{E}^F$	*Nodes of* $\mathcal{M}^F$	*Nodes of* $\mathcal{S}^T$	*Nodes of* $\mathcal{E}^T$	*Nodes of* $\mathcal{M}^T$
$k = 1$	0.755	0.088	0.676	0.580	0.223	0.696
$k = 2$	0.512	0.058	0.529	0.339	0.071	0.509
$k = 3$	0.392	0.049	0.402	0.169	0.0	0.348
$k = 4$	0.294	0.019	0.353	0.098	0.0	0.304
$k = 5$	0.196	0.009	0.333	0.080	0.0	0.277
$k = 6$	0.147	0.0	0.284	0.062	0.0	0.268
$k = 7$	0.118	0.0	0.265	0.053	0.0	0.241
$k = 8$	0.078	0.0	0.235	0.036	0.0	0.196
$k = 9$	0.078	0.0	0.216	0.027	0.0	0.196

Table 7.10: Analysis of the presence of backbones linking the Survivors during the pre-bubble period

Once we have clarified the kind of information we want to look for, let us consider Table 7.10, which concerns the Survivors' ego networks during the pre-bubble period. From the analysis of this table we can see that many of the ego-networks of $\mathcal{S}^F$ (resp., $\mathcal{S}^T$) have, among their alters, several nodes belonging to $\mathcal{S}^F$ (resp., $\mathcal{S}^T$), along with several nodes belonging to $\mathcal{M}^F$ (resp., $\mathcal{M}^T$). Instead, the number of ego networks of $\mathcal{S}^F$ (resp., $\mathcal{S}^T$) having one or more nodes of $\mathcal{E}^F$ (resp., $\mathcal{E}^T$) among the alters is very small. This allows us to assume that there is a backbone linking the nodes of $\mathcal{S}^F$ (resp., $\mathcal{S}^T$). The presence of many nodes of $\mathcal{M}^F$ (resp., $\mathcal{M}^T$) among the alters of the ego networks of $\mathcal{S}^F$ (resp., $\mathcal{S}^T$) is not surprising because, during the pre-bubble period, the nodes of $\mathcal{M}^F$ (resp., $\mathcal{M}^T$) were power addresses. Finally, we observe that the presence of Survivors and Missings nodes among the alters of the ego networks of Survivors nodes is more marked for `from_addresses` than for `to_addresses`, as we can see comparing the first three and the last three columns of Table 7.10.

Consider, now, Table 7.11 that refers to the Missings' ego networks during the pre-bubble period. The structure and the semantics of this table are analogous to the ones of Table 7.10. From the analysis of this table, we can observe that many ego

networks of $\mathcal{M}^F$ (resp., $\mathcal{M}^T$) have one or two nodes of $\mathcal{M}^F$ (resp., $\mathcal{M}^T$) or of $\mathcal{S}^F$ (resp., $\mathcal{S}^T$) among their alters. However, compared to the case of the Survivors, reported in Table 7.10, this phenomenon is much smaller both as fraction of ego-networks and as value of k. Therefore, we can conclude that there is a backbone also among the nodes of $\mathcal{M}^F$ (resp., $\mathcal{M}^T$), although this is less strong than the one observed for the nodes of $\mathcal{S}^F$ (resp., $\mathcal{S}^T$). The presence of many nodes of $\mathcal{S}^F$ (resp., $\mathcal{S}^T$) among the alters of the ego networks of $\mathcal{M}^F$ (resp., $\mathcal{M}^T$) is justified by the fact that both these categories of nodes were power addresses during the pre-bubble period. The difference between `from_addresses` and `to_addresses` in the Missings' ego networks is much smaller than the one observed in the Survivors' ego networks.

	Ego networks of $\mathcal{M}^F$			*Ego networks of* $\mathcal{M}^T$		
	Nodes of $\mathcal{S}^F$	*Nodes of* $\mathcal{E}^F$	*Nodes of* $\mathcal{M}^F$	*Nodes of* $\mathcal{S}^T$	*Nodes of* $\mathcal{E}^T$	*Nodes of* $\mathcal{M}^T$
$k = 1$	0.466	0.010	0.497	0.390	0.024	0.406
$k = 2$	0.277	0.0	0.214	0.162	0.0	0.225
$k = 3$	0.165	0.0	0.115	0.093	0.0	0.138
$k = 4$	0.098	0.0	0.070	0.056	0.0	0.089
$k = 5$	0.059	0.0	0.049	0.039	0.0	0.068
$k = 6$	0.040	0.0	0.033	0.031	0.0	0.052
$k = 7$	0.018	0.0	0.029	0.025	0.0	0.037
$k = 8$	0.004	0.0	0.027	0.021	0.0	0.032
$k = 9$	0.004	0.0	0.027	0.018	0.0	0.028

Table 7.11: Analysis of the presence of backbones linking the Missings during the pre-bubble period

Now, we conduct the same analysis for the Entrants' ego networks. The results obtained are shown in Table 7.12. The structure and the semantics of this table are similar to the ones of Tables 7.10 and 7.11. From the analysis of Table 7.12 we can conclude that there is no backbone linking the Entrants during the pre-bubble period. This result is justified considering that, during this period, the Entrants were not power addresses. The presence of some nodes of the Survivors or of the Missings in the alters of the Entrants is simply due to the fact that the Survivors and the Missings were power addresses during the pre-bubble period.

To also give a graphical idea of the results on the presence of backbones obtained above, we consider a social network $\mathcal{N}^F_{Pre}$, obtained from $\mathcal{N}_{Pre}$ considering only the power `from_addresses`.

In order to extract a subnet of $\mathcal{N}^F_{Pre}$ containing nodes strongly connected to each other, we should consider the cliques of $\mathcal{N}^F_{Pre}$. However, since the computation of cliques is an NP-hard problem, we decided to use a relaxation of the concept of clique and focused on k-core. We recall that a k-core of a network $\mathcal{N}$ is a connected

	Ego networks of $\mathcal{E}^F$			*Ego networks of* $\mathcal{E}^T$		
	Nodes of $\mathcal{S}^F$	*Nodes of* $\mathcal{E}^F$	*Nodes of* $\mathcal{M}^F$	*Nodes of* $\mathcal{S}^T$	*Nodes of* $\mathcal{E}^T$	*Nodes of* $\mathcal{M}^T$
$k=1$	0.326	0.140	0.163	0.194	0.0	0.222
$k=2$	0.140	0.0	0.023	0.083	0.0	0.056
$k=3$	0.070	0.0	0.0	0.056	0.0	0.056
$k=4$	0.0	0.0	0.0	0.056	0.0	0.056
$k=5$	0.0	0.0	0.0	0.056	0.0	0.028
$k=6$	0.0	0.0	0.0	0.056	0.0	0.028
$k=7$	0.0	0.0	0.0	0.056	0.0	0.028
$k=8$	0.0	0.0	0.0	0.056	0.0	0.028
$k=9$	0.0	0.0	0.0	0.056	0.0	0.028

Table 7.12: Analysis of the presence of backbones linking the Entrants during the pre-bubble period

maximal induced subnetwork of $\mathcal{N}$ in which all nodes have degree at least k. A k-core can be used as an indicator of the presence of backbones. In fact, if some nodes, say $n_1, n_2, \ldots, n_q$, belong to a k-core, then each of them will be connected to at least k of the other ones.

Consider the 5-core of $\mathcal{N}^F_{Pre}$ shown in Figure 7.4. In it, we indicate in yellow the Survivors nodes, in red the Missings nodes and in blue all the other ones. The 5-core consists of 175 nodes. As we can see from the figure, there is a strong backbone connecting 32 Survivors nodes and another weaker backbone connecting 13 Missings nodes. Consider, now, the 7-core of $\mathcal{N}^F_{Pre}$ shown in Figure 7.5. It contains even more strongly connected nodes than the 5-core. The total number of its nodes is 86. Again, there is a strong backbone connecting 19 Survivors nodes and a weaker backbone connecting 5 Missings nodes. Both these figures provide a graphical idea of the analytical results found previously.

The next analysis concerns the Survivors', the Missings' and the Entrants' ego networks during the bubble period. The results obtained by carrying out the same tasks seen for the pre-bubble period are reported in Tables 7.13, 7.14 and 7.15.

From the analysis of these tables we can detect the following knowledge patterns:

- There is a very strong backbone linking the Survivors, as can be seen by examining Table 7.13.
- In the same table, we can observe that there are some Entrants and Missings nodes among the alters of the Survivors' ego networks. This can be explained taking into account that the Entrants are power addresses during the bubble period, while the Missings, although not anymore, were power addresses in the period immediately before.
- Table 7.14 shows that there is no longer a backbone linking the Missings.

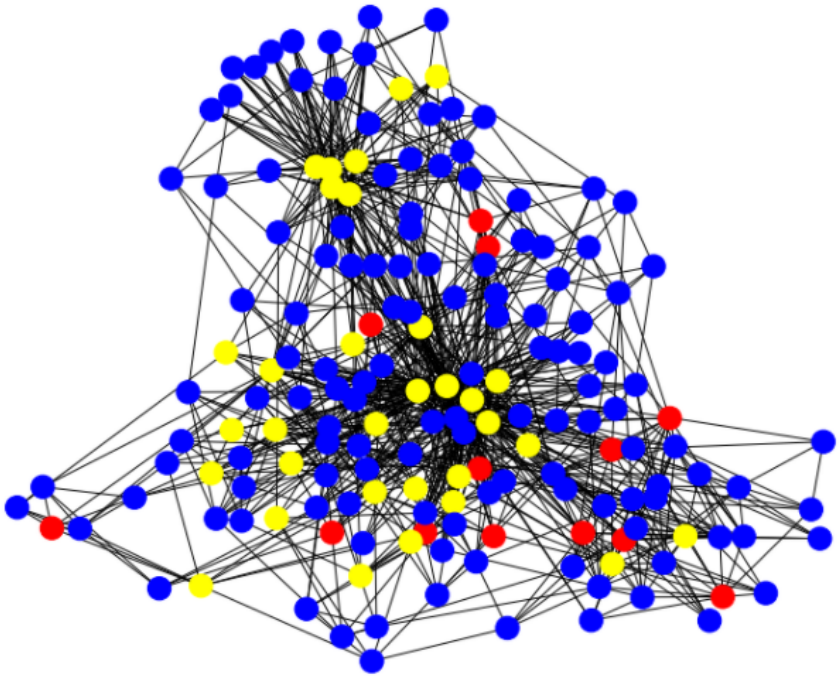


Fig. 7.4: A 5-core of $\mathcal{N}^F_{Pre}$

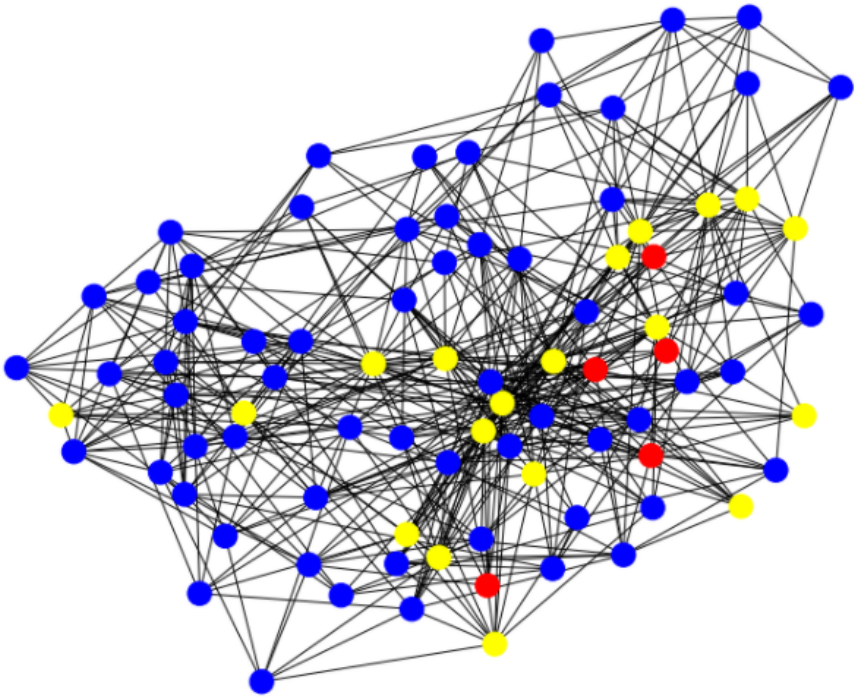


Fig. 7.5: A 7-core of $\mathcal{N}^F_{Pre}$

- Table 7.15 reveals that a backbone linking the Entrants starts to exist, even if it is not very strong yet.

	Ego networks of $\mathcal{S}^F$			*Ego networks of* $\mathcal{S}^T$		
	Nodes of $\mathcal{S}^F$	*Nodes of* $\mathcal{E}^F$	*Nodes of* $\mathcal{M}^F$	*Nodes of* $\mathcal{S}^T$	*Nodes of* $\mathcal{E}^T$	*Nodes of* $\mathcal{M}^T$
$k=1$	0.824	0.451	0.461	0.750	0.688	0.714
$k=2$	0.598	0.245	0.333	0.554	0.509	0.491
$k=3$	0.431	0.167	0.284	0.312	0.357	0.339
$k=4$	0.373	0.127	0.265	0.143	0.223	0.232
$k=5$	0.304	0.078	0.225	0.098	0.152	0.161
$k=6$	0.265	0.069	0.216	0.071	0.062	0.134
$k=7$	0.196	0.029	0.147	0.036	0.054	0.098
$k=8$	0.147	0.020	0.137	0.027	0.045	0.089
$k=9$	0.108	0.020	0.118	0.027	0.036	0.089

Table 7.13: Analysis of the presence of backbones linking the Survivors during the bubble period

	Ego networks of $\mathcal{M}^F$			*Ego networks of* $\mathcal{M}^T$		
	Nodes of $\mathcal{S}^F$	*Nodes of* $\mathcal{E}^F$	*Nodes of* $\mathcal{M}^F$	*Nodes of* $\mathcal{S}^T$	*Nodes of* $\mathcal{E}^T$	*Nodes of* $\mathcal{M}^T$
$k=1$	0.338	0.125	0.138	0.283	0.166	0.217
$k=2$	0.163	0.054	0.023	0.095	0.034	0.049
$k=3$	0.111	0.035	0.006	0.042	0.014	0.026
$k=4$	0.065	0.021	0.004	0.026	0.010	0.014
$k=5$	0.044	0.015	0.002	0.020	0.008	0.012
$k=6$	0.021	0.013	0.0	0.020	0.006	0.010
$k=7$	0.019	0.010	0.0	0.016	0.004	0.008
$k=8$	0.010	0.004	0.0	0.010	0.004	0.006
$k=9$	0.006	0.002	0.0	0.010	0.004	0.006

Table 7.14: Analysis of the presence of backbones linking the Missings during the bubble period

	Ego networks of $\mathcal{E}^F$			*Ego networks of* $\mathcal{E}^T$		
	Nodes of $\mathcal{S}^F$	*Nodes of* $\mathcal{E}^F$	*Nodes of* $\mathcal{M}^F$	*Nodes of* $\mathcal{S}^T$	*Nodes of* $\mathcal{E}^T$	*Nodes of* $\mathcal{M}^T$
$k=1$	0.337	0.572	0.217	0.335	0.477	0.335
$k=2$	0.175	0.295	0.127	0.152	0.284	0.152
$k=3$	0.096	0.169	0.084	0.081	0.142	0.081
$k=4$	0.066	0.096	0.054	0.061	0.076	0.051
$k=5$	0.048	0.066	0.042	0.061	0.046	0.030
$k=6$	0.036	0.030	0.036	0.056	0.030	0.025
$k=7$	0.024	0.024	0.036	0.046	0.030	0.020
$k=8$	0.024	0.0	0.036	0.041	0.025	0.015
$k=9$	0.024	0.0	0.036	0.036	0.025	0.015

Table 7.15: Analysis of the presence of backbones linking the Entrants during the bubble period

To also give a graphical idea of these results, we consider the $\mathcal{N}_B^F$ network. It is defined similarly to N_{Pre}^F, but taking the bubble period into account. We also consider the corresponding 5-core and 7-core, shown in Figures 7.6 and 7.7, respectively. In them, we represent the Survivors nodes in yellow, the Entrants nodes in

green and all the other nodes in blue. The 5-core consists of 149 nodes. Here, there is a very strong backbone involving 47 Survivors nodes and a weaker one involving 17 Entrants nodes. The 7-core consists of 67 nodes. Also in this case there is a very strong backbone connecting 30 Survivors nodes and a weaker backbone connecting 13 Entrants nodes.

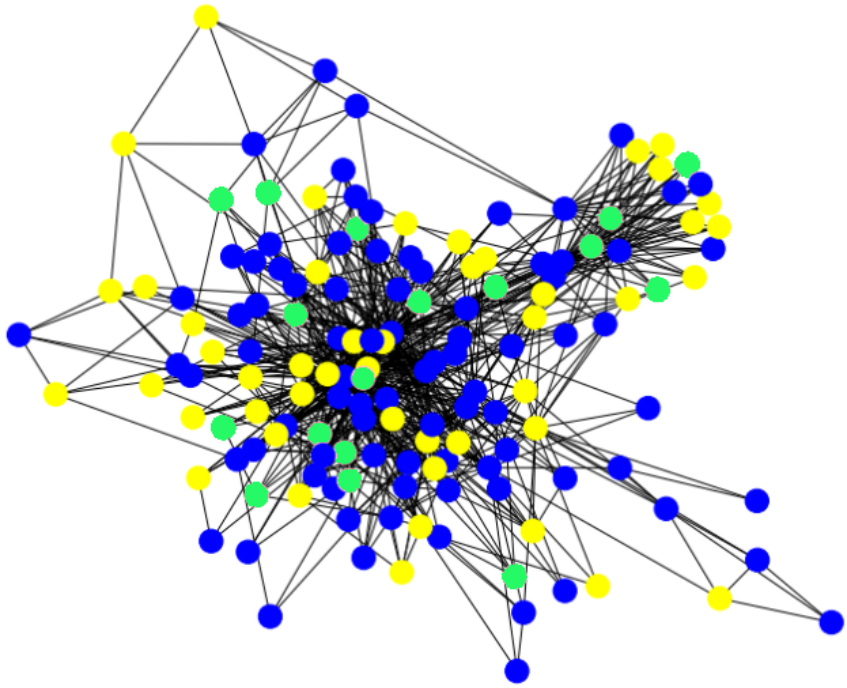


Fig. 7.6: A 5-core of $\mathcal{N}_B^F$

The last analysis concerns the Survivors', the Missings' and the Entrants' ego networks during the post-bubble period. The results obtained are reported in Tables 7.16, 7.17 and 7.18.

	Ego networks of $\mathcal{S}^F$			*Ego networks of* $\mathcal{S}^T$		
	Nodes of $\mathcal{S}^F$	*Nodes of* $\mathcal{E}^F$	*Nodes of* $\mathcal{M}^F$	*Nodes of* $\mathcal{S}^T$	*Nodes of* $\mathcal{E}^T$	*Nodes of* $\mathcal{M}^T$
$k = 1$	0.716	0.490	0.353	0.741	0.768	0.518
$k = 2$	0.510	0.265	0.206	0.607	0.598	0.330
$k = 3$	0.363	0.167	0.167	0.384	0.446	0.188
$k = 4$	0.265	0.147	0.108	0.223	0.366	0.143
$k = 5$	0.216	0.137	0.088	0.116	0.268	0.089
$k = 6$	0.186	0.098	0.078	0.080	0.223	0.089
$k = 7$	0.108	0.069	0.059	0.062	0.134	0.080
$k = 8$	0.088	0.059	0.049	0.045	0.098	0.062
$k = 9$	0.059	0.039	0.039	0.045	0.062	0.045

Table 7.16: Analysis of the presence of backbones linking the Survivors during the post-bubble period

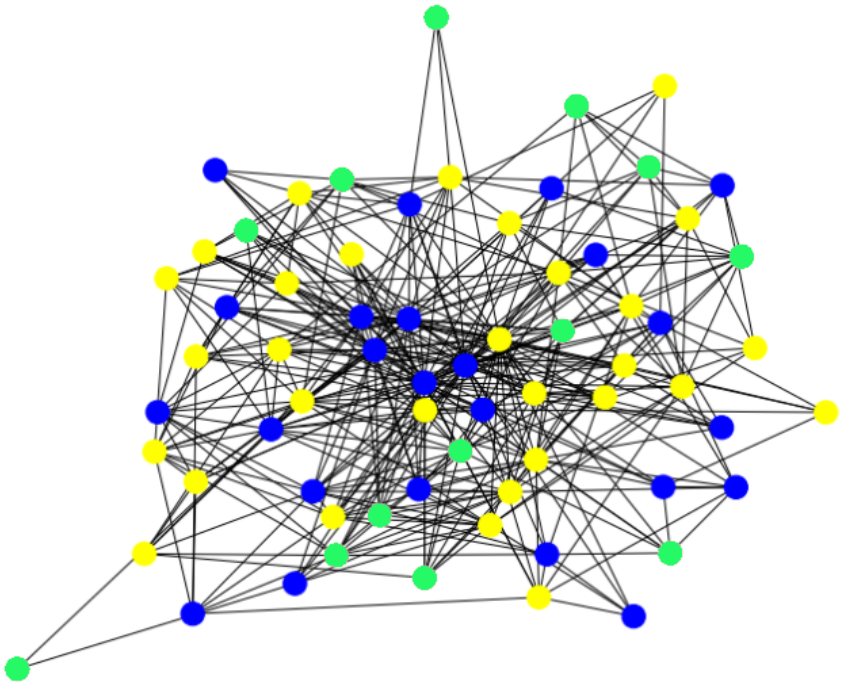


Fig. 7.7: A 7-core of $\mathcal{N}_B^F$

	Ego networks of $\mathcal{M}^F$			*Ego networks of* $\mathcal{M}^T$		
	Nodes of $\mathcal{S}^F$	*Nodes of* $\mathcal{E}^F$	*Nodes of* $\mathcal{M}^F$	*Nodes of* $\mathcal{S}^T$	*Nodes of* $\mathcal{E}^T$	*Nodes of* $\mathcal{M}^T$
$k = 1$	0.263	0.193	0.119	0.274	0.167	0.070
$k = 2$	0.122	0.126	0.015	0.067	0.040	0.027
$k = 3$	0.056	0.081	0.007	0.032	0.019	0.013
$k = 4$	0.033	0.059	0.007	0.027	0.011	0.008
$k = 5$	0.026	0.052	0.004	0.019	0.011	0.008
$k = 6$	0.015	0.041	0.004	0.016	0.008	0.005
$k = 7$	0.011	0.033	0.004	0.013	0.005	0.003
$k = 8$	0.011	0.022	0.0	0.011	0.005	0.0
$k = 9$	0.007	0.011	0.0	0.008	0.005	0.0

Table 7.17: Analysis of the presence of backbones linking the Missings during the post-bubble period

From the analysis of these tables we can deduce the following knowledge patterns:

- There is a strong backbone linking the Survivors, as can be seen in Table 7.16. Comparing Tables 7.13 and 7.16 we can see that this backbone, while continuing to remain strong, undergoes a weakening, compared to the pre-bubble period. This is physiological because, during the post-bubble period, the number of transactions made decreased considerably with respect to the ones of the bubble period.

	Ego networks of $\mathcal{E}^F$			*Ego networks of* $\mathcal{E}^T$		
	Nodes of $\mathcal{S}^F$	*Nodes of* $\mathcal{E}^F$	*Nodes of* $\mathcal{M}^F$	*Nodes of* $\mathcal{S}^T$	*Nodes of* $\mathcal{E}^T$	*Nodes of* $\mathcal{M}^T$
$k = 1$	0.331	0.651	0.211	0.431	0.675	0.376
$k = 2$	0.187	0.380	0.133	0.223	0.457	0.096
$k = 3$	0.133	0.193	0.084	0.091	0.310	0.036
$k = 4$	0.090	0.108	0.048	0.076	0.198	0.020
$k = 5$	0.054	0.078	0.048	0.071	0.122	0.015
$k = 6$	0.036	0.066	0.048	0.061	0.086	0.015
$k = 7$	0.036	0.042	0.048	0.061	0.056	0.015
$k = 8$	0.030	0.018	0.048	0.056	0.051	0.015
$k = 9$	0.024	0.018	0.042	0.056	0.046	0.010

Table 7.18: Analysis of the presence of backbones linking the Entrants during the post-bubble period

- We continue to observe the presence of some Entrants and Missings nodes among the alters of the Survivors' ego networks. The reasons for this fact are the same as those seen for the bubble period.
- The backbone linking the Missings, which had already started to disappear during the bubble period, has completely dissolved, as evidenced by the further decrease of the values in the fourth and seventh columns of Table 7.17, compared to the corresponding ones of Table 7.14.
- The backbone linking the Entrants, which was already visible during the bubble period, is further consolidated during the post-bubble period, as can be seen by examining Table 7.18.

Also in this case we can use k-cores to give a graphical idea of the results obtained. For this purpose, we consider the network $\mathcal{N}^F_{Post}$, obtained similarly to $\mathcal{N}^F_{Pre}$ and $\mathcal{N}^F_B$. We also consider the corresponding 5-core and 7-core, shown in Figures 7.8 and 7.9, respectively. The meaning of the colors of the nodes in this figure is the same as the one seen for Figures 7.6 and 7.7. In this case, the 5-core consists of 202 nodes. Here, there is a strong backbone linking 42 Survivors nodes. Furthermore, there is a backbone linking 31 Entrants nodes. Note that, compared to the bubble period, the backbone linking the Entrants nodes has strengthened. A similar reasoning also applies to the 7-core. It consists of 111 nodes. In it, we can observe a strong backbone linking 24 Survivors nodes and a backbone linking 16 Entrants nodes. Also this last backbone appears strengthened compared to the corresponding one relative to the 7-core during the bubble period shown in Figure 7.7. All these graphical results are totally in line with the analytical ones relative to the post-bubble period presented above.

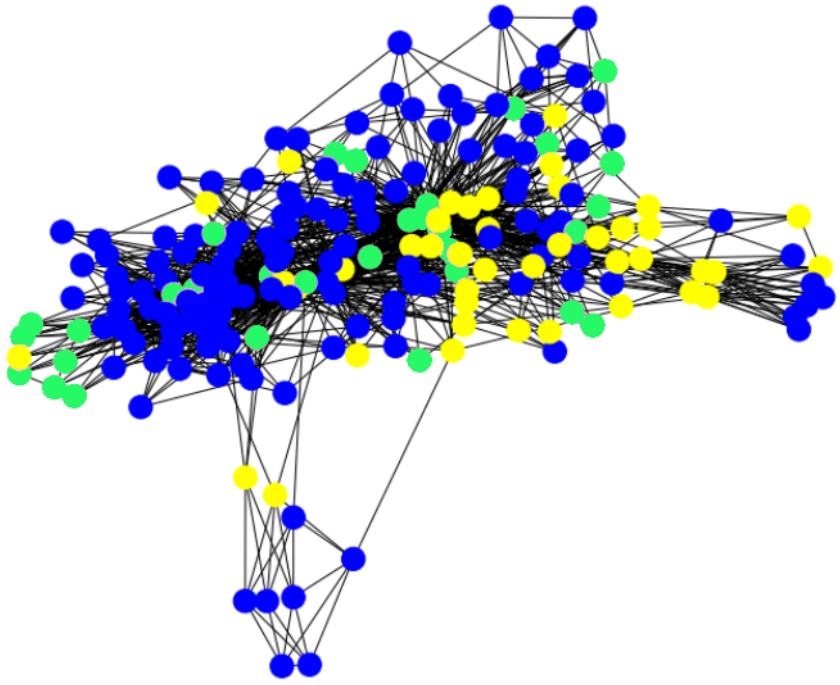


Fig. 7.8: A 5-core of $\mathcal{N}^F_{Post}$

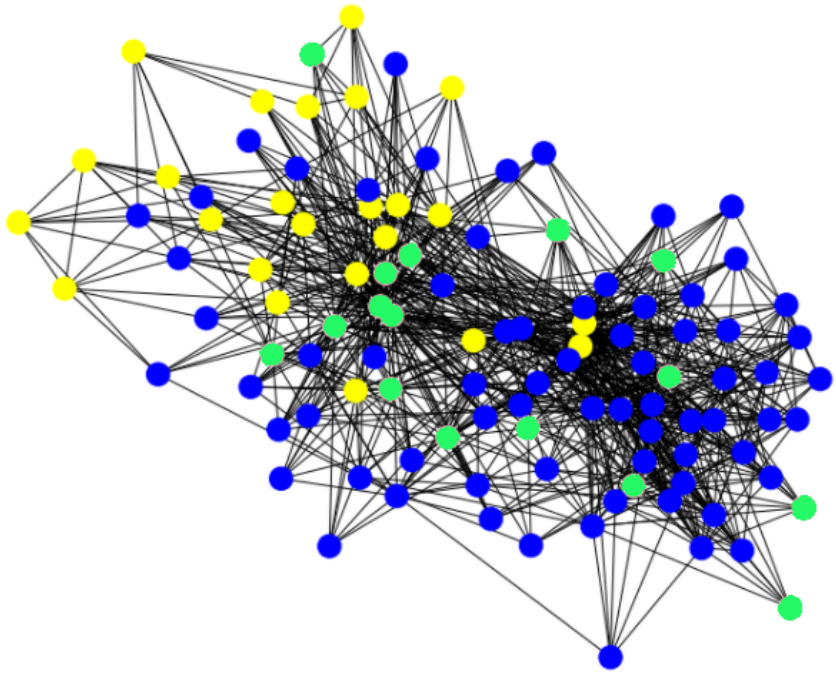


Fig. 7.9: A 7-core of $\mathcal{N}^F_{Post}$

7.2.2 Graphical backbone evaluations through k-trusses

In subsection 7.2.1, we have said that, in order to verify the possible existence of backbones among Survivors, Missings or Entrants, the concept of cliques could be

used. We have also said that the computation of cliques was a NP-hard problem and, for this reason, we chose to replace cliques with k-cores. In fact, the k-core concept is a relaxation of the clique concept and, unlike cliques, the computation of k-cores can be done in polynomial time. However, it is worth checking that the results obtained with k-core are not unduly influenced by the properties of this structure. One way to carry out this verification is to repeat the experiments performed with k-cores using another data structure that can be considered a relaxation of the clique concept and can be computed in polynomial time. To this end, we focused on the concept of k-truss [174]. A k-truss is a non-trivial, one component subgraph such that each edge is reinforced by at least $k-2$ pairs of edges making a triangle with that edge. Observe that each clique of order k is contained in a k-truss, whereas a k-truss does not necessarily contain a clique of order k. Furthermore, each k-truss is a subgraph of a (k-1)-core. All these properties support the idea that a k-truss is a concept that lies somewhere between the clique concept, which is too restrictive, and the k-core concept, which is too lax. Furthermore, similarly to k-cores and unlike cliques, the computation of k-trusses requires polynomial time.

At this point, similarly to what we did for the k-core, we computed the 5-truss of $\mathcal{N}^F_{Pre}$ and we saw that: *(i)* it consists of 152 nodes; *(ii)* there is a strong backbone connecting 27 Survivors; *(iii)* there is a weaker backbone connecting 7 Missings. Next, we computed the 7-truss of $\mathcal{N}^F_{Pre}$ and we obtained that: *(i)* it consists of 74 nodes; *(ii)* there is a strong backbone connecting 16 Survivors; *(iii)* there is no significant backbone among Missings.

Proceeding with our analysis, we computed the 5-truss of $\mathcal{N}^F_B$; analyzing it, we saw that: *(i)* it consists of 134 nodes; *(ii)* there is a very strong backbone involving 41 Survivors; *(iii)* there is an additional backbone involving 15 Entrants. The analysis of the 7-truss of $\mathcal{N}^F_B$ allows us to say that: *(i)* it consists of 61 nodes; *(ii)* there is a very strong backbone involving 26 Survivors; *(iii)* there is a weaker backbone involving 10 Entrants.

Our analysis on k-trussed ends with the computation of the 5-truss and 7-truss of $\mathcal{N}^F_{Post}$. Regarding the 5-truss we obtained that: *(i)* it consists of 194 nodes; *(ii)* there is a strong backbone connecting 36 Survivors; *(iii)* there is an additional backbone connecting 26 Entrants. Regarding the 7-truss of $\mathcal{N}^F_{Post}$ we saw that: *(i)* it consists of 96 nodes; *(ii)* there is a strong backbone connecting 22 Survivors; *(iii)* there is an additional backbone connecting 12 Entrants.

Comparing the results obtained through the k-truss analysis with those regarding the k-core analysis shown in subsection 7.2.1, we can observe that they are similar. In fact, the k-truss analysis confirms everything was found through the k-core analysis. The only exception regards the fact that the k-core analysis detects a backbone

(albeit a very weak one) between the Missings in the 7-core associated with $\mathcal{N}^{F}_{Pre}$. Such a backbone is not detected in the corresponding 7-truss. However, this minimal difference can be explained considering that the detected backbone of the 7-core is anyway very weak as well as taking into account that the concept of k-truss is more "severe" than the one of k-core.

At the end of this analysis, we can conclude that the strong similarity of the results obtained using k-cores and k-trusses allows us to say that these are intrinsic in the data and are not unduly caused by the properties of the k-cores.

7.2.3 Defining the identikit of bubble speculators

In the previous subsection, we extracted some knowledge patterns involving various kinds of addresses present in a cryptocurrency blockchain. In this subsection, we want to verify whether the suitable integration of these knowledge patterns allows us to build an identikit of speculators.

In performing this task we start with the information about the ego network obtained in subsection 7.1.3. It tells us that: *(i)* in the pre-bubble period, the Survivors have much larger ego networks than the other nodes; *(ii)* in the bubble and post-bubble periods, the Survivors have larger ego networks than the other nodes; *(iii)* in the bubble period, the Survivors' ego networks are much larger than even the Entrants' ego networks; this difference fades in the post-bubble period. Recall that having a large ego network means having the possibility to influence a large number of nodes.

Now, we consider the information on backbones extracted in subsection 7.2.1. It tells us that: *(i)* in the pre-bubble period, there is a strong backbone among the Survivors and a weaker backbone among the Missings; *(ii)* in the bubble period, there is a very strong backbone among the Survivors and a weaker backbone among the Entrants; this last is stronger than the corresponding one of the bubble period. Recall that the presence of a backbone among a set of nodes is an indicator that they tend to act in a coordinated way with each other.

We continue our investigation by considering the characteristics of the future main actors extracted in subsection 7.2.4. In that subsection, we saw that the address that best survive a bubble must be sought among those that, in the pre-bubble and bubble periods, made the most transactions and had the most contacts. But, from what we saw in subsection 7.1.3, the addresses with such characteristics are first those of the Survivors and then those of the Entrants.

Finally, an analysis of the nodes active in the period corresponding to the Ethereum bubble of the years 2017-2018 that are still active today also leads us to the same re-

sults, namely that most of the Survivors and a good portion of the Entrants present in the 2017-2018 Ethereum bubble are still active today.

All these considerations lead us to conclude that indeed in the Ethereum speculative bubble of 2017-2018, a group of speculators existed. Regarding the profile of the users belonging to this group, we can conclude that most of them were Survivors and were already present in the pre-bubble period. They are flanked in the bubble period by a group of speculators that formed the Entrants set. Initially, these were not the leaders of the phenomenon; at first, the leadership was of the Survivors alone. However, as time passed, the Entrants gradually consolidated and reached the level of leadership that previously characterized the Survivors alone.

7.2.4 Predicting the characteristics of the main future actors

All the previous analyses are mainly descriptive and diagnostic. In this subsection, instead, we want to go one step further proposing a predictive analysis with the aim of understanding, during a period (specifically, pre-bubble, bubble), what are the features of the addresses that will probably play a leading role during the next period (specifically, bubble, post-bubble). The importance of this analysis (in itself already evident) is reinforced by the results obtained in the previous subsection, telling us that these main actors are often connected by backbones. Consequently, identifying (and possibly acting on) some of them gives the possibility to identify (and act on) most of the others connected through the backbones.

In Table 7.19, we show the number of transactions, the number of contacts and the average value of transactions for the following addresses:

- T^F_{Pre}: the power `from_addresses` in the pre-bubble period.
- $\mathcal{S}^F$: the Survivors `from_addresses`. By definition, each element of $\mathcal{S}^F$ must also be an element of T^F_{Pre} and an element of T^F_B, i.e., the power `from_addresses` in the bubble period.
- $\mathcal{M}^F$: the Missings `from_addresses`. By definition, each element of $\mathcal{M}^F$ must also be an element of T^F_{Pre}, while it cannot belong to T^F_B.
- $\mathcal{E}^F_{Pre}$: the `from_addresses` that appeared in the bubble period but were already present (albeit not as power addresses) in the pre-bubble period. By definition, each element of $\mathcal{E}^F_{Pre}$ must also be an element of T^F_B, while it cannot belong to T^F_{Pre}.

From the analysis of this table we can see that the addresses of $\mathcal{S}^F$ have a significantly higher number of transactions and contacts than the corresponding ones not only of $\mathcal{M}^F$ and $\mathcal{E}^F_{Pre}$ but also of T^F_{Pre}. Instead, the average value of transactions is smaller for $\mathcal{S}^F$, $\mathcal{M}^F$ and T^F_{Pre} than for $\mathcal{E}^F_{Pre}$.

	T^F_{Pre}	$\mathcal{S}^F$	$\mathcal{M}^F$	$\mathcal{E}^F_{Pre}$
Average Number of Transactions	30,346.55	175,729.30	11,064.18	473.83
Average Number of Contacts	4,817.39	27,088.52	1,259.26	242.98
Average Value of Transactions (Eth)	8.65	8.18	7.32	106.53

Table 7.19: Average number of transactions, average number of contacts and average values of transactions for T^F_{Pre}, $\mathcal{S}^F$, $\mathcal{M}^F$ and $\mathcal{E}^F_{Pre}$

This result is even more evident considering Figure 7.10 (resp., 7.11). Here, we show the distribution of the addresses of $\mathcal{S}^F$ and $\mathcal{M}^F$ against the number of transactions (resp., contacts) of T^F_{Pre}. The abscissae axis is divided into deciles. In the figure, we indicate the decile with the highest values with D_{10} and the one with the lowest value with D_1. Figure 7.10 shows that most of the addresses of $\mathcal{S}^F$ belong to the highest deciles of T^F_{Pre}. This does not happen for the addresses of $\mathcal{M}^F$ that show a rather uniform distribution among the deciles of T^F_{Pre}, except for the lowest decile where they are almost absent. Figure 7.11 shows a similar trend except for the lowest decile, which comprises a lot of addresses for both $\mathcal{S}^F$ and $\mathcal{M}^F$.

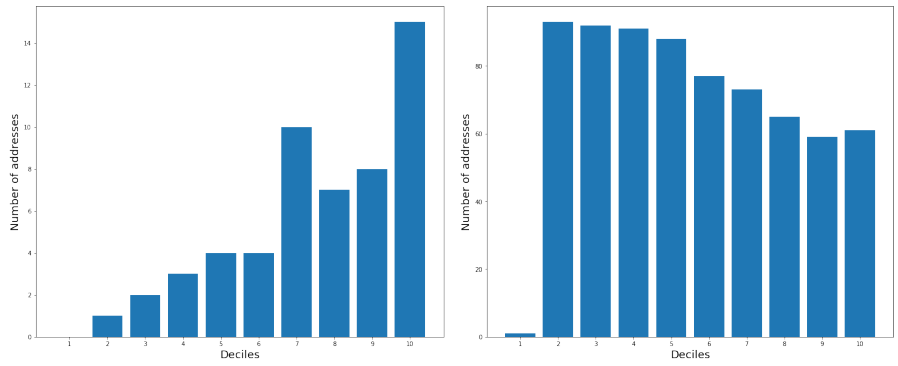


Fig. 7.10: Distribution of the addresses of $\mathcal{S}^F$ (at left) and $\mathcal{M}^F$ (at right) against the number of transactions of T^F_{Pre}

Both Table 7.19 and Figures 7.10 and 7.11 give us the same important following indication: "The addresses that will survive a bubble are to be searched among the ones that, in the pre-bubble period, have carried out the highest numbers of transactions and have the highest numbers of contacts". This indication is very strong for the number of transactions while it is a bit weaker for the number of contacts. In fact, as for this last parameter, we can see that the lowest decile contains a certain number not only of Missings nodes but also of Survivors ones.

Instead, Table 7.19 does not seem to give any indication on how searching, in the pre-bubble period, the future Entrants that will be among the main actors in the bubble and post-bubble periods.

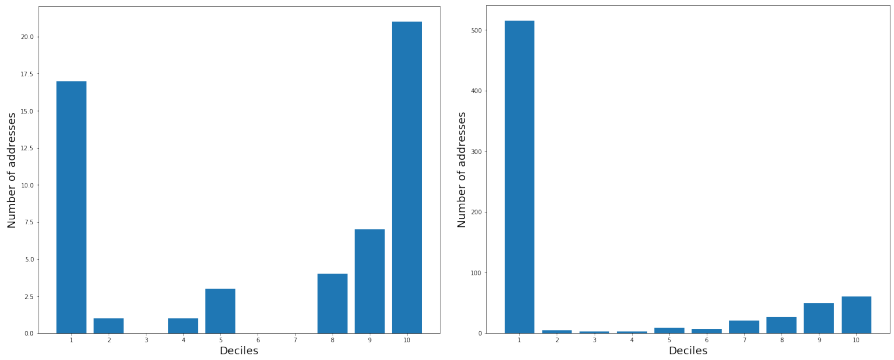


Fig. 7.11: Distribution of the addresses of $\mathcal{S}^F$ (at left) and $\mathcal{M}^F$ (at right) against the number of contacts of T^F_{Pre}

All previous analyses performed for `from_addresses` in the pre-bubble period can be repeated for `to_addresses` in the same period. In Table 7.20, we report the average number of transactions, the average number of contacts and the average value of transactions for T^T_{Pre}, $\mathcal{S}^T$, $\mathcal{M}^T$ and $\mathcal{E}^T_{Pre}$ (the latter defined similarly to $\mathcal{E}^F_{Pre}$, but for `to_addresses` instead of `from_addresses`). Furthermore, in Figure 7.12 (resp., 7.13), we show the distribution of the addresses of $\mathcal{S}^T$ and $\mathcal{M}^T$ against the number of transactions (resp., contacts) of T^T_{Pre}. Both the table and the two figures confirm, for `to_addresses`, the same results that we found previously for `from_addresses`.

	T^T_{Pre}	$\mathcal{S}^T$	$\mathcal{M}^T$	$\mathcal{E}^T_{Pre}$
Average Number of Transactions	28,035.76	138,663.66	10,121.69	599.78
Average Number of Contacts	5,329.76	23,007.33	2,165.56	294.28
Average Value of Transactions (Eth)	9.05	6.79	14.17	4.86

Table 7.20: Average number of transactions, average number of contacts and average value of transactions for T^T_{Pre}, $\mathcal{S}^T$, $\mathcal{M}^T$ and $\mathcal{E}^T_{Pre}$

So far we have examined pre-bubble data to identify some characteristics allowing us to predict who will be the main actors of the bubble period. Now, we want to do the same activity but examining bubble data to look for features allowing us to predict who will be the protagonists of the post-bubble period. In this analysis, we consider the following addresses:

- T^F_B: the top 1000 `from_addresses` in the bubble period;
- $\mathcal{S}^F$: the Survivors `from_addresses`;
- $\mathcal{E}^F$: the Entrants `from_addresses`.

In Table 7.21, we show the average number of transactions, the average number of contacts and the average value of transactions for T^F_B, $\mathcal{S}^F$ and $\mathcal{E}^F$.

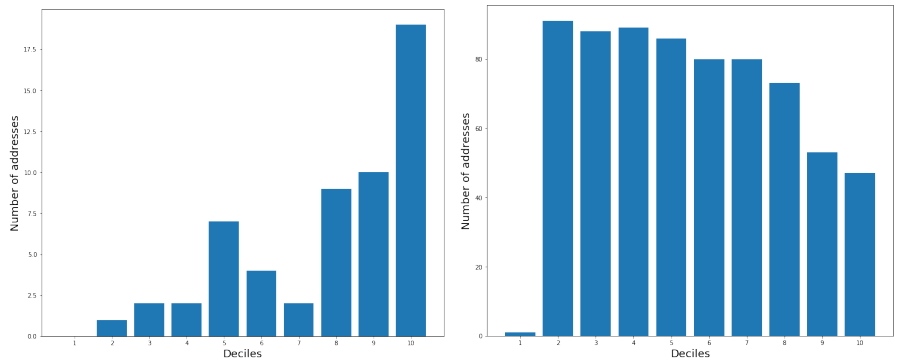


Fig. 7.12: Distribution of the addresses of $\mathcal{S}^T$ (at left) and $\mathcal{M}^T$ (at right) against the number of transactions of T^T_{Pre}

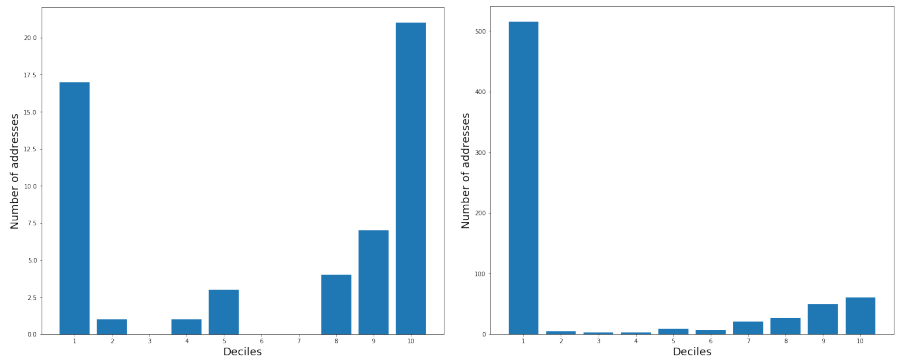


Fig. 7.13: Distribution of the addresses of $\mathcal{S}^T$ (at left) and $\mathcal{M}^T$ (at right) against the number of contacts of T^T_{Pre}

	T^F_B	$\mathcal{S}^F$	$\mathcal{E}^F$
Average Number of Transactions	45,418.29	266,183.77	46,010.31
Average Number of Contacts	10,100.95	55,029.89	12,851.75
Average Value of Transactions (Eth)	2.43	2.49	3.73

Table 7.21: Average number of transactions, average number of contacts and average value of transactions for T^F_B, $\mathcal{S}^F$ and $\mathcal{E}^F$

From the analysis of Table 7.21 we can see that, once again, it is easy to identify the Survivors of the post-bubble period. In fact, they generally have a significantly higher number of transactions and contacts than the other power `from_addresses`. Instead, the Entrants are not easily distinguishable, because they have only slightly more transactions and contacts than the other power `from_addresses`. This represents a confirmation of what we had deduced from the analysis of Tables 7.13 - 7.18 and Figures 7.6 - 7.9, where we derived that the set of the Entrants is formed during the bubble period but it consolidates only during the post-bubble period.

This result is confirmed and substantially reinforced by Figures 7.14 and 7.15. In them, we can see that the Survivors are in the highest deciles, and this was expected considering the results of Table 7.21. However, a similar trend, although less marked, is also found for the Entrants. This represents a further important result because it allows us to define, at least partially, which nodes will be the Entrants in the post-bubble period. Similarly to what happened in the pre-bubble period, the distribution against the number of transactions is better than the one against the number of contacts in discriminating the Survivors and the Entrants against the other nodes during the post-bubble period. Indeed, in the case of the number of contacts, there is a certain number of addresses in the lowest decile, which, in fact, represents an outlier.

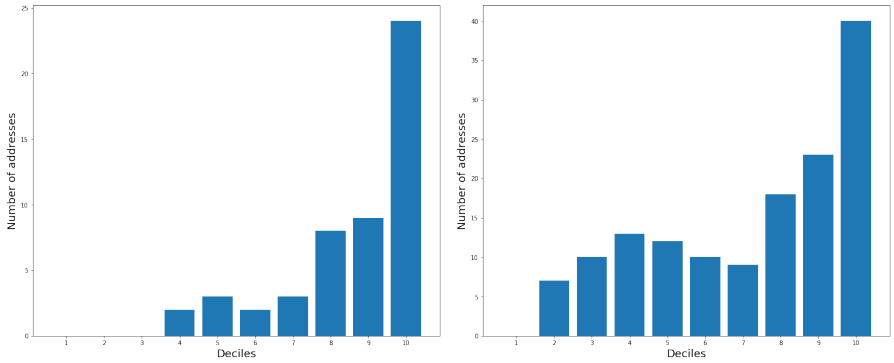


Fig. 7.14: Distribution of the addresses of $\mathcal{S}^F$ (at left) and $\mathcal{E}^F$ (at right) against the number of transactions of T_B^F

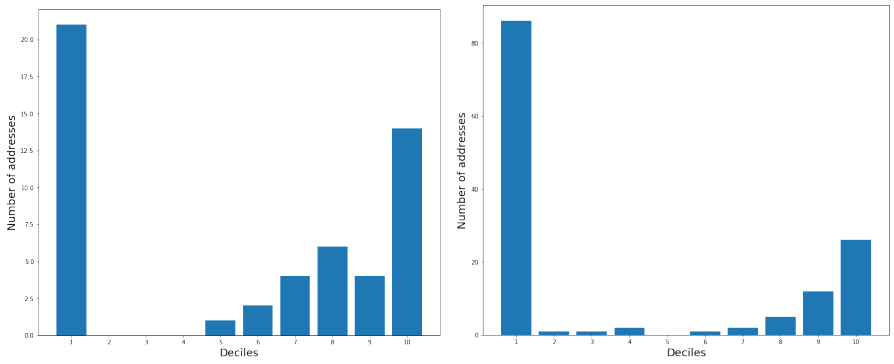


Fig. 7.15: Distribution of the addresses of $\mathcal{S}^F$ (at left) and $\mathcal{E}^F$ (at right) against the number of contacts of T_B^F

Both Table 7.21 and Figures 7.14 and 7.15 give us the same important following indication: "The addresses that will survive a speculative bubble are to be searched

among those that, in the bubble period, have carried out the highest numbers of transactions and have the highest numbers of contacts. If they also had this property in the pre-bubble period they belong to the Survivors, otherwise they belong to the Entrants.".

All previous analyses performed for `from_addresses` in the bubble period can be repeated for `to_addresses` in the same period. In Table 7.22, we report the average number of transactions, the average number of contacts and the average value of transactions for T_B^T, $\mathcal{S}^T$ and $\mathcal{E}^T$. Furthermore, in Figure 7.16 (resp., 7.17), we show the distribution of the addresses of $\mathcal{S}^T$ and $\mathcal{E}^T$ against the number of transactions (resp., contacts) of T_B^T.

	T_B^T	$\mathcal{S}^T$	$\mathcal{E}^T$
Average Number of Transactions	49,912.89	219,068.94	58,823.91
Average Number of Contacts	11,963.66	45,949.34	14,134.10
Average Value of Transactions (Eth)	1.90	1.98	1.71

Table 7.22: Average number of transactions, average number of contacts and average value of transactions for T_B^T, $\mathcal{S}^T$ and $\mathcal{E}^T$

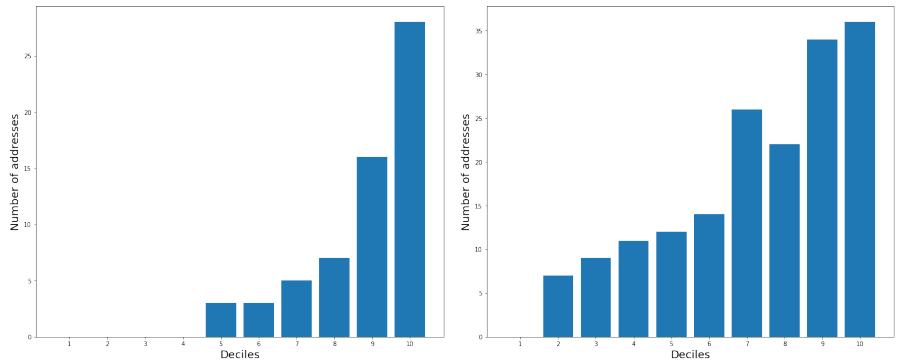


Fig. 7.16: Distribution of the addresses of $\mathcal{S}^T$ (at left) and $\mathcal{E}^T$ (at right) against the number of transactions of T_B^T

Table 7.22 and Figure 7.16 confirm, for `to_addresses`, the same results we found previously for `from_addresses`. Figures 7.17, if compared with Figure 7.15, shows that, as for the number of contacts of the Survivors, the outlier represented by the lowest decile is strongly reduced. Instead, this outlier remains for the Entrants. However, for this last category of addresses, we can observe that, similarly to what happens for the Survivors, and differently from what happened in Figure 7.15, most of the addresses are in the highest deciles, even if, once again, this phenomenon is less marked than the corresponding one observed for the Survivors.

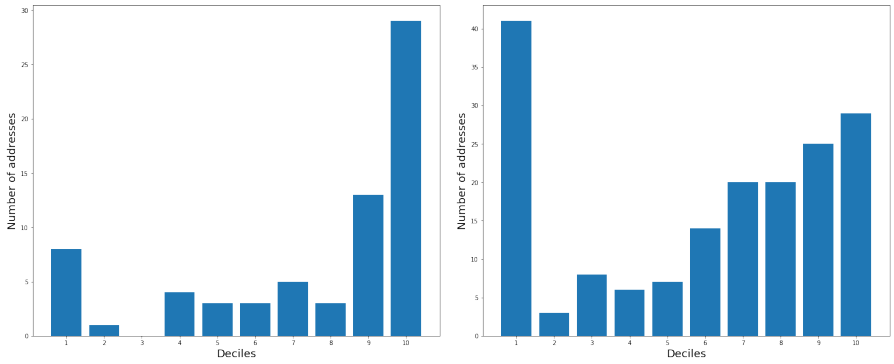


Fig. 7.17: Distribution of the addresses of $\mathcal{S}^T$ (at left) and $\mathcal{E}^T$ (at right) against the number of contacts of T_B^T

As a last analysis, we investigated how the power addresses of the post-bubble period behaved during the months following the time interval considered for our dataset, i.e., from January 2019 until today. For this purpose, we considered three subsets of the power addresses, i.e., the Survivors, the Entrants and the other nodes (hereafter, the Others), and we examined the date of the last transaction for them. The distribution of the Survivors (resp., the Entrants, the Others) against this date is shown in Figure 7.18 (resp., 7.19, 7.20) for `from_addresses`, and in Figure 7.21 (resp., 7.22, 7.23) for `to_addresses`. From the analysis of these figures we can observe that:

- As for `from_addresses`, we can see that most of the Survivors are still active. Many Entrants are also active but, unlike the Survivors, there is a fraction of them that ceased to operate in the second half of 2019. The date of the end of activity of the Others is, instead, more uniformly distributed. This is a further confirmation that the Survivors represent the vast part of the guiding users in Ethereum.
- As far as `to_addresses` are concerned, we can see that most of the Survivors and the Entrants are still active. The date of the end of activity of the Others is distributed in a more balanced way, even if there is a large amount of addresses still active also in this case. Therefore, as for `to_addresses`, we can deduce that the Survivors include most of the guiding users in Ethereum. However, differently from what happens for `from_addresses`, they have been flanked as leaders by the Entrants.

7.2.5 Adoption of our approach in the next speculative bubble

The main objective of this Chapter was to study the cryptocurrency speculative bubble during the years 2017-2018 to understand the behavior of some particularly in-

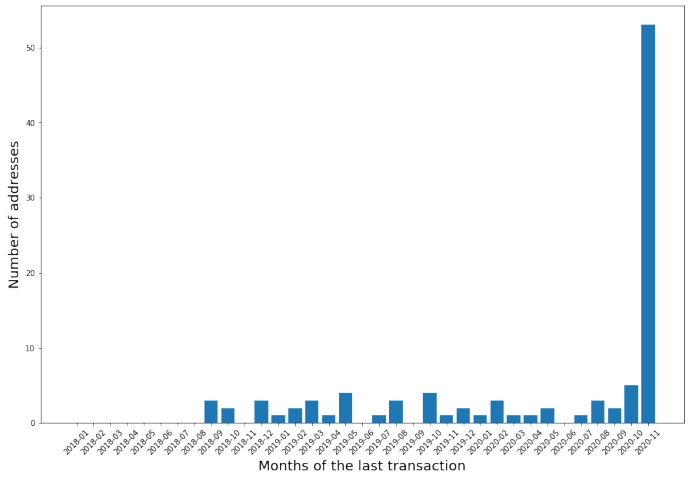


Fig. 7.18: Distribution of the Survivors (`from_addresses`) against the date of the last transaction

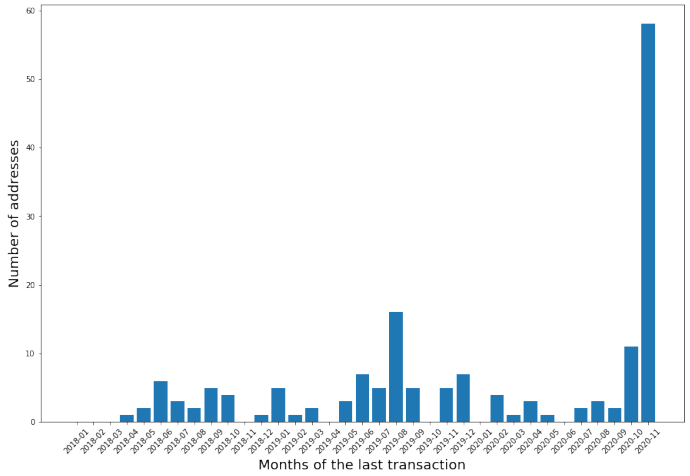


Fig. 7.19: Distribution of the Entrants (`from_addresses`) against the date of the last transaction

teresting categories of users and to try to identify a profile of possible speculators. However, the knowledge pattern extracted in this way do not represent only an abstract knowledge related to a past event, but can become an extremely valuable tool for the future.

In fact, the cryptocurrency context is considered a highly speculative environment by many graduates of the Nobel Memorial Prize in Economic Sciences, central bankers and investors. Speculations on cryptocurrencies have also been observed recently. For example, on March 8^{th}, 2020 the price of Bitcoin was 8,901 USD. On March 12^{th}, 2020, it was 6,206 USD, with a decrease of about 30%. In October 2020 this price was already doubled again and was about 13,000 USD. On January 3^{rd}, 2021 the price of Bitcoin was 34,792 USD; the next day it decreased by 17%. On

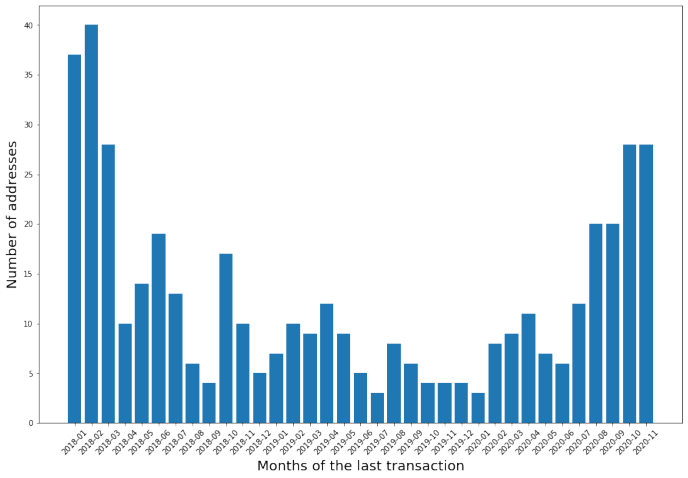


Fig. 7.20: Distribution of the Others (`from_addresses`) against the date of the last transaction

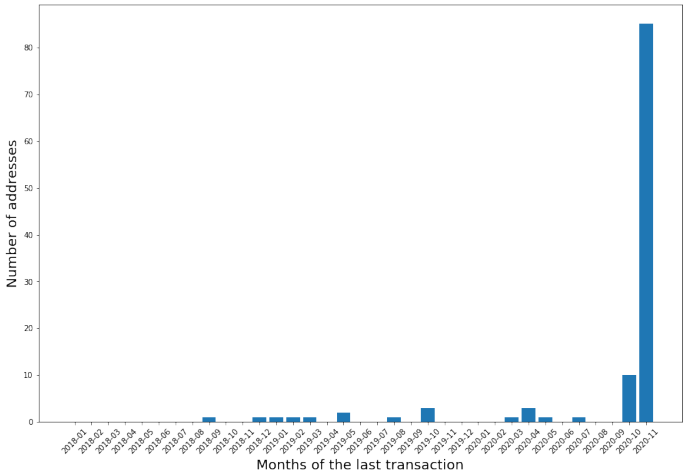


Fig. 7.21: Distribution of the Survivors (`to_addresses`) against the date of the last transaction

January 8^{th}, 2021 its value exceeded 40,000 USD and on February 16^{th}, 2021 it exceeded 50,000 USD. In March 2021 its value was 58,734 USD, while on May 9^{th}, 2021 it reached its highest value in history being 58,788 USD. On May 18^{th}, 2021 (which corresponds to the time of writing of this subsection) it had fallen again to 43,144 USD losing 26.61% of its value in 9 days.

Similar trends apply to other cryptocurrencies. For example, the value of Ether was about 750 USD in December 2020, about 1,350 USD in January 2021, about 1,800 USD in March 2021 and about 2,700 USD in April 2021. On May 12^{th}, 2021 this value was equal to 4,132.76 USD and represents the highest value reached by this currency so far. On May 15^{th}, 2021 its value was still 4,100.03 USD. On May

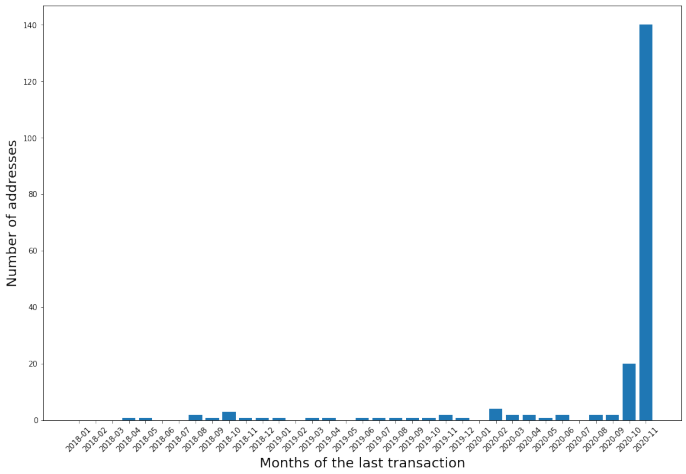


Fig. 7.22: Distribution of the Entrants (`to_addresses`) against the date of the last transaction

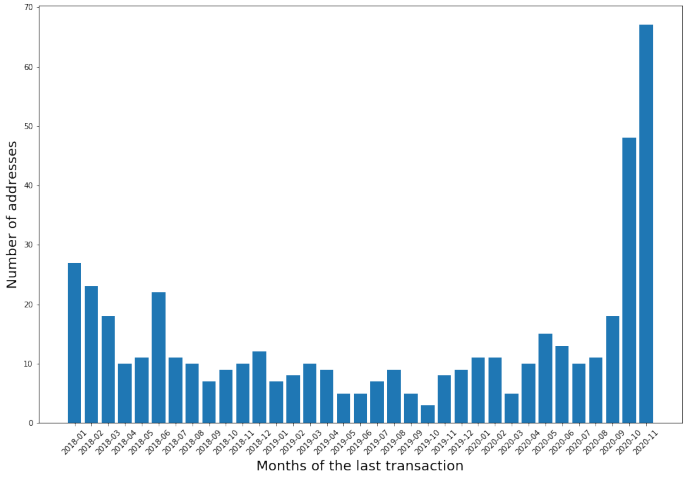


Fig. 7.23: Distribution of the Others (`to_addresses`) against the date of the last transaction

16^{th}, 2021 (which corresponds to the time of writing of this subsection) the value of the Ether was 3,231.94 USD with a collapse of 21.81% in 6 days.

The above examples highlight how prone the cryptocurrency world is to speculation. In addition, the trends of the last month lead us to believe that we are in the midst of a speculative bubble similar to the one of 2017-2018. If this is the case, the proposed approach would allow us to extract many knowledge patterns about the behaviors of the various players operating in this market and could even support analysts in understanding who are the speculators behind these bubbles. Therefore, we believe that the proposed approach has not only a value for the past but it provides useful predictive tools for the present and for the future.

8

Representation, detection and usage of the content semantics of comments

The analysis of people's comments in social platforms is a widely investigated topic because comments are the place where people show their spontaneity most clearly. In this chapter, we present a network-based data structure and a related approach to represent and manage the underlying semantics of a set of comments. Our approach is based on the extraction of text patterns that take into account not only the frequency, but also the utility of the analyzed comments. Our data structure and approach are "multidimensional" and "holistic", in the sense that they can simultaneously handle content semantics from multiple perspectives. They are also easily extensible, because additional content semantics perspectives can be easily added to them. Furthermore, our approach is able to evaluate the semantic similarity of two sets of comments. In this chapter, we also illustrate the results of several tests we conducted on Reddit comments, even if our approach can be applied to any social platform. Finally, we provide an overview of some possible applications of this research.

The material presented in this chapter was derived from [258].

8.1 Methods

8.1.1 Comment filtering and text pattern extraction

In this section, we present our approach to filter the starting set of comments and construct a set of text patterns from them. These represent the core for the construction of the CS-Nets to be used in the various applications of interest and which we illustrate in Section 8.3.

Our approach receives a set of comments. These should hopefully be homogeneous (e.g., comments related to the same post, comments written by the same user, comments present in a certain subreddit, comments related to a very specific topic or written at a very particular time of the year). Actually, in principle, comments should also be randomly selected, although this would make little sense in real applications.

Our approach first proceeds with a phase of Data Cleaning and Annotation. During this phase, it performs:

- The removal of bot-generated content.
- The cleaning of the textual content present in the comments and the next tokenization and lemmatization of these last ones.
- The annotation of data performed by associating a sentiment value with each comment; for this purpose, we use the compound score [317]. This last technique returns a sentiment value between -1 (most extreme negative) and +1 (most extreme positive).
- The enrichment of comments with features regarding them, their users and the posts they refer to.

Once the Data Cleaning and Annotation activities have been completed, our approach proceeds with the extraction of text patterns from the comments thus obtained. In this activity, an important role is played by pattern mining. This is a well known task in the literature, which aims at extracting text patterns with certain characteristics from a set of lemmatized texts (which, in our case, are the lemmatized comments obtained at the end of the previous phase).

Generally, the extraction of patterns is carried out based on their frequency assuming that a pattern is more important the more frequent it is [248, 11, 448, 259]. This assumption is true in most cases, but there are situations where it does not hold. In fact, there could exist patterns characterized by a low frequency but an extremely high utility (given a certain notion of it). Several utility functions have been introduced to handle this situation. In this way, the focus shifts from frequent pattern mining to High Utility Pattern Mining (hereafter, HUPM) [247, 260, 657]. In this case, a utility function denotes an ordering of user preferences over a set of choices [269]. Consequently, it is a subjective measure and depends on the user's preferences.

Once the utility function of interest is defined, our approach operates as follows. First, it extracts patterns having a frequency higher than a minimum threshold. For this purpose, it can use one of the classical techniques for frequent pattern mining, such as FPGrowth [295]. Then, it associates each pattern with the features appearing in the comments it is present in. These features will be used for the next analyses. Afterwards, it applies the chosen utility function to each pattern for computing the pattern's utility value. Finally, it selects and returns those patterns whose utility value is greater than a minimum threshold.

If we choose to filter only extremely rare patterns, and therefore to give a little weight to frequency, the utility function plays a key role in filtering patterns and

allows us to direct the pattern selection towards a strategy rather than another. Two utility functions very interesting in our case are the following:

- The average sentiment value of the comments which the pattern of interest, say p_j, refers to. It can be formalized as:

$$f_s(p_j) = avg_{c_{j_k} \in \mathcal{C}_j}\{\gamma(c_{j_k})\}$$

  Here: *(i)* $f_s(\cdot)$ is the utility function we are defining; *(ii)* p_j is the generic pattern, of which we want to compute the utility function; *(iii)* $\mathcal{C}_j$ is the set of comments in which p_j is present; *(iv)* $\gamma(\cdot)$ is a function that receives a comment and returns its compound score (and, therefore, its sentiment value); *(v)* $avg(\cdot)$ is a function computing the average of the values received as input.
- The Pearson's correlation [495] between the sentiment and the score of the comments where a certain pattern p_j is present. Recall that the Pearson's correlation is a measure of the linear correlation between two sets of data. Its value belongs to the real interval $[-1, 1]$, where -1 (resp., 1) denotes a negative (resp., positive) linear correlation, while 0 indicates a lack of correlation. It can be formalized as follows:

$$f_p(p_j) = \frac{\sum_{i=1}^{n}(x_i-\overline{x})(y_i-\overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\overline{y})^2}}$$

  Here: *(i)* p_j and $\mathcal{C}_j$ have been already explained for $f_s(\cdot)$; *(ii)* X (resp., Y) is the set of sentiment values (resp., score) related to the comments of $\mathcal{C}_j$; *(iii)* x_i (resp., y_i) indicates the i^{th} element of X (resp., Y); $\overline{x}$ (resp., $\overline{y}$) represents the mean of the values of X (resp., Y). Note that a positive (resp., negative) value of $f_p(\cdot)$ indicates that there is a direct (resp., inverse) correlation between the sentiment elicited by a comment and the score it gets. During the experimental campaign, which we describe in Section 4.2, we observed that there exist many patterns and comments with negative values of $f_p(\cdot)$. This allows us to say that a positive (resp., negative) sentiment in a comment does not necessarily lead it to receive a high (resp., low) score. This is especially true for certain kinds of comment, e.g., those related to Not Safe For Work (resp., NSFW) posts, which are the ones investigated in the experiments of this paper.

We end this section by pointing out that many other utility functions could be defined. Here, we have focused on $f_s(\cdot)$ and $f_p(\cdot)$ to give an idea of their potential and possible variety.

8.1.2 Content Semantics Network definition

Let $\mathcal{C} = \{c_1, c_2, \cdots c_n\}$ be a set of lemmatized comments and let $\mathcal{L} = \{l_1, l_2, \cdots, l_q\}$ be the set of all lemmas that can be found in a comment of $\mathcal{C}$. Each comment $c_k \in \mathcal{C}$ can

be represented as a set of lemmas $c_k = \{l_1, l_2, \ldots, l_m\}$. As a consequence, we have that $c_k \subseteq \mathcal{L}$.

A text pattern p_h is a set of lemmas; more specifically, $p_h \subseteq \mathcal{L}$. In principle, p_h can occur in zero, one or more comments of $\mathcal{C}$. Actually, as pointed out above, we are interested in those patterns whose frequency and utility function are higher than two suitable thresholds. In the following, we call $\mathcal{P}$ this set of patterns.

A Content Semantics Network (hereafter, CS-Net) $\mathcal{N}$ is defined as:

$$\mathcal{N} = \langle N, A^c \cup A^r \rangle$$

N is the set of nodes of $\mathcal{N}$. There is a node $n_i \in N$ for each lemma $l_i \in \mathcal{L}$. Since there exists a biunivocal correspondence between n_i and l_i, in the following we will use these two symbols interchangeably.

A^c is the set of co-occurrence arcs. An arc $(n_i, n_j, w_{ij}) \in A^c$ indicates that the lemmas l_i and l_j appear at least once together in a pattern of $\mathcal{P}$. w_{ij} is a real number in the interval $[0,1]$ denoting the strength of the co-occurrence. The higher w_{ij}, the higher this strength. For instance, w_{ij} can be computed considering the number of patterns in which l_i and l_j co-occur.

A^r is the set of semantic relationship arcs. An arc $(n_i, n_j, w_{ij}) \in A^r$ denotes that there exists a form of semantic relationship between l_i and l_j. w_{ij} is a real number in the interval $[0,1]$ denoting the strength of the relationship. The higher w_{ij}, the higher this strength. For instance, w_{ij} can be computed using ConceptNet [405] and taking into account the number of times in which l_j is present in the set of "related terms" of l_i, along with the values of the corresponding weights.

A comment about the structure of the CS-Net is in order. As specified in the Introduction, in this paper we want to make an effort to define the semantics of a set of contents, for example those published in comments to Reddit posts. The CS-Net is intended as a tool to support this activity. For this purpose, it considers two perspectives derived from the past literature. The first is related to the concept of co-occurrence and specifies that two semantically related lemmas which tend to appear together very often in sentences. The second concerns the concept of relationships and semantically related terms. These summarize the results of several researches carried out in the past both in Information Retrieval [202] and Natural Language Processing [95].

Clearly, additional perspectives could be considered and we also do not exclude doing so in the future. From this point of view, we highlight that our model is highly scalable. In fact, if we wanted to consider a further perspective, it will be sufficient to flank A^c and A^r with an additional set of arcs that represents this new perspective.

8.1.3 Evaluation of the semantic similarity of two CS-Nets

In this section, we illustrate our approach for computing the semantic similarity of the contents expressed by two CS-Nets $\mathcal{N}_1$ and $\mathcal{N}_2$. In the previous section, we have said that the CS-Net model currently adopts two perspectives for the semantic similarity evaluation, namely co-occurrences and semantic relationship between lemmas (see Section 8.1.2). We have also said that this model is scalable allowing the adoption of new perspectives, if desired. We aim to preserve such scalability also in the approach to evaluate the semantic similarity of two CS-Nets we are presenting here.

Given this premise, we are now ready to describe our approach. It receives two CS-Nets $\mathcal{N}_1$ and $\mathcal{N}_2$ and returns a coefficient σ_{12} that measures the semantic similarity of the contents represented by $\mathcal{N}_1$ and $\mathcal{N}_2$. For this purpose:

- It constructs two pairs of subnetworks $(\mathcal{N}_1^c, \mathcal{N}_2^c)$ and $(\mathcal{N}_1^r, \mathcal{N}_2^r)$, obtained by selecting only the co-occurrence and semantic relationship arcs from the networks $\mathcal{N}_1$ and $\mathcal{N}_2$, respectively. Specifically:

$$\mathcal{N}_1^c = \langle \mathcal{N}_1, A_1^c \rangle \qquad \mathcal{N}_2^c = \langle \mathcal{N}_2, A_2^c \rangle \qquad \mathcal{N}_1^r = \langle \mathcal{N}_1, A_1^r \rangle \qquad \mathcal{N}_2^r = \langle \mathcal{N}_2, A_2^r \rangle$$

  If, in the future, the number of perspectives, and therefore the number of arc sets, increases, it will be sufficient to build a pair of subnetworks for each perspective.
- It determines the weights to be associated with the two subnetworks. These weights are computed as:

$$\omega_1^c = \frac{|A_1^c|}{|A_1^c|+|A_1^r|} \qquad \omega_2^c = \frac{|A_2^c|}{|A_2^c|+|A_2^r|} \qquad \omega_1^r = 1 - \omega_1^c \qquad \omega_2^r = 1 - \omega_2^c$$

$$\omega_{12}^c = \frac{\omega_1^c+\omega_2^c}{2} \qquad \omega_{12}^r = \frac{\omega_1^r+\omega_2^r}{2}$$

  The reasoning underlying these formulas is that, in determining the overall semantics of a content, the importance of a perspective with respect to the other ones is directly proportional to the number of pairs of lemmas it is able to involve.
- It computes the semantic similarity degree σ_{12}^c and σ_{12}^r for the pairs of networks $(\mathcal{N}_1^c, \mathcal{N}_2^c)$ and $(\mathcal{N}_1^r, \mathcal{N}_2^r)$, respectively. We describe this computation in detail in Subsection 8.1.4.
- It computes the overall semantic similarity degree σ_{12} associated with the networks $\mathcal{N}_1$ and $\mathcal{N}_2$ as a weighted mean of the two semantic similarity degrees σ_{12}^c and σ_{12}^r:

$$\sigma_{12} = \frac{\omega_{12}^c \cdot \sigma_{12}^c + \omega_{12}^r \cdot \sigma_{12}^r}{\omega_{12}^c + \omega_{12}^r}$$

  If we set:

$$\alpha = \frac{\omega_{12}^c}{\omega_{12}^c+\omega_{12}^r} = \frac{\omega_1^c+\omega_2^c}{2} = \frac{1}{2}\cdot\left(\frac{|A_1^c|}{|A_1^c|+|A_1^r|} + \frac{|A_2^c|}{|A_2^c|+|A_2^r|}\right)$$

then, the formula for the computation of σ_{12} can be written as:

$$\sigma_{12} = \alpha\cdot\sigma_{12}^c + (1-\alpha)\cdot\sigma_{12}^r$$

In this formula, α is a coefficient that weights the semantic similarity defined through co-occurrences against the one defined through semantic relationships between lemmas. The rationale behind the formula of α is that the greater the amount of information carried by one perspective, compared to another, the greater its weight in defining the overall semantics. Now, since $|N_1^c| = |N_1^r|$ and $|N_2^c| = |N_2^r|$, the amount of information carried by co-occurrences with respect to semantic relationships between lemmas can be computed by considering the cardinality of the corresponding sets of arcs. Finally, note that σ_{12} ranges in the real interval $[0,1]$. The higher σ_{12}, the greater the similarity of $\mathcal{N}_1$ and $\mathcal{N}_2$.

Our approach for the computation of σ_{12} is extensible, because if in the future we want to enrich the CS-Net model with additional perspectives to model content semantics, it will be sufficient to flank to σ_{12}^c and σ_{12}^r an additional similarity coefficient for each perspective and modify the formula for the computation of σ_{12} accordingly.

8.1.4 Semantic similarity degree computation

In the previous section, we have seen that our approach for computing the similarity between two CS-Nets $\mathcal{N}_1$ and $\mathcal{N}_2$ constructs "projections" or "subnetworks" for each network (i.e., $\mathcal{N}_1^c$ and $\mathcal{N}_1^r$ for $\mathcal{N}_1$, and $\mathcal{N}_2^c$ and $\mathcal{N}_2^r$ for $\mathcal{N}_2$), computes the similarity coefficients σ_{12}^c between $\mathcal{N}_1^c$ and $\mathcal{N}_2^c$, and σ_{12}^r between $\mathcal{N}_1^r$ and $\mathcal{N}_2^r$ separately, and then combines them appropriately. In this context, the way in which the coefficient σ_{12}^x, $x \in \{c,r\}$, is computed becomes extremely important.

In order to define an approach for the computation of σ_{12}^x as holistic as possible, we strove to define a formula that takes into account more factors that may influence the semantic similarity degree of two networks $\mathcal{N}_1^x$ and $\mathcal{N}_2^x$, $x \in \{c,r\}$. In particular, there are at least two factors that we think can contribute to define this semantic similarity degree.

The first factor concerns the topological similarity of the networks, and thus the similarity of their structural features (e.g., number of nodes and arcs, density, clustering coefficient, etc.). In fact, the structure of a network is determined by the arcs existing between the corresponding nodes. In our case, nodes represent lemmas involved in comments and arcs represent features (i.e., co-occurrences or semantic relationships) playing a key role to define the semantics of the lemmas they link.

This reasoning is also reinforced by the fact that the definition of the semantics of a lemma is certainly improved by looking at the lemmas to which it is related in the network (in this claim, the extension, to the CS-Net model, of the homophily principle [435] characterizing social networks, comes into play).

The second factor is much more straighforward and concerns the semantic meaning of the concepts expressed by the network nodes, because each of them represents a lemma of the corresponding comments.

As for the first factor, in the literature there are many approaches designed for computing the similarity degree of the structural features of two networks (see [264, 72, 237], just to cite a few of them). We decided to adopt one of them and our choice fell on NetSimile [72]. In fact, this approach has a much shorter computation time than most of the other ones performing the same task proposed in the past literature. Furthermore, the accuracy level it guarantees is adequate for our application context. NetSimile extracts and evaluates the structural characteristics of each node based on the structural characteristics (such as the average clustering coefficient, the average number of nodes and arcs, etc.) of its ego network . As a consequence, in order to obtain the similarity score of two networks, NetSimile computes the similarity degree of their vectors of features.

As far as the second factor is concerned, we decided to consider the portion of nodes with the same meaning, or rather with similar meanings, present in the two subnetworks. A simple, but very effective, way to evaluate this portion could consist of the computation of the Jaccard coefficient between the sets of lemmas associated with the nodes of the two networks. Actually, to increase the result accuracy, it is necessary to take lexicographic relationships (e.g., synonymies and homonymies) [487, 196] between lemmas into account. As we mentioned above, these can be identified from an advanced dictionary, such as ConceptNet [405], which includes WordNet [442], a thesaurus widely used in the past literature for this purpose. In the following, we will adopt the symbol J^* to denote the Jaccard coefficient enhanced in such a way as to take lexicographic relationships into account.

We are now able to define the formula for computing σ_{12}^x. Specifically, we have:

$$\sigma_{12}^x = \beta^x \cdot \nu(\mathcal{N}_1^x, \mathcal{N}_2^x) + (1 - \beta^x) \cdot J^*(N_1^x, N_2^x)$$

Here:

- $\nu(\mathcal{N}_1^x, \mathcal{N}_2^x)$ is a function computing the topological similarity of $\mathcal{N}_1^x$ and $\mathcal{N}_2^x$ by applying the NetSimile approach.
- β^x is a coefficient defining the weight of the topological similarity of the networks with respect to the semantic similarity of the lemmas associated with the corresponding nodes. In order to define a formula for β^x, we made the following

reasoning. Intuitively, one can assume that the denser the networks, the more the information about their topology (and, thus, ν) becomes relevant. In other words, while the information contained in the nodes (expressed by J^*) does not vary against the density of the networks, the information contained in the arcs varies. In fact, a larger number of arcs implies an increase of the amount of information available, as well as of the strength of the relationships between the lemmas in the network. This is due to the fact that: *(i)* arcs represent semantic relationships existing between lemmas; *(ii)* for the homophily principle, a higher number of arcs implies, for each node, a higher number of neighbors that can contribute to better define the semantics of the lemma associated with it.

The above reasoning is at the basis of our formula for computing β^x. In order to define it, we need to introduce the concept of mean density of a set of CS-Nets. In fact, as will be clear in the following, the formula of β^x depends on whether the density of $\mathcal{N}_1^x$ and $\mathcal{N}_2^x$ is greater or less than the mean density $\overline{d^x}$ of the CS-Nets generally present in the reference context. In fact, we do not have a predefined set of CS-Nets on which we can operate, but these are derived from the subset $\overline{\mathcal{C}} \subseteq \mathcal{C}$ of the comments returned at the end of the comment filtering and text pattern extraction activities. Therefore, in order to compute the mean density $\overline{d^x}$, we built a set $\overline{\mathcal{CN}} = \langle \overline{\mathcal{N}_1}, \overline{\mathcal{N}_2}, \cdots, \overline{\mathcal{N}_t} \rangle$ of CS-Nets by deriving it randomly from the comments of $\overline{\mathcal{C}}$. The process of constructing $\overline{\mathcal{CN}}$ was as follows. First, we randomly constructed a set $\overline{\mathcal{CS}} = \langle \overline{\mathcal{C}_1}, \overline{\mathcal{C}_2}, \cdots, \overline{\mathcal{C}_t} \rangle$ of comment sets such that $\overline{\mathcal{C}_h} \subseteq \overline{\mathcal{C}}, 1 \leq h \leq t$. The randomness in the construction of $\overline{\mathcal{C}_h}$ involves both its cardinality and the lemmas comprising it. A CS-Net $\overline{\mathcal{N}_h} = \langle \overline{N_h}, \overline{A_h} = \overline{A_h^c} \cup \overline{A_h^r} \rangle$ can be constructed for each subset $\overline{\mathcal{C}_h}$, $1 \leq h \leq t$, by applying the approach described in Section 8.1.2. Let $\overline{\mathcal{N}_h^x} = \langle \overline{N_h^x}, \overline{A_h^x} \rangle$, $x \in \{c, r\}$, be the subnetworks of $\overline{\mathcal{N}_h}$ obtained by selecting only the arcs of type x. Let $\overline{\mathcal{CN}^x} = \langle \overline{\mathcal{N}_1^x}, \overline{\mathcal{N}_2^x}, \cdots, \overline{\mathcal{N}_t^x} \rangle$ be the set of subnetworks of type x.

The density $\overline{d_h^x}$ of $\overline{\mathcal{N}_h^x}$ is defined as:

$$\overline{d_h^x} = \frac{\overline{|A_h^x|}}{\frac{\overline{|N_h^x|} \cdot (\overline{|N_h^x|} - 1)}{2}}$$

The mean density of $\overline{\mathcal{CN}^x}$ is defined as:

$$\overline{d^x} = \frac{\sum_{h=1}^{t} \overline{d_h^x}}{t}$$

Consider now the subnetworks $\mathcal{N}_1^x$ and $\mathcal{N}_2^x$ of our interest. We define their average density d_{12}^x as:

$$d_{12}^x = \frac{d_1^x + d_2^x}{2}$$

where the formula to compute d_1^x and d_2^x is the same as the one presented above for $\overline{d_h^x}$.

At this point, we are able to define β^x. In particular, we have that:

$$\beta^x = \begin{cases} min\left(0.5 + \frac{d^x_{12} - \overline{d^x}}{\overline{d^x}}, \beta^x_{max}\right) & if\ d^x_{12} \geq \overline{d^x} \\ max\left(\beta^x_{min}, 0.5 - \frac{\overline{d^x} - d^x_{12}}{\overline{d^x}}\right) & if\ d^x_{12} < \overline{d^x} \end{cases}$$

This definition of β^x takes into account the reasoning expressed above regarding the correlation between the density of $\mathcal{N}_1^x$ and $\mathcal{N}_2^x$ and the importance of their topological components in the computation of σ^x_{12}. However, at the same time, it imposes that β^x can oscillate in a range between β^x_{min} and β^x_{max} (which we set at 0.25 and 0.75, respectively). This constraint allows the contribution of ν (resp., J^*) not to become irrelevant, in case the density is very low (resp., high).

Note that σ^x_{12} ranges in the real interval $[0,1]$. The higher σ^x_{12}, the greater the similarity of $\mathcal{N}_1^x$ and $\mathcal{N}_2^x$.

We will return to the choice of the values of β^x in Section 8.2.2, where we illustrate an experiment that we conducted about this issue.

We point out that our approach for computing σ^x_{12} is capable of operating on any projection $\mathcal{N}_1^x$ and $\mathcal{N}_2^x$ of the networks $\mathcal{N}_1$ and $\mathcal{N}_2$. The only constraint it imposes is that all arcs must be of the same type x. This helps making our overall approach scalable in that, if in the future we want to add an additional perspective of modeling content semantics, then the similarity degree of the corresponding projections of $\mathcal{N}_1$ and $\mathcal{N}_2$ can be still computed using it.

8.1.5 Dataset

As we mentioned in the Introduction, the set of comments on which we apply our approach should be homogeneous, e.g., related to a specific topic or a specific time of the year, or both. Following this guideline, we decided to focus on comments related to Not Safe For Work (hereafter, NSFW) posts in our experiments. This choice is also motivated by the fact that this topic has its intrinsic interest, regardless of our approach. Therefore, it has a double benefit, i.e., it allows us to test our approach and shed some light on a relevant phenomenon in Reddit, which is still little studied. Reddit is one of the few social networks to handle NSFW content in a straightforward and well-structured way. Despite this, only a few researchers have analyzed the phenomenon of NSFW content in this social platform [433, 457, 180].

In order to build our dataset of comments on NSFW posts, we used the website `pushshift.io` [65], which represents one of the main data repositories for Reddit. Specifically, we considered 449 NSFW adult subreddits listed at the address `https://www.reddit.com/r/ListOfSubreddits/wiki/nsfw` and downloaded comments

to all posts published from January 1^{st}, 2020 to March 31^{st}, 2020. The number of posts considered is 3,064,758, while the total number of comments is 11,627,372.

We performed an ETL (Extraction, Transformation, and Loading) activity on this data. During it, we observed that some of the posts downloaded from `pushshift.io` were published by authors who had left Reddit. We decided to remove these posts and the associated comments from our dataset. Moreover, we removed all the comments related to posts whose field `over_18` was set to `false`. After this ETL activity, the total number of NSFW posts in our dataset is 2,981,601, corresponding to 97% of the initial ones. The total number of NSFW comments present in our dataset is 8,383,499, corresponding to 72.20% of the initial ones.

In Table 8.1, we report some information about the authors of posts and comments. We can see that the number of authors who wrote comments is much larger than the number of authors who published posts. In addition, we can observe that half of the authors who published posts also published comments.

Parameter	*January 2020*	*February 2020*	*March 2020*	*Total*
Authors publishing posts	91,894	92,530	110,873	218,433
Authors publishing comments	369,014	351,967	392,871	738,216
Authors publishing both posts and comments	46,427	44,733	53,063	115,686

Table 8.1: Some parameters regarding authors in the dataset

Figure 8.1 shows the distributions of comments against posts, while Figure 8.2 reports the distribution of scores against comments. As can be seen from these figures, both distributions follow a power law.

In Table 8.2, we report the values of the coefficients α and δ, along with the minimum and maximum values, relative to these distributions.

Field	α	δ	*Minimum Value*	*Maximum Value*
Comments against posts	3.0821	0.0159	1	1,462
Scores against comments (left*)	3.8485	0.0255	-521	0
Scores against comments (right)	2.1456	0.0158	1	3,425

Table 8.2: Values of α and δ, minimum and maximum values of the distributions of interests for the dataset - *The values of α and δ for the left part of the distribution of scores against comments were computed considering the absolute values of scores

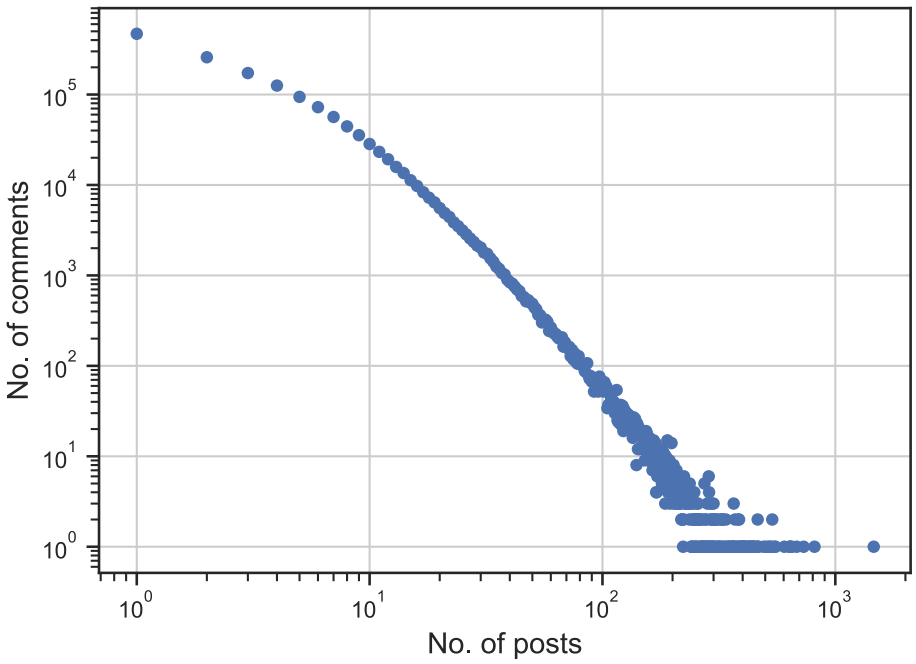


Fig. 8.1: Distributions of comments against posts

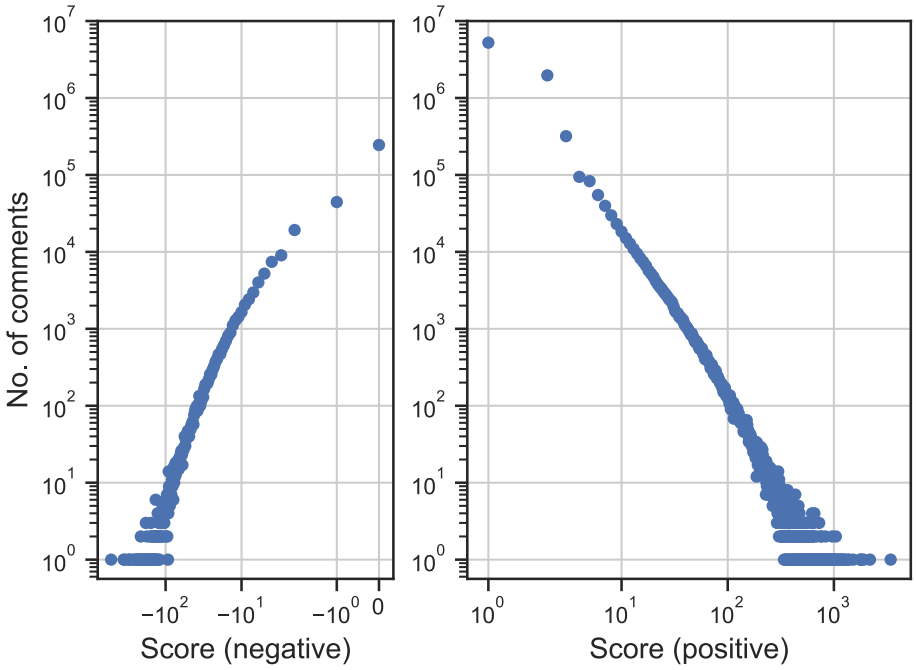


Fig. 8.2: Distributions of scores against comments

8.2 Results

8.2.1 Analysis of generated Content Semantic Network

We have seen that our approach extracts text patterns from which it constructs the CS-Nets to analyze. The text pattern detection approaches proposed in the past lit-

erature aim at selecting the most frequent patterns. In addition to the frequency of patterns, our approach takes into account their utility, expressed by a utility function, and selects the patterns with the highest values of this function. However, the main focus of our approach is content semantics. So, it is extremely important to verify whether, besides extracting the most frequent and useful comments, it is able to build CS-Nets having a homogeneous and meaningful semantics.

Recall that, in our approach, semantic links between lemmas are expressed by means of arcs connecting the corresponding nodes. Therefore, we can say that the greater the number of arcs we observe in the generated CS-Nets, the greater the number of semantic links between the corresponding lemmas. Moreover, the greater the number of such links, the greater the semantic significance of the CS-Net and, ultimately, the better the quality of our approach.

To test whether our approach is capable of constructing semantically meaningful CS-Nets from a set of comments, we planned to compare it with an approach that builds the networks randomly and can serve as a null model in a significance test. To this end, we considered four sets of comments $\mathcal{C}_1, \cdots, \mathcal{C}_4$. For each set, we initially applied our approach and constructed the CS-Nets $\mathcal{N}_1 = \langle N_1, A_1 = A_1^c \cup A_1^r \rangle, \cdots, \mathcal{N}_4 = \langle N_4, A_4 = A_4^c \cup A_4^r \rangle$. Next, we applied the random approach with the goal of constructing the CS-Nets $\overline{\mathcal{N}_1} = \langle \overline{N_1}, \overline{A_1} = \overline{A_1^c} \cup \overline{A_1^r} \rangle, \cdots, \overline{\mathcal{N}_4} = \langle \overline{N_4}, \overline{A_4} = \overline{A_4^c} \cup \overline{A_4^r} \rangle$.

In particular, given the set $\mathcal{C}_k$ of comments, to construct the corresponding CS-Net $\overline{\mathcal{N}_k}$, we selected uniformly at random a number of lemmas from $\mathcal{C}_k$ equal to the cardinality of N_k, such that $|\overline{N_k}| = |N_k|$. In this way, $\mathcal{N}_k$ and $\overline{\mathcal{N}_k}$ had the same number of nodes. Then, we constructed $\overline{A_k}$ as follows: given two nodes $n_i \in \overline{N_k}$ and $n_j \in \overline{N_k}$, we inserted an arc $a_{ij}^c \in \overline{A_k^c}$ if the lemmas l_i and l_j, corresponding to n_i and n_j, were simultaneously present in at least one comment of $\mathcal{C}_k$. In addition, we inserted an arc $a_{ij}^r \in \overline{A_k^r}$ if there is a semantic relationship between l_i and l_j in ConceptNet.

For each set $\mathcal{C}_k$ of comments, we performed the random approach described above 30 times. Finally, we computed the number of arcs of A_k obtained through our approach (applying the two different utility functions $f_s(\cdot)$ and $f_p(\cdot)$) and the mean of the number of the arcs of $\overline{A_k}$ obtained by averaging the number of arcs of the 30 CS-Nets $\overline{\mathcal{N}_k}$ built by applying the random approach. These numbers are shown in Table 8.3.

From the analysis of this table, we can observe that, in all cases, our approach returns CS-Nets with a higher number of arcs than the random one.

To assess the significance of this result, we performed the t-test between the outputs of our approach (with the two different utility functions) and those obtained from the null model. At the end of this task, we computed the corresponding p-values. They are reported in Table 8.4.

Sets of comments	*Number of nodes* of $\mathcal{N}_k$ and $\overline{\mathcal{N}_k}$	*Number of arcs of* $\mathcal{N}_k$ *Utility function:* $f_s(.)$	*Number of arcs of* $\mathcal{N}_k$ *Utility function:* $f_p(.)$	*Number of arcs of* $\overline{\mathcal{N}_k}$
$\mathcal{C}_1$	98	2,351.14	2,116.26	1,587.21
$\mathcal{C}_2$	111	3,191.85	2,872.66	1,834.77
$\mathcal{C}_3$	103	2,400.97	2,160.87	1,798.34
$\mathcal{C}_4$	105	2,527.42	2,274.68	1,311.77

Table 8.3: Average number of arcs of the CS-Nets generated by applying our approach, with two different utility functions, and the random one

Sets of comments	$f_s(.)$	$f_p(.)$
$\mathcal{C}_1$	$8.90 \cdot 10^{-25}$	$8.59 \cdot 10^{-20}$
$\mathcal{C}_2$	$4.51 \cdot 10^{-21}$	$7.81 \cdot 10^{-18}$
$\mathcal{C}_3$	$2.40 \cdot 10^{-14}$	$8.59 \cdot 10^{-20}$
$\mathcal{C}_4$	$3.07 \cdot 10^{-15}$	$5.42 \cdot 10^{-19}$

Table 8.4: p-values obtained by performing the t-test between the outputs of our approach and those returned by the null model

From the analysis of this table, we can observe that, with both utility functions, the p-values are very low, much lower than 0.05. This result leads us to conclude that our approach actually returns CS-Nets with a larger number of arcs, and therefore semantically more homogeneous and meaningful.

Based on what we said at the beginning of this section, this result is very encouraging because it says that our approach not only selects very frequent and useful patterns but also builds high-quality CS-Nets from the content semantics point of view.

We described above our experiment for four sets of comments $\mathcal{C}_1, \cdots, \mathcal{C}_4$. After obtaining the results described in Table 8.4, we repeated it with 50 other sets of comments and obtained similar results. Due to space constraints, we cannot report here their details.

8.2.2 Investigating β^x

In Section 8.1.4, we have seen that the semantic similarity degree σ_{12}^x between two subnetworks $\mathcal{N}_1^x$ and $\mathcal{N}_2^x$, obtained from $\mathcal{N}_1$ and $\mathcal{N}_2$ considering only arcs of type x, with $x \in \{c, r\}$, depends on a coefficient β^x. This defines the weight of the topological similarity of the networks with respect to the semantic similarity of the lemmas associated with the corresponding nodes. In the same section, we have also defined a formula for β^x and we have seen that it is essentially related to the density of $\mathcal{N}_1^x$ and $\mathcal{N}_2^x$.

In this experiment, we aim at performing some analyses on the trend of the value of β^x against the number of nodes of $\mathcal{N}_1^x$ and $\mathcal{N}_2^x$. For this purpose, we performed the following tasks:

- We considered 50 sets of comments of different sizes.
- We performed the activities described in Sections 8.1.1, 8.1.2 and 8.1.3 on each set, and obtained 50 CS-Nets of different sizes.
- We considered all possible pairs $(\mathcal{N}_1, \mathcal{N}_2)$ of CS-Nets that could be constructed from the initial 50 networks.
- For each pair $(\mathcal{N}_1, \mathcal{N}_2)$ of CS-Nets, we generated two pairs of subnetworks $(\mathcal{N}_1^c, \mathcal{N}_2^c)$ and $(\mathcal{N}_1^r, \mathcal{N}_2^r)$.
- For each pair $(\mathcal{N}_1^x, \mathcal{N}_2^x)$ of subnetworks, $x \in \{c, r\}$, we computed both $|N_1^x| + |N_2^x|$ and β^x. In the following, we call ρ^x the parameter $|N_1^x| + |N_2^x|$.
- We constructed 30 bins of values of ρ^x; specifically, the first bin groups all values of ρ^x between 1 and 10, the second bin includes all values of ρ^x between 11 and 20, and so on. The last bin comprises all values of ρ^x between 291 and 300.
- We assigned each pair $(\mathcal{N}_1^x, \mathcal{N}_2^x)$ of subnetworks to the suitable bin, based on the corresponding value of ρ^x.
- For each bin, we computed the mean value of β^x by averaging the values of β^x of all the pairs of subnetworks assigned to it.

We report the results obtained in the histogram of Figure 8.3.

Observe that this histogram starts from the range $[90, 100]$ of ρ^x because no pairs of networks fall in lower bins. From the analysis of this figure, we can observe that, as ρ^x increases, the mean value of β^x decreases, although this trend is gradual. From the graph theory point of view, this can be explained by considering that there is a direct proportionality relationship between β^x, on one side, and d_1^x and d_2^x, on the other side. Now, as ρ^x increases, the denominators of d_1^x and d_2^x grow according to a quadratic trend, while their numerators grow at most with a quadratic trend, but generally with a trend between linear and quadratic. This tendency for the numerators to grow less than the denominators is reflected in the trend of d_1^x and d_2^x against ρ^x and, consequently, in the trend of β^x against the same parameter. From our analyses viewpoint, this implies that, as ρ^x increases, the importance of the semantic similarity against the topological similarity increases too. This is justified taking into account that, as ρ^x increases, the number of lemmas available to define each network increases as well, and therefore the possibility to better define the semantics expressed by these last ones grows. This semantics is certainly richer than the one that can be defined through the simple topological analysis of the network.

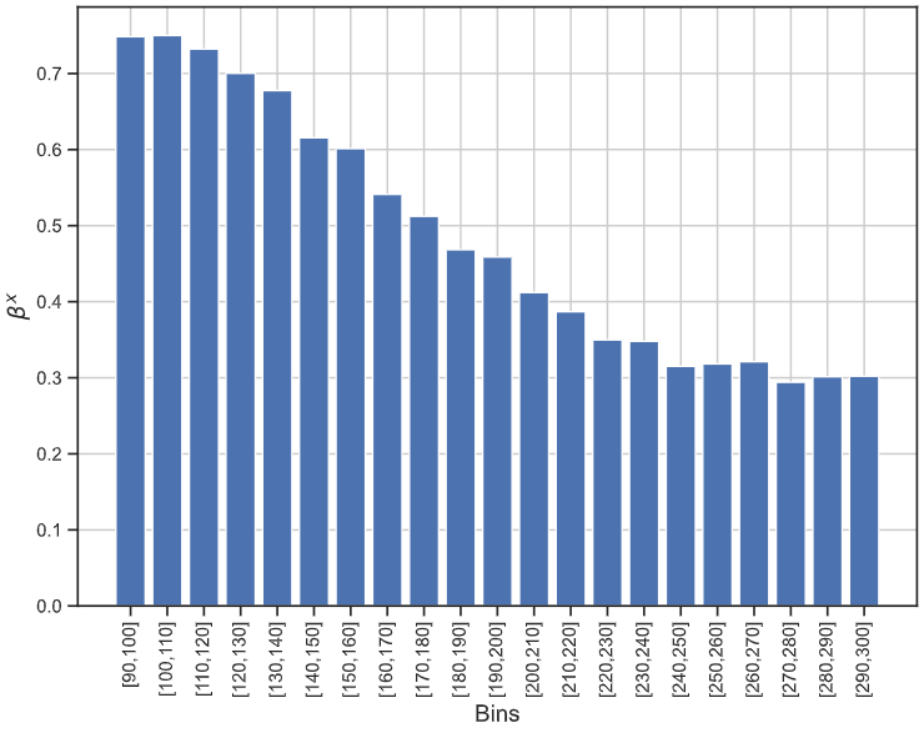


Fig. 8.3: Mean values of β^x against values of $\rho^x = |N_1^x| + |N_2^x|$

8.2.3 Investigating α

In Section 8.1.3, we have seen that the semantic similarity degree σ_{12} between two subnetworks $\mathcal{N}_1$ and $\mathcal{N}_2$ depends on the coefficient α. This defines the weight of the semantic similarity expressed by co-occurrences against the one expressed through the semantic relationships between lemmas. In the same section, we have defined a formula for α and we have seen that it is substantially related to the values of $|A_1^c|$, $|A_1^r|$, $|A_2^c|$ and $|A_2^r|$.

In this experiment, we aim at performing some analyses on the trend of the value of α against the variation of the four parameters above. To this end, we have carried out the following tasks:

- We considered 50 sets of comments of different sizes.
- We performed the activities described in Sections 8.1.1, 8.1.2 and 8.1.3 on each set and obtained 50 CS-Nets of different sizes.
- We considered all possible pairs $(\mathcal{N}_1, \mathcal{N}_2)$ of CS-Nets that could be constructed from the initial 50 CS-Nets.
- For each pair $(\mathcal{N}_1, \mathcal{N}_2)$ of CS-Nets, we computed the value of the parameter $\phi = |N_1| + |N_2|$ (i.e., the overall number of nodes of $\mathcal{N}_1$ and $\mathcal{N}_2$) and the value of α.

- We constructed 30 bins of values of ϕ; specifically, the first bin groups all values of ϕ between 1 and 10, the second bin includes all values of ϕ between 11 and 20, and so on. The last bin comprises all values of ϕ between 291 and 300.
- We assigned each pair $(\mathcal{N}_1, \mathcal{N}_2)$ of subnetworks to the suitable bin, based on the corresponding value of ϕ.
- For each bin, we computed the mean value of α by averaging the values of α of all the pairs of CS-Nets assigned to it.

We report the results obtained in the histogram of Figure 8.4. Analogously to what happens for β^x, the first bins are not present in the histogram because there was no pair of CS-Nets belonging to them.

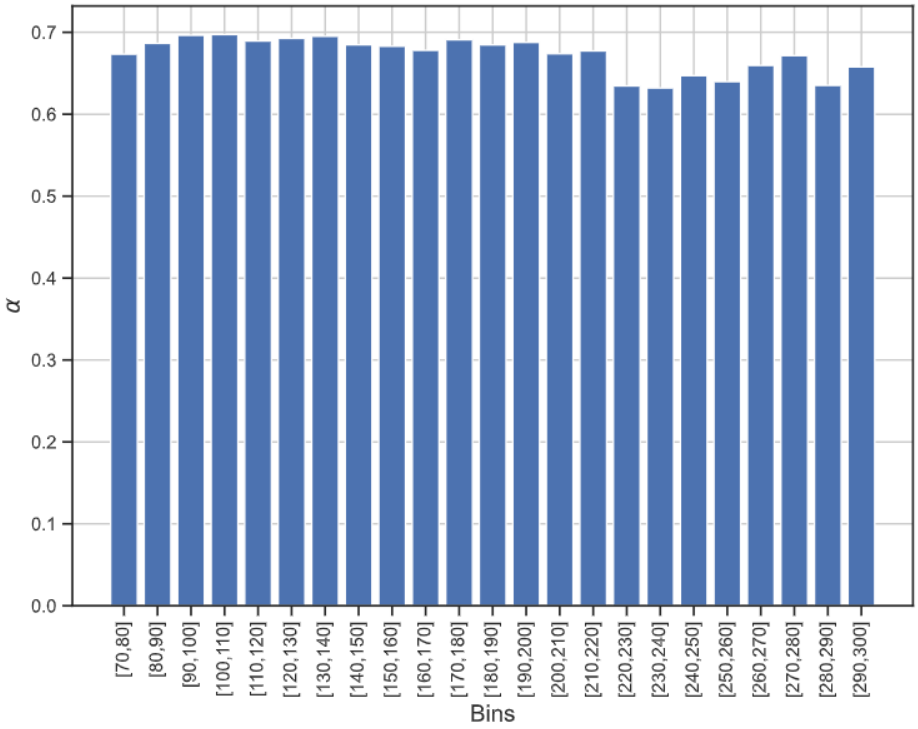


Fig. 8.4: Mean values of α against values of $\phi = |N_1| + |N_2|$

From the analysis of this figure, we can observe no specific trend in the values of α against ϕ. This can be explained by considering that, as $|N_1|$ and $|N_2|$ grow, it is presumable that $|A_1^c|$ and $|A_2^c|$ on the one hand, and $|A_1^r|$ and $|A_2^r|$ on the other hand, will also grow. The value of α depends on how fast these values grow. Specifically, if $|A_1^c|$ and $|A_2^c|$ grow faster than $|A_1^r|$ and $|A_2^r|$ then α increases; in the opposite case, α decreases. However, this fact is totally independent of the growth of $|N_1|$ and $|N_2|$, because it depends exclusively on the number of co-occurrences of the nodes in the text patterns, on the one hand, and the number of semantic relationships between

the lemmas corresponding to the nodes, on the other hand. In any case, there is a constant element to observe in Figure 8.4 and it concerns the fact that α is always between 0.6 and 0.7. This means that, in the computation of σ_{12}, the component expressing the co-occurrences of lemmas has a higher weight than the one representing the semantic relationships between them. This is reasonable if we consider that the component related to co-occurrences expresses the semantics derived from the dynamic and real use of the lemmas in the comments, while the component related to semantic relationships expresses the semantics as theoretically provided by the language adopted. However, the formula for the computation of α has been defined in such a way that if, in an application scenario, we have more semantic relationships and much less co-occurrences between lemmas, the weights of the two components are automatically inverted.

Thus, as for the variation of their values against the size of the involved (sub)networks, the parameters α and β^x show a completely different behavior.

8.2.4 Extracting knowledge from a real world scenario

This latest experiment is intended as a demonstration of the potentialities of our approach in a real world scenario. At the same time, it represents a bridge between the previous subsections, dedicated to experiments, and the next section, concerning applications. In particular, having a Reddit dataset at our disposal, we thought to evaluate, given a user following one or more subreddits, the ability of our approach to recommend new subreddits potentially interesting for her. In this case, our approach would behave as the engine of a content-based recommender system.

The steps of a recommender system employing our approach as an engine and suggesting to a user u new subreddits to join are the following:

1. Consider the set $\mathcal{C}_u$ of comments that u posted in the past.
2. Apply the first two steps of our approach to construct the CS-Net $\mathcal{N}_u$ associated with $\mathcal{C}_u$.
3. Consider a set $SSet$ of subreddits not yet accessed by u; the subreddits of $SSet$ could be chosen based on parameters like their creation date (favoring the most recent ones), the number of users already accessing them, the number of posts and comments already published in them, etc.
4. For each subreddit $S_l \in SSet$, let $\mathcal{C}_l$ be the set of its comments.
   4.1. For each $\mathcal{C}_l$, apply the first two steps of our approach to construct the CS-Net $\mathcal{N}_l$ corresponding to it.
   4.2. For each $\mathcal{N}_l$, apply the third step of our approach to compute the semantic similarity degree σ_l between $\mathcal{N}_l$ and $\mathcal{N}_u$.

5. Sort the values of σ_l thus obtained in a descending order.
6. Recommend to u the top k subreddits of the list. The value of k can be chosen based on several parameters, such as the seniority of u on Reddit, the number of subreddits u is currently accessing, her activity level on Reddit, etc.

We point out that, albeit we presented the previous algorithm with reference to Reddit, it could be applied to several other social networks (such as Facebook and Twitter) with very few changes.

As it is clear from the previous steps, as well as from the way of proceeding of our approach, which is the engine of the recommender system we are describing, the presence of a large set of comments from the user to whom we want to provide recommendations plays a key role on the quality of the results that can be obtained. On the other hand, this is a typical feature of any content-based recommender system. As a consequence, in performing this experiment, we decided to filter out users with few comments. To this end, we computed the distribution of users against comments. It is shown in Figure 8.5. From the analysis of this figure, we can observe that, even if this distribution does not follow a perfect power law, there are in any case many users posting few comments and few users posting many comments. In our experiment, we judged a number of comments less than 20 as not significant for tracking the interests of a user. Therefore, we selected only users who published more than 20 comments.

To evaluate the performance of the recommender system described above, we borrowed the concepts of *true label* and *Top-k Accuracy* from Machine Learning. Specifically, in the classification task, a true label represents the assignment of a correct class to an observation, while a false label corresponds to a misclassification. The Top-k Accuracy considers the k predictions of a model having the highest probability. If one of them corresponds to a true label, it considers the prediction as correct; otherwise, it considers the prediction as incorrect. Note that the classical concept of accuracy corresponds to a special case of the Top-k Accuracy one, with $k = 1$. Given the complexity of our scenario, in which two or more subreddits could be related to the same topic, and given the huge number of text patterns that could be extracted from a set of comments, we judged that Top-1 Accuracy was a too rigid metric to evaluate the performance of our recommender system and, for this reason, we decided to adopt Top-k Accuracy, with $1 \leq k \leq 5$. We point out that we chose the maximum value of k empirically. In particular, we observed that the values of k we selected allowed us to obtain the maximum set of subreddits reflecting the scenario of interest. Indeed, as shown in Figure 8.6, larger values of k do not lead to an improvement in the hit ratio.

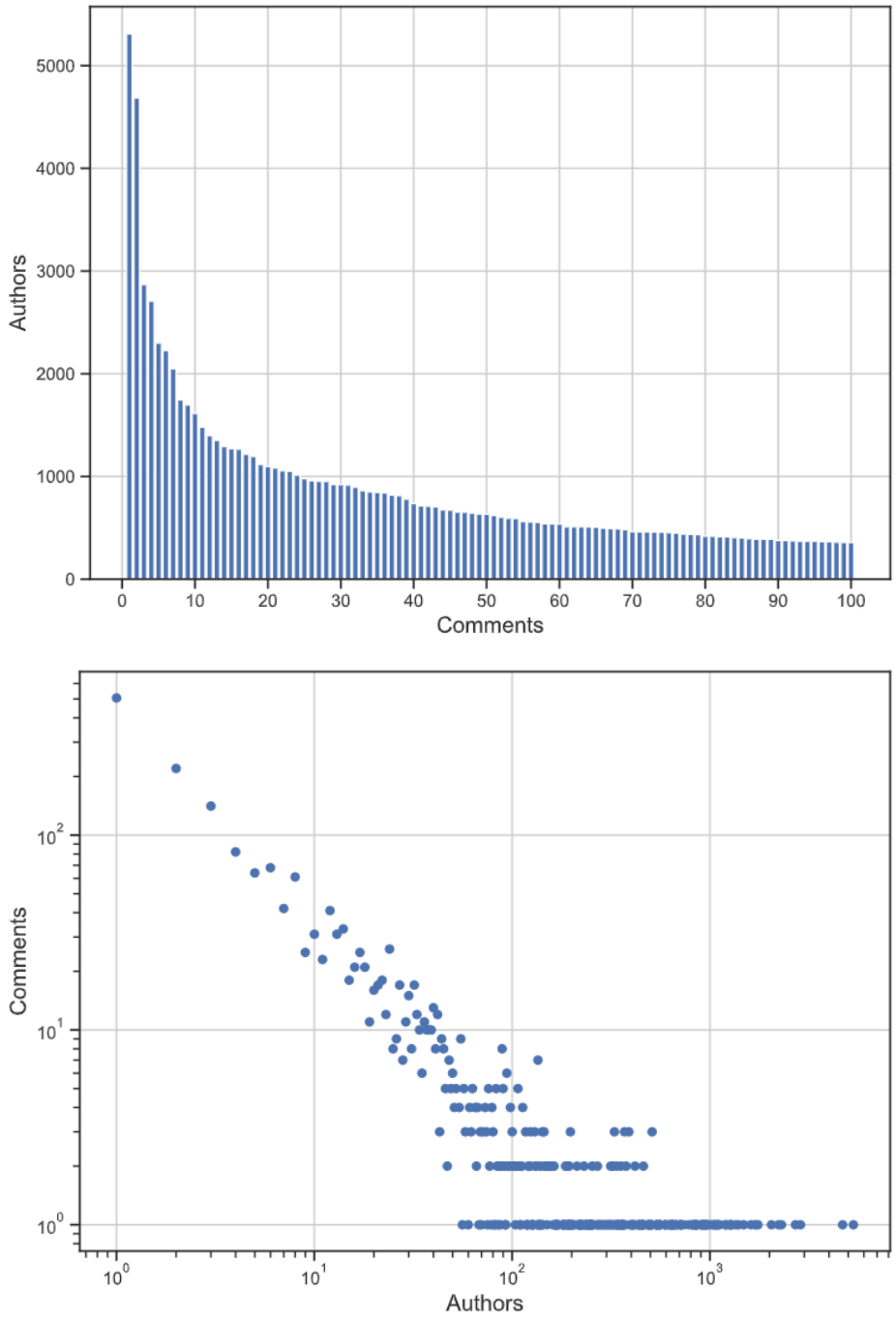


Fig. 8.5: Distribution of authors against comments on linear scale (top) and log-log scale (bottom)

In order to define the true label of a user, we relied on the homophily principle of Social Network Analysis [435] and made the following assumption: *"the subreddits closest to a specific user are those where she writes the most comments"*. In fact, if a user often visits a subreddit and writes many comments in it, then it means that the topics

discussed therein are of her interest. This also means that the patterns characterizing her profile are similar to those used in that subreddit.

Similarly to what we have seen for Top-k Accuracy, we considered that assuming the presence of only one true label for a user is a too rigid hypothesis for the reference context. Therefore, we decided to assume that, for each user, h true labels are possible, $1 \leq h \leq 3$ and these are the h subreddits in which she posted the highest number of comments. Clearly, the comments in these h subreddits were not used to build $\mathcal{N}_u$. Similarly to the range of k, we set the range of h empirically. In particular, for larger values of h, we did not observe significant variations in the hit ratio value against k, as shown in Figure 8.6.

Having defined how to proceed and the metrics used in our experiment, we are now able to illustrate how we conducted it. Specifically, we considered all users who posted more than 20 comments. Let u be one of these users and let $\mathcal{N}_u$ be the corresponding CS-Net. We ran our recommendation algorithm for her and computed the k subreddits whose corresponding CS-Nets have the top-k similarity degree with $\mathcal{N}_u$. If at least one of the k subreddits is present in the h true labels of u, we considered the whole prediction as a "hit"; otherwise, we categorized it as a "miss".

In Figure 8.6, we report the hit ratio, averaged over all users publishing more than 20 eligible comments (i.e., different from those used to build the corresponding profiles), with the values of k ranging from 1 to 5 and the values of h ranging from 1 to 3.

From the analysis of this figure we can see that our recommendation algorithm works very well in many cases. The results are already promising for $h = 1$ (although this is a very stringent condition for the reasons outlined above) as long as the value of k is greater than or equal to 3. However, we argue that the scenarios best representing the reference context are those with $h \geq 2$ and $k \geq 3$. In this case, the results we obtain are really satisfactory in that the average hit ratio ranges from 81.31% (for $h = 2$ and $k = 3$) to 93.46% (for $h = 3$ and $k = 5$).

In this experiment, we used the past data at our disposal, in particular the subreddits already frequented by users, as a test set to evaluate the performance of our recommendation algorithm. It is clear that, in a real world scenario, our recommendation algorithm would not be employed to suggest a user the subreddits that she is already following. Rather, it will be adopted to suggest her subreddits she is not aware of and appearing close to her interests, based on her past behavior.

We end this section pointing out that the issue of recommending a subreddit (and, more generally, a community) to a user in a social network goes far beyond what we have seen in this experiment. In fact, it represents one of the most investigated application issues in the Social Network Analysis literature. Also for this

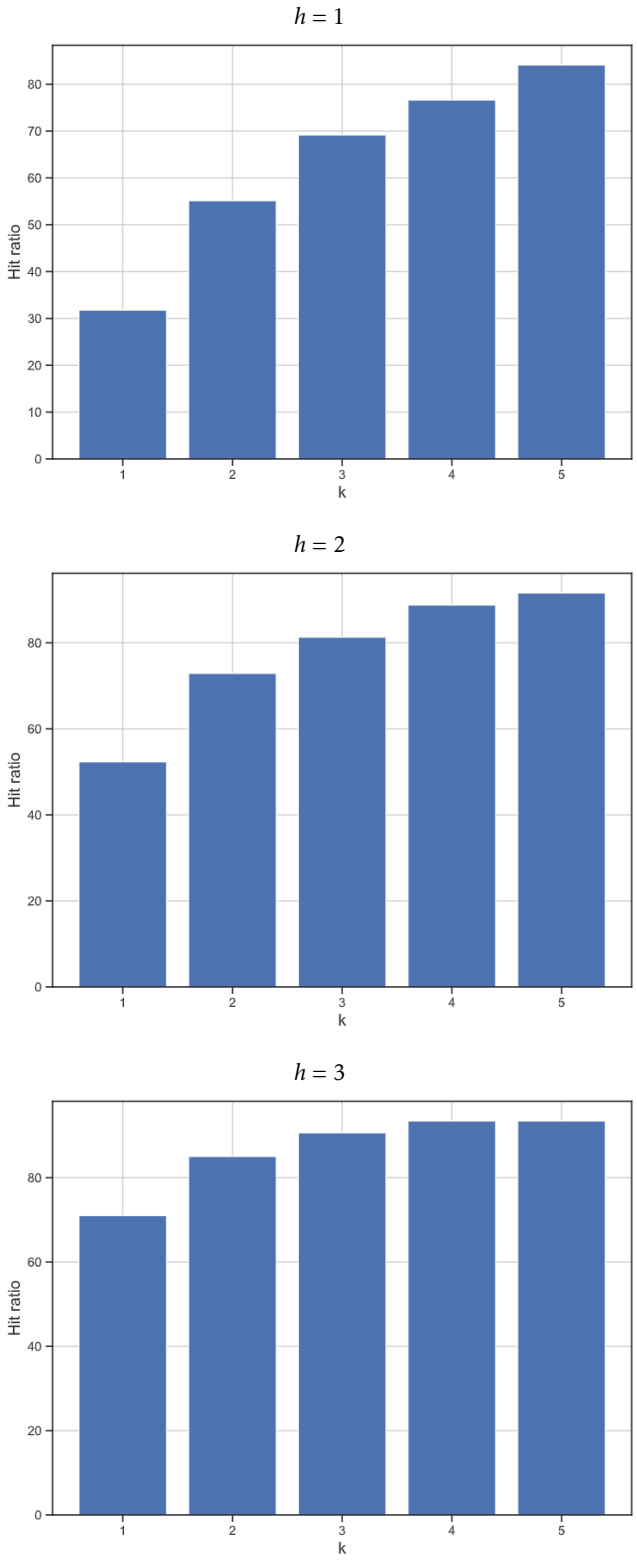


Fig. 8.6: Hit ratio with different values of h and k.

reason, we return to this issue in the next section, where we show how our approach can provide an interesting contribution in this setting.

8.3 Possible Applications

As we mentioned in the previous sections, our approach is general in the sense that it proposes: *(i)* a data model capable of representing and handling a set of comments, regardless of their source; *(ii)* a technique to filter comments based on both their frequency and their utility; *(iii)* a technique to construct a CS-Net for each set of filtered comments; *(iv)* a technique to evaluate the semantic similarity of two CS-Nets.

As a consequence, it may have various applications depending on the origin of comments. In this section, we mention some of them while pointing out that several others can be thought once one or more sets of comments of interest for a given scenario have been identified. Before starting this examination, we would like to point out that the objective of this section is not to fully and thoroughly define the various applications with all their technical details. This study, accompanied by the corresponding tests aimed at highlighting the applications' correctness and performance, will be the subject of future work. Our goal now is showing that the approach defined in this paper might be exploited in various application scenarios.

8.3.1 Content-based recommender systems

Let u_1 be a user and let $\mathcal{C}_1$ be a set of lemmatized comments that she expressed in a past time interval. The length of the time interval can be arbitrarily defined taking into account that the further back in the past we go, the richer $\mathcal{C}_1$ could be, but, at the same time, the higher the risk that it includes topics no longer of interest to u_1. Starting from $\mathcal{C}_1$, a set $\mathcal{P}_1$ of patterns can be derived by applying the techniques explained in Section 8.1.1. Once $\mathcal{P}_1$ has been constructed, it is possible to build a CS-Net $\mathcal{N}_1$ that indicates the interest of u_1 based on the comments she made in the past. Specifically:

$$\mathcal{N}_1 = \langle N_1, A_1^c \cup A_1^r \rangle$$

N_1 is the set of nodes of $\mathcal{N}_1$. There is a node $n_i \in N_1$ for each lemma l_i present in at least one pattern of $\mathcal{P}_1$. An arc $(n_i, n_j, w_{ij}) \in A_1^c$ indicates that the lemmas l_i and l_j occur together in at least one pattern of $\mathcal{P}_1$; w_{ij} depends on the number of patterns of $\mathcal{P}_1$ in which l_i and l_j occur together. An arc $(n_i, n_j, w_{ij}) \in A_1^r$ denotes that there is a form of semantic relationship between l_i and l_j; according to what we said about this issue in Section 8.1.2, w_{ij} denotes the strength of that relationship.

Similarly, let $\mathcal{C}_2$ be a second set of lemmatized comments associated with a set $PSet_2$ of posts or a subreddit S_2, which u_1 has not commented yet, e.g., because she does not know of its existence. Starting from $\mathcal{C}_2$, it is possible to construct a set $\mathcal{P}_2$

of patterns, by applying the techniques explained in Section 8.1.1, and a CS-Net $\mathcal{N}_2$ corresponding to $\mathcal{C}_2$. The structure and semantics of $\mathcal{N}_2$ are similar to those of $\mathcal{N}_1$.

At this point, by applying the technique expressed in Section 8.1.3, it is possible to compute a coefficient σ_{12} that indicates the semantic similarity between $\mathcal{N}_1$ and $\mathcal{N}_2$. If this similarity is high, we can conclude that the set $PSet_2$ of posts or the subreddit S_2 may be of interest to u_1 and, thus, may be recommended to her. In this way, we can implement a content-based recommender system that can suggest new posts or subreddits to u_1 based on her past history.

8.3.2 Collaborative filtering recommender systems

Let u_1 be a user and let $\mathcal{C}_1$ be the set of lemmatized comments that she expressed in a past time interval. Let $USet$ be a set of users about whom we make no assumptions. Let u_h be a user of $USet$, let $\mathcal{C}_h$ be the set of lemmatized comments she expressed in the same time interval considered for u_1. Applying the same reasoning seen in the previous subsection, we can build a CS-Net $\mathcal{N}_1$ that represents the profile of u_1 and a CS-Net $\mathcal{N}_h$ for each user $u_h \in USet$.

At this point, it is possible to compute the similarity coefficients between u_1 and each user $u_h \in USet$ by computing the corresponding similarity between $\mathcal{N}_1$ and $\mathcal{N}_2$. Having these coefficients at disposal, it is possible to apply a k-Nearest-Neighborhood approach to identify the set $\overline{USet}$ of users with interests most similar to those of u_1. Thanks to the homophily principle of Social Network Analysis [435], it is possible to assume that the posts and subreddits of interest to users of $\overline{USet}$ are also of interest to u_1. As a consequence, if u_1 does not already know them, they can be recommended to her.

In this way, we have realized a collaborative filtering recommender system that can suggest new posts or subreddits to u_1, based on the behavior of users with interests similar to her.

8.3.3 Building new user communities and/or identifying outliers

Let $USet$ be a set of users on whom we make no initial assumption about their membership in specific communities or about the similarity of their interests. Let u_h be a user of $USet$ and let $\mathcal{C}_h$ be the set of lemmatized comments she expressed in a past time interval. As for the length of this interval, the same considerations seen in Section 8.3.1 can be applied. Performing the procedure seen in that section, we can construct a CS-Net $\mathcal{N}_h$, which represents the interests of u_h as they emerge from $\mathcal{C}_h$.

At this point, for each pair of users u_1 and u_2 belonging to $USet$, we can compute the semantic similarity coefficient σ_{12} by applying the procedure described in

Section 8.1.3. The knowledge of this coefficient for each possible pair of users of $USet$ gives us the possibility to apply on the users of $USet$ one clustering algorithm among those existing in literature, e.g. DBSCAN [294] that provides very accurate results and allows us to identify outliers. The clusters thus defined allow us to build virtual communities of users (one for each cluster) characterized by similar topics. In Reddit, they could be exploited to build new subreddits.

Furthermore, the outliers thus identified would correspond to users with interests very far from those of the other ones. They could become the "seeds" for new communities dealing with issues different from those already existing (for instance, extremely innovative issues). In other circumstances, the detection of outliers could allow the discovery of users with illegal interests (e.g., fanatics, terrorists, etc.) to be reported to the police.

8.3.4 Building new subreddits and/or identifying outliers

Let $SSet$ be a set of subreddits on which we make no initial assumptions about the similarity of the interests of the users joining them. Let S_h be a subreddit of $SSet$, let $PSet_h$ be the set of its posts and let $\mathcal{C}_{h_k}$ be the set of comments corresponding to the post $p_{h_k} \in PSet_h$. Applying a procedure similar to the one seen for the users of $USet$ in Section 8.3.3, we can construct a CS-Net $\mathcal{N}_{h_k}$ that represents the interests of people involved in p_{h_k} as they emerge from their past comments.

At this point, we can apply the approach described in the previous section to the resulting CS-Nets. In this way, we can identify clusters of posts (perhaps belonging to different real subreddits) with similar topics. These posts can be grouped into homogeneous virtual subreddits obtained from the real ones. Each virtual subreddit thus obtained can be recommended to each user who had accessed at least one post included in it. In this way, the information, knowledge and opinion exchange between users belonging to different real communities and having similar interests are favored. These users can look very favorably and enthusiastically at this cross-contamination process.

Last, but not the least, the presence of outliers is an indicator of the existence of posts with contents very different from those of the others. These posts could become the seeds of new subreddits, similarly to what we have seen in the previous subsection.

9

Defining user spectra to classify user behaviors in cryptocurrencies

The classification of users of a blockchain is currently a highly investigated research topic with many possible applications. In the literature, some of the proposed approaches to perform this task are tailored to specific blockchains (e.g., Bitcoin, Ethereum, etc.), while other ones consider only the behavior of the individual user. In both cases, it is extremely important to evaluate user interactions, because they can unveil interesting patterns and provide new features to classify them. To the best of our knowledge, few approaches take these interactions into account, and none of them uses a suitable structure (like a multivariate time series) to finely represent the user behavior over time. In this chapter, we provide a contribution in this setting and present an automatic social network-based approach for classifying blockchain users based on their past behavior. Given a time period, our approach associates each user with a spectrum showing the trend of some behavioral features obtained from the social network representation of the whole blockchain. Each class of users has its own spectrum, obtained by averaging the spectra of its users. In order to evaluate the similarity between the spectrum of a class and the one of a user, we propose a tailored similarity measure obtained by adapting to this context some general measures proposed in the past. Finally, we test our approach on a dataset derived from the Ethereum blockchain.

The material presented in this chapter was derived from [254].

9.1 Methods

9.1.1 Proposed method

In this section, we present our approach. As mentioned in the Introduction, it consists of several steps, each introducing innovations with respect to the corresponding tasks proposed in the past. More specifically, the outline of our approach is as follows:

1. Construction of a social network supporting the representation of a training set concerning Ethereum users and their behavior.

2. Construction of the spectrum of users of the training set from their data stored in the dataset and some metrics computed on the social network built at Step 1.
3. Selection of the classes of interest. These are presumably the ones most prevalent in the dataset and, thus, in Ethereum. However, if we want to focus on one or more uncommon classes (e.g., for studying an outlier class), we can do it.
4. Construction of the spectrum of each class selected at Step 3 starting from the spectra of the users of the training set associated with that class.
5. Definition of a new version of the Eros distance tailored to our scenario and computation of the corresponding weights starting from the dataset.
6. For each user to be classified (whether she belongs to the test set or is a new user of whom nothing is known):
   a) Construction of the corresponding spectrum.
   b) Computation of the Eros distance between the spectrum built at Step 6(a) and the class spectra built at Step 4.
   c) Assignment of the user to the nearest class according to the values of the Eros distance computed at Step 6(b). Otherwise, assignment of the user to no class if the Eros distance between her spectrum and that of all available classes is higher than a certain threshold.

In the next subsections, we describe the various steps of our approach in detail.

9.1.2 Modeling a blockchain as a social network

A blockchain can be modeled through a social network in a very direct way. In fact, the social network nodes can represent the blockchain addresses, while its arcs can denote the transactions between the addresses corresponding to the involved nodes. The capability of building such a model for a blockchain leads to the possibility of extracting knowledge about the behavior of blockchain actors by employing the Social Network Analysis based techniques proposed in the past [361, 224, 417]. In the following, we show this property taking Ethereum as the reference blockchain because it is the blockchain of interest for this paper. However, we point out again that our approach to build and characterize a social network from a blockchain (and, consequently, the next classification approach representing the core of this paper) can be applied to most blockchains. Indeed, the features used to model Ethereum as a social network (such as the sender, receiver and timestamp of a transaction, and the amount of transferred money) are also present in many other blockchains, like Bitcoin, Litecoin, and so on.

After this necessary preliminary remark, we can now see how a social network $\mathcal{G}$, representing the Ethereum blockchain, can be built. Specifically:

$$\mathcal{G} = \langle N, A \rangle$$

Here, N is the set of nodes of $\mathcal{G}$. A node $n \in N$ corresponds to an Ethereum address that has made at least one transaction. Since there is a biunivocal correspondence between a node of $\mathcal{G}$ and an Ethereum address, in the following we will use these two terms interchangeably. Each node n has associated a label l_n, indicating the class which it belongs to (see below); l_n is set to null if no class has been assigned to n yet.

A represents the set of arcs of $\mathcal{G}$. There is an arc $a = (n_i, n_j, TrS_{ij}) \in A$ if there was at least one transaction from n_i to n_j. TrS_{ij} consists of a set of triplets $(tr_{ij_k}, \tau_{ij_k}, v_{ij_k})$, where tr_{ij_k} represents the k^{th} transaction from n_i to n_j, τ_{ij_k} indicates the corresponding timestamp and v_{ij_k} denotes the amount of Wei[1] transferred from n_i to n_j through tr_{ij_k}.

Modeling Ethereum as a social network allows us to use various Social Network Analysis measures to characterize each Ethereum address. In particular, we chose a set F of features that can support in distinguishing one class from another. They are:

- `In-degree`: it represents the number of arcs incoming to n_i and, therefore, the number of nodes of $\mathcal{G}$ pointing to n_i. It can be determined by computing the cardinality of the set:
$$IN_i = \{n_j | (n_j, n_i, TrS_{ji}) \in A\}$$
- `Out-degree`: it denotes the number of arcs outgoing from n_i and, therefore, the number of nodes of $\mathcal{G}$ which n_i points to. It can be determined by computing the cardinality of the set:
$$OUT_i = \{n_j | (n_i, n_j, TrS_{ij}) \in A\}$$
- `In-transaction`: it indicates the number of transactions towards n_i made by the nodes of $\mathcal{G}$. It can be computed as:
$$\sum_{n_j \in IN_i} |TrS_{ji}|$$
where $|TrS_{ji}|$ denotes the cardinality of the set TrS_{ji}.
- `Out-transaction`: it represents the number of transactions towards the nodes of $\mathcal{G}$ made by n_i. It can be computed as:
$$\sum_{n_j \in OUT_i} |TrS_{ij}|$$
- `In-value`: it denotes the total amount of Wei received by n_i. It can be computed as:

[1] Wei is the smallest fraction of Ether; it corresponds to 10^{-18} Ethers.

$$\sum_{n_j \in IN_i} \sum_{k=1..|TrS_{ji}|} v_{ji_k}$$

- `Out-value`: it indicates the total amount of Wei sent by n_i. It can be computed as:

$$\sum_{n_j \in OUT_i} \sum_{k=1..|TrS_{ij}|} v_{ij_k}$$

- `Clustering-coefficient`: it represents the clustering coefficient of n_i. Recall that, in Social Network Analysis, this parameter is an indicator of the tendency of n_i and its neighbors to form a cluster.
- `PageRank`: it denotes the PageRank of n_i. This parameter is an indicator of the number of links received by n_i, the centrality of the neighbors of n_i and their propensity to link to each other [430].

In our reference scenario, the time factor plays a key role. As a consequence, our model should take time into account. In fact, users continuously make transactions on Ethereum, which leads to continuous changes in the structure of the corresponding social network and the labels of its arcs.

In order to take time into consideration, given a time instant t, we denote with $\mathcal{G}(t)$ the social network associated with Ethereum that considers the transactions made on that blockchain from its appearance until t and, therefore, the transactions whose timestamp is less than or equal to t.

Similarly, given two time instants t_α and t_β, we can build a social network $\mathcal{G}(t_\alpha, t_\beta)$ representing Ethereum, and the transactions made on it, in the time interval $(t_\alpha, t_\beta]$. More formally, $\mathcal{G}(t_\alpha, t_\beta)$ considers only the transactions on Ethereum such that the corresponding timestamp is higher than t_α and less than or equal to t_β.

9.1.3 Defining the spectrum of a user or a class of users

We have introduced the eight features able to characterize an Ethereum address and we have presented the social network $\mathcal{G}(t_\alpha, t_\beta)$, modeling Ethereum in the time interval $(t_\alpha, t_\beta]$. We are now able to define the concept of spectrum of an Ethereum address in $(t_\alpha, t_\beta]$.

Let F be the set of features introduced in the previous section and let $T = (t_\alpha, t_\beta]$ be a time interval. We assume that T consists of a certain number of days. Let d_h be the h^{th} day of T. T can be represented as a succession $T = \{d_{\alpha+1} = d_1, d_2, \cdots, d_h, \cdots, d_q = d_\beta\}$ of q days. Let f_p be a parameter of F. It can have associated a time series $\Phi_p = \{\phi_{p_1}, \phi_{p_2}, \cdots, \phi_{p_h}, \cdots, \phi_{p_q}\}$, where ϕ_{p_h} is the value assumed by f_p at a constant and default time of d_h (for instance, at 12:00 am).

We define the spectrum $\mathcal{S}_i^T$ of a node n_i in the time interval T as the set $\mathcal{S}_i^T = \{\phi_{p_i} | f_p \in F$ and ϕ_{p_i} is the succession of the values assumed by f_p in n_i during $T\}$. In

other words, the spectrum of n_i in T is given by a set of successions, one for each feature of F. Each succession is made of the values assumed by the corresponding feature for the Ethereum address associated with n_i for the days belonging to T.

The spectrum $\mathcal{S}_i^T$ can be represented by a matrix that has q rows (one for each day of T) and nine columns. The first column is used to indicate the date, while the other eight ones correspond to the features of F. In particular, the semantics of the columns is as follows:

1. `Day`: its h^{th} element indicates the date corresponding to d_h.
2. `In-degree`: its h^{th} element denotes the number of addresses from which n_i received transactions during the time interval τ_h between 12:00 am of d_{h-1} and 12:00 am of d_h.
3. `Out-degree`: its h^{th} element indicates the number of addresses to which n_i has made transactions during τ_h.
4. `In-transaction`: its h^{th} element denotes the number of transactions received by n_i during τ_h.
5. `Out-transaction`: its h^{th} element indicates the number of transactions made by n_i during τ_h.
6. `In-value`: its h^{th} element denotes the amount of Wei received from n_i during τ_h.
7. `Out-value`: its h^{th} element indicates the amount of Wei sent by n_i during τ_h.
8. `Clustering-coefficient`: its h^{th} element denotes the clustering coefficient of n_i in the social network $\mathcal{G}(d_{h-1}, d_h)$.
9. `PageRank`: its h^{th} element indicates the PageRank of n_i in $\mathcal{G}(d_{h-1}, d_h)$.

9.1.4 Defining the new version of the Eros Distance

The algorithm for the Eros distance computation applies Principal Component Analysis [640] to two multivariate time series, each represented by means of a matrix. First it generates the principal components and their corresponding eigenvalues and eigenvectors. In our case, the eigenvectors are associated with the eight spectrum features. More specifically, each eigenvector corresponds to a feature and the associated eigenvalue represents the importance of that feature for the characterization of the address or the class which the spectrum refers to. Then, the algorithm uses principal components and their associated eigenvectors to compute the similarity of the two matrices associated with the multivariate time series under consideration. It is easy and fast to implement; at the same time, as stated in [653], the Eros distance outperforms other traditional similarity measures for multivariate time series, such as the Dynamic Time Warping [75], the Weighted Sum SVD [554], and so forth.

We selected the Eros distance as the reference metric for computing spectra similarities in our classification algorithm. In fact, this computes the distance between

a blockchain address to be classified and each possible class and assigns the address to the closest class. In this context, the Eros distance allows us to measure the similarity degree between two multivariate time series representing the spectrum of the address to classify and the one of a class.

The way our algorithm proceeds and the adoption of the Eros distance allow us to perform the address classification in a way that minimizes the distances between the spectra of the addresses of the same class and maximizes the distances between the spectra of the addresses of different classes.

The algorithm for the Eros distance computation uses some weights, one for each time series considered and, therefore, one for each feature. Each weight denotes the relative importance of the corresponding time series (and, therefore, of the corresponding feature) with respect to all the other ones.

The original version of the Eros distance described in [653] obtains these weights from the eigenvalues associated with the eigenvectors representing the time series being considered. Initially, we applied this version but, as we will see in Section 9.1.6, the results of the classification obtained in this way were not particularly satisfactory.

Nevertheless, we considered that the possibility, offered by the Eros distance, to associate a single value with the distance between two sets of multivariate time series was a key feature for our context. Therefore, we planned to define a new version of the Eros distance in which the weights are computed in a way tailored to our reference scenario. Regarding this, we recall that, in our case, whenever the Eros distance measures the similarity degree of two spectra, it has to consider two sets, each consisting of 8 time series. Each time series has associated a weight and the overall sum of the weights must be equal to 1. Therefore, in principle, we should consider 2 sets of 8 weights that can vary in any way between 0 and 1, with the only constraint that their overall sum must be equal to 1. It is reasonable to assume that the weights are decimal numbers with two digits after the decimal point. Even with this assumption, the problem is still NP-hard, because it would be necessary to exhaustively examine all the possible valid combinations of weights. As a consequence, despite the fact that, at the moment, the classes are only 4 and the features are only 8, we have judged opportune to preserve the scalability of our approach and to determine since now a heuristics to solve it. We have defined such a heuristics, which is reported in Algorithm 5.

Our heuristics receives in input:

- The set Cl of the classes of interest; in our case, this set consists of the classes "Token Contract", "Exchange", "Bancor" and "Uniswap".
- The set $\mathcal{S}_{Cl}$ of the spectra of the classes of Cl; as for our dataset, these are the spectra shown in Figures 9.4, 9.6, 9.8 and 9.10.

Input

- Cl: the set of the classes of interest
- $\mathcal{S}_{Cl}$: the set of the spectra of the classes of Cl
- $\mathcal{S}_{train}$: a set of sets of address spectra; $\mathcal{S}^i_{train}$ comprises all the addresses of the training set already assigned to the class Cl_i
- $step$: a decimal number in the interval $[0,1]$

Output

- $\mathcal{W}_{best}$: a set of weight sets such that $\mathcal{W}^i_{best}$ comprises all the weights computed for the class Cl_i

Require: min_d, max_d, d, min_q, max_q: a real; $Eros(S_x, S_y, w)$: a function computing the Eros distance between the spectra S_x and S_y using the set w of weights; w_t: a set of weights such that each weight is a two-digit decimal number in the interval $[0,1]$ and $\sum_{k=1}^{8} w_t^k = 1$; $\mathcal{W}_{temp}$: a set of weight sets

for $Cl_i \in Cl$ **do**
  $max_d = 0$
  $min_d = +\infty$
  initialize w_t as a random combination of two-digit decimal numbers such that $\sum_{k=1}^{8} w_t^k = 1$
  $\mathcal{W}_{temp} = \{w_t\}$
  Add to $\mathcal{W}_{temp}$ all the possible sets of weights obtained by increasing one component of w_t of one or more steps and decreasing another component of w_t of the same number of steps
  for $w_q \in \mathcal{W}_{temp}$ **do**
    $max_t = 0$
    $min_t = +\infty$
    for $S_j \in \mathcal{S}^i_{train}$ **do**
      $d = Eros(S_i, S_j, w_q)$
      if $d < min_q$ **then**
        $min_q = d$
      end if
    end for
    for $S_j \in \mathcal{S}^k_{train}, k \neq i$ **do**
      $d = Eros(S_i, S_j, w_q)$
      if $d > max_q$ **then**
        $max_q = d$
      end if
    end for
    if $(max_d < max_q)$ **and** $(min_d > min_q)$ **then**
      $\mathcal{W}^i_{best} = w_q$
      $max_d = max_q$
      $min_d = min_q$
    end if
  end for
end for
return $\mathcal{W}_{best}$

Algorithm 5: Heuristics for computing the best weight combination for each class

- The set $\mathcal{S}_{train}$ of the spectra of the training addresses; the element $\mathcal{S}^i_{train}$ represents the set of spectra of the training addresses assigned to the class Cl_i.
- The parameter $step$, which is a decimal number in the range $[0,1]$. As we will see below, it allows the management of a tradeoff between the accuracy and the computation time of our heuristics. In fact, the smaller the step, the more accurate the output of our heuristics, but the longer its computation time.

Our heuristics returns a set $\mathcal{W}_{best}$ of weights sets, one for each class. $\mathcal{W}_{best}$ is computed in such a way as to minimize the Eros distance between the spectra of the addresses of the same class and maximize the Eros distance between the spectra of the addresses of different classes. It also uses a function $Eros$ that receives two spectra S_x and S_y and a set w of weights and computes the Eros distance between S_x and S_y using the weights specified in w.

For each class Cl_i belonging to Cl, our heuristics builds the set w_t of weights as a random combination of two-digit decimal numbers such that $\sum_{k=1}^{8} w_t^k = 1$. This last condition is required by the Eros distance and must be verified by any admissible set of weights.

Starting with w_t as seed, our heuristics builds a set $\mathcal{W}_{temp}$ by increasing one of the weights of w_t of a value equal to $step$ and decreasing another one of the same value. It repeats this procedure for any pair of weights of w_t. In doing so, it may happen that some of the new combinations obtained are not admissible because one or both of the modified weights do not fall within the range $[0,1]$. These combinations are discarded.

Once the construction of this initial version of $\mathcal{W}_{temp}$ is finished, our heuristics proceeds with its enrichment. For this purpose, it repeats the same procedure by increasing a weight of w_t of a value equal to $2 \cdot step$ and decreasing another one of the same value. After this second iteration has been finished, it repeats the same procedure by increasing and decreasing the weights of w_t of a value equal to $3 \cdot step$, $4 \cdot step$, and so on. The enrichment of $\mathcal{W}_{temp}$ terminates when, during one iteration of this procedure, no new admissible pair is obtained.

From this description, we can see how $step$ acts as a regulator between accuracy and computation time. In fact, the lower its value, the higher the number of weight sets present in $\mathcal{W}_{temp}$ and, consequently, the higher the accuracy of our heuristics, but the longer its computation time. On the contrary, the higher the value of $step$, the lower the accuracy of our heuristics but the smaller its computation time.

At this point, $\mathcal{W}_{temp}$ has been completely constructed. Now, for each set $w_q \in \mathcal{W}_{temp}$, our heuristics applies the $Eros$ function, with the set w_q of weights, for computing the minimum distance min_q between the spectrum S_i of Cl_i and the spectrum S_j of any address assigned to Cl_i. Then, it applies $Eros$, with the same set of weights, for computing the maximum distance max_q between S_i and the spectrum S_j of any address assigned to a class different from Cl_i.

If the minimum current distance min_d concerning Cl_i is greater than min_q and the maximum current distance max_d concerning Cl_i is less than max_q, then max_d is set to max_q, min_d is set to min_q, w_q becomes the new best current set of weights for Cl_i and is assigned to $\mathcal{W}_{best}^i$.

After all the sets of weights of $\mathcal{W}_{temp}$ have been examined, the current value of $\mathcal{W}^i_{best}$ becomes final. At this point, a new class of Cl is selected and the whole procedure described above is repeated. After all the classes of Cl have been examined, our heuristics terminates and returns $\mathcal{W}_{best}$.

We end this description of the heuristics with some considerations regarding its accuracy and computation time. As mentioned above, our heuristics has one parameter, namely *step*, which acts as regulator. Its presence guarantees that our heuristics terminates (in fact, it would be enough to choose a high value of *step*). Clearly, this is not enough to say that our heuristics is adequate for the problem for which it was designed. In fact, it is necessary: *(i)* to show that the accuracy of results is acceptable; *(ii)* to verify that the computation time is acceptable and, in any case, much less than the time taken by an exhaustive approach for defining weights; *(iii)* if possible, to find a default value for *step* that can guarantee in most cases an excellent tradeoff between accuracy and computation time. We will devote Section 9.2 of the paper to address these issues. For now we anticipate that: *(i)* we found that setting *step* to 0.05 guarantees an excellent tradeoff between accuracy and computation time; *(ii)* the accuracy of the results obtained by our heuristics proved to be comparable with the one of the exhaustive approach; *(iii)* the computation time employed by our heuristics is much (in particular, several orders of magnitude) less than that of the exhaustive approach. In light of these results, we can say that our heuristics is adequate for the problem it aims to address.

9.1.5 Classifying users based on their spectra

In this section, we define a classification algorithm that, given a time interval T and an address a_j whose spectrum in T is known, assuming that the spectra of the four classes of interest in T are known, is able to classify a_j. In particular, the algorithm may assign a_j to one of the four classes or may conclude that a_j does not belong to any of them.

We observe that the classification problem we are considering is complex because it involves comparing spectra and calculating a similarity degree between them. In particular, each spectrum consists of a set of time series. As we saw in Section 9.1.10, these are not independent of each other but are correlated. Even if, given two features with a correlation degree equal to 1, we remove one of them and keep the other, we would not have solved the problem because the remaining features would still be partially correlated to each other. As a consequence, we must handle multivariate time series.

Recall that, as stated in the Introduction, the past literature provides some approaches to classify multivariate time series [336, 66, 547]. We have also specified

that, to the best of our knowledge, there is no out-of-the-box classification approach that can be easily implemented in our case. Therefore, we preferred to define a new technique tailored to the characteristics of the problem we want to face. This technique involves the modeling of the blockchain as a social network and the next derivation of the appropriate features from it.

The core of such an algorithm consists of a metric able to compute a similarity degree between multivariate time series. In order to perform this task, we rely on the Eros distance, also known as Extended Frobenius Norm [653].

Once the weights of $\mathcal{W}_{temp}$ have been computed, the definition of the classification algorithm is straightforward. In fact, given an address a_j to be classified, it is sufficient to compute the Eros distance between the spectrum S_j of a_j and the spectrum of each available class. a_j will be assigned to the class with the minimum distance. We report the corresponding pseudo-code in Algorithm 6.

Input

- a_j: the Ethereum address to be classified
- $\mathcal{S}_j$: the spectrum of a_j
- Cl: the set of the classes of interest
- $\mathcal{S}_{Cl}$: the set of the spectra of the classes of Cl
- $\mathcal{W}_{best}$: the set of the best weights identified by our heuristics

Output

- the class $\overline{Cl}$ which a_j is assigned to

Require: min_d, th_{dmax}, d: a real; $Eros(S_x, S_y, w)$: a function computing the Eros distance between the spectra S_x and S_y using the set w of weights;

$\overline{Cl} = null$;
$min_d = +\infty$
for $Cl_i \in Cl$ **do**
  $d = Eros(S_i, S_j, \mathcal{W}^i_{best})$
  if $d < min_d$ **then**
    $min_d = d$
    $\overline{Cl} = Cl_i$
  end if
end for
return $\overline{Cl}$

Algorithm 6: Algorithm for classifying a new address

9.1.6 Experiments

In this section, we present several experiments that helped us to define the details of our approach. In particular, in Subsection 9.1.7, we present the dataset we used for training and testing it. In Subsection 9.1.8, we describe an example of user spectrum. In Subsection 9.1.9, we present the process that led us to define the classes of interest. In Subsection 9.1.10, we illustrate the spectra of the selected classes. Finally, in Subsection 9.1.11, we present the application, to the dataset of interest, of the method for computing the weights of the Eros distance.

In order to carry out our experiments, we used a server equipped with 16 Intel Xeon E5520 CPUs and 96 GB RAM with the Ubuntu 18.04.3 operating system. We adopted Python 3.6 as programming language, its library Pandas to perform ETL operations on data, and its library NetworkX to carry out operations on networks.

9.1.7 Dataset

In order to carry out our analyses, we derived a dataset from Ethereum. In particular, we downloaded the corresponding data from Google BigQuery[2]. The data we selected covers a period from September 1^{st}, 2019 to October 31^{st}, 2019. We chose it because we wanted to test our approach in a "normal" period for Ethereum, i.e., a period when there were no particular speculative bubbles. In fact the latter can heavily modify user behaviors and deserve a separate study [88]. We selected all the transactions made on Ethereum in that period. The total number of transactions considered in the dataset is 41,420,435, whereas the total number of addresses is 5,553,645. We computed some statistics on the dataset; they are reported in Table 9.1.

Table 9.1: Some preliminary statistics performed on our dataset

Parameter	*Value*
Number of transactions	41,420,435
Total number of addresses	5,553,645
Total number of `from_address`	4,980,691
Total number of `to_addresses`	4,471,985
Cardinality of the intersection between `from_address` and `to_address`	3,899,031
Number of null `from_address`	1
Number of null `to_address`	2

The distribution of transactions over time is reported in Figure 9.1. From the analysis of this figure we can see that the number of transactions is always in a range

[2] `https://www.kaggle.com/bigquery/ethereum-blockchain`

between 600,000 and 800,000. This trend is substantially constant with a slight decrease observed in the second half of September balanced by an increase in the first half of October. In any case, in the time interval of our dataset, we do not observe significant peaks that could suggest the presence of a speculative bubble.

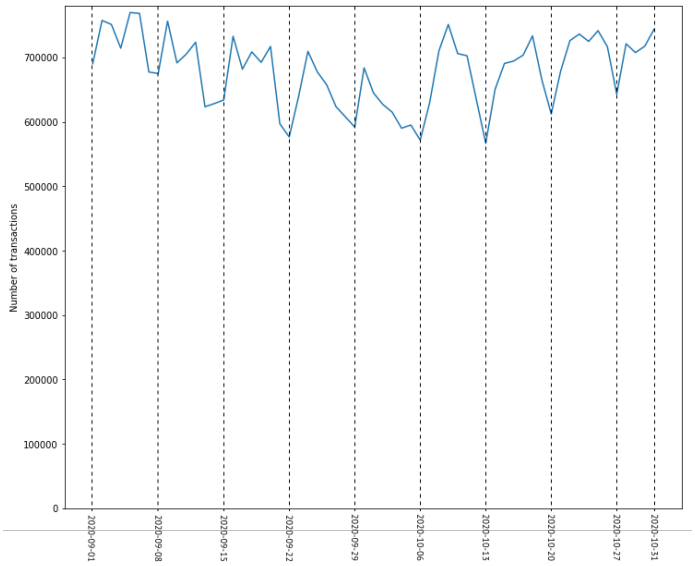


Fig. 9.1: Number of transactions over time

During the dataset construction we had to perform some ETL (Extraction, Transformation and Loading) operations. In particular, first we removed some duplicate transactions that were present in the dataset since they cannot exist in a blockchain. Their presence was likely due to a download error. In addition, we removed all transactions in which at least one field had a null value. In fact, this type of transactions could not be used for our tests. After these basic tasks, we performed some additional, more specific, ones. In particular, we removed transactions in which at least one of the addresses involved had a wrong hexadecimal value, different from the standard expected by Ethereum. We also removed transactions in which a "dead address" was present, i.e., those transactions in which tokens are sent to be burned. Last but not least, we unified all amounts of money exchanged by representing them with a single currency, i.e., Wei.

After them, we were able to associate a dataset row with each transaction. Each row consists of four columns, namely: *(i)* `from_address`, representing the blockchain address starting the transaction; *(ii)* `to_address`, denoting the blockchain address receiving the transaction; *(iii)* `timestamp`, indicating the transaction timestamp; *(iv)* `value`, representing the amount of Wei transferred during the transaction.

We split our dataset into two parts. The former contains all the transactions made in September 2019; it consists of 20,465,806 transactions and was used for training. The latter comprises all the transactions made in October 2019; it consists of 20,954,629 transactions and was employed for testing.

Everything we describe in this section refers to an Exploratory Data Analysis on the dataset, as well as on training activities. Instead, we will describe the testing activities in Section 9.2.

9.1.8 An example of user spectrum

An example of user spectrum is shown in Table 9.2. It refers to the Ethereum address encoded as `0xf0ee6b27b759c9893ce4f094b49ad28fd15a23e4` and to the time interval T ranging from September 1^{st}, 2019 to September 30^{th}, 2019.

Table 9.2: An example of a user spectrum

day	*In-degree*	*Out-degree*	*In-transactions*	*Out-transactions*	*In-value*	*Out-value*	*Clustering-coefficient*	*PageRank*
2019-09-01	14	0	36	0	36	0	0.000020	0.021978
2019-09-02	11	0	24	0	24	0	0.000014	0.010526
2019-09-03	30	0	45	0	45	0	0.000019	0.003171
2019-09-04	21	0	36	0	36	0	0.000015	0.003025
2019-09-05	16	0	28	0	28	0	0.000013	0.002261
2019-09-06	22	0	46	0	46	0	0.000013	0.002272
2019-09-07	25	0	54	0	54	0	0.000014	0.002922
2019-09-08	18	0	46	0	46	0	0.000026	0.002871
2019-09-09	15	0	45	0	45	0	0.000026	0.002669
2019-09-10	22	0	63	0	63	0	0.000028	0.002312
2019-09-11	24	0	78	0	78	0	0.000031	0.002150
2019-09-12	25	0	85	0	85	0	0.000031	0.002070
2019-09-13	18	0	49	0	49	0	0.000031	0.002020
2019-09-14	8	0	22	0	22	0	0.000030	0.001925
2019-09-15	10	0	12	0	12	0	0.000029	0.001733
2019-09-16	24	0	34	0	34	0	0.000031	0.001689
2019-09-17	12	0	18	0	18	0	0.000030	0.001578
2019-09-18	24	0	34	0	34	0	0.000031	0.001543
2019-09-19	13	0	16	0	16	0	0.000031	0.001587
2019-09-20	24	0	35	0	35	0	0.000031	0.001542
2019-09-21	23	0	29	0	29	0	0.000031	0.001501
2019-09-22	12	0	20	0	20	0	0.000032	0.001494
2019-09-23	15	0	29	0	29	0	0.000032	0.001462
2019-09-24	19	0	43	0	43	0	0.000031	0.001436
2019-09-25	28	0	55	0	55	0	0.000032	0.001481
2019-09-26	20	0	31	0	31	0	0.000031	0.001436
2019-09-27	15	0	33	0	33	0	0.000031	0.001440
2019-09-28	17	0	29	0	29	0	0.000032	0.001339
2019-09-29	27	0	57	0	57	0	0.000033	0.001308
2019-09-30	19	0	27	0	27	0	0.000033	0.001308

9.1.9 Defining the classes of interest

In order to define our classification approach, it was necessary to identify the classes of interest. For this purpose, we exploited information provided by Etherscan. At the time of writing, this service provider has defined 426 possible classes. Clearly, it is impractical to think of building a classification approach with such a large number of classes. Therefore, it seemed appropriate to detect the most common ones by checking the distribution of the current addresses against the classes provided by Etherscan. To this end, we selected uniformly at random a set of 2,010,729 Ethereum addresses from the training data of our dataset and verified their classes (if any) on Etherscan. This check returned a class for 4,443 of them. Figure 9.2 shows the distribution of these addresses against the main classes handled by Etherscan.

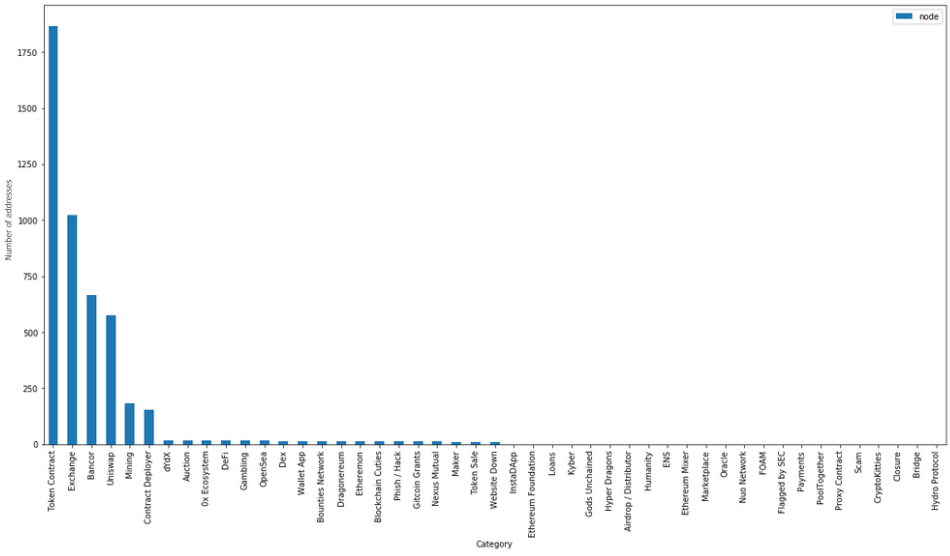


Fig. 9.2: Distribution of Ethereum training addresses against the main Etherscan classes

From the analysis of this figure, it is clear that the distribution follows a power law. The majority of the addresses (41.99%) belongs to the class “Token Contracts”. Immediately after, there are the classes “Exchange” (22.97%), “Bancor” (14.98%) and “Uniswap” (12.98%). Overall, these four classes cover 92.92% of Ethereum addresses labeled by Etherscan. For this reason, we decided to focus our classification approach on them in order to reconstruct, for each class, a very precise profile, clearly distinguishing it from the others. The addition of more classes would have risked creating partially overlapping class profiles with a negligible increase in the number of addresses that could be classified. The semantics of the four classes we chose is as follows:

- The *"Token Contract"* class includes addresses using tokens instead of Ether. Tokens are an alternative currency to Ether, used to fasten up and simplify processes.
- The *"Exchange"* class includes addresses acting as money changers; these allow clients to buy and sell cryptocurrencies.
- The *"Bancor"* class includes addresses acting as banks. A bancor allows clients to deposit and convert each available token in the network, without counterparts, automatically at a given price, using a simple web wallet.
- The *"Uniswap"* class includes addresses using the "Uniswap"[3] protocol for the automatic exchange of tokens in Ethereum.

In Table 9.3, we report the number of addresses for each of these classes.

Table 9.3: Number of addresses belonging to each class of interest for our investigation

Class	*Number of addresses*
Token Contract	1,866
Exchange	1,021
Bancor	666
Uniswap	577

9.1.10 Defining class spectra

After determining the classes of interest, in this section we want to define the spectrum of each class. As a first step, we need to check if all the features identified in Section 9.1.3 are independent from each other or if there are correlations between them.

To answer this question, for all the addresses of our training set, we computed the spectrum with reference to the corresponding time interval, i.e., from September 1^{st}, 2019 to September 30^{th}, 2019. Then, we computed the overall correlation matrix associated with all the addresses of our training set. For this purpose, we set the value of each element of the matrix equal to the average of the values of the corresponding elements for all addresses. The matrix thus obtained is shown in Figure 9.3.

From the analysis of this figure we can see that there are totally correlated features. In fact, `In-transaction` is totally correlated with `In-value`, while `Out-transaction` is totally correlated with `Out-value`. Furthermore, there are other strong correlations. For instance, `In-degree` is strongly correlated with `In-transaction`

[3] `https://uniswap.org`

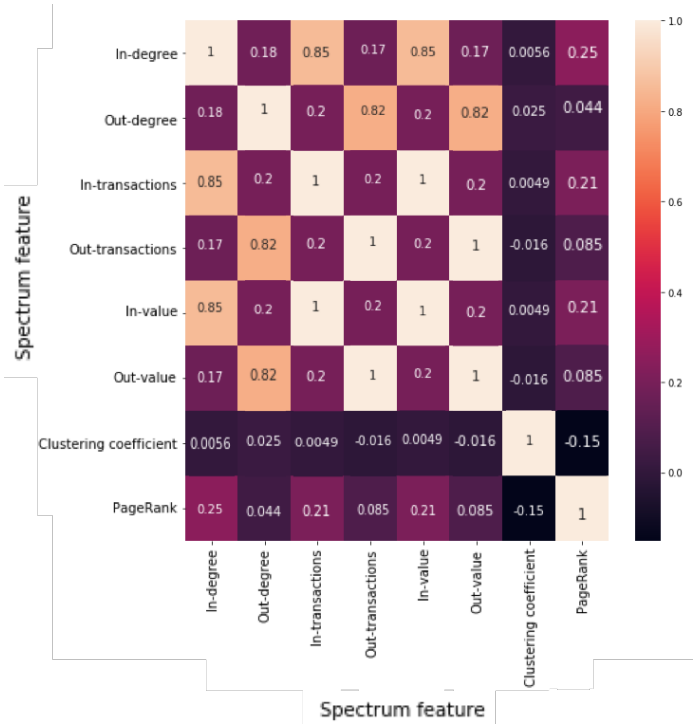


Fig. 9.3: Correlation matrix for the spectrum features of all the addresses in the training data set

and `In-value`, while `Out-degree` is strongly correlated with `Out-transaction` and `Out-value`.

This result is extremely important because it allows us to draw the following two relevant conclusions:

- In principle, we could remove one feature between `In-transaction` and `In-value` and one feature between `Out-transaction` and `Out-value` from the spectrum. We decided not to do so because the result refers to a specific time interval. We believe it is plausible that it applies to the other time intervals as well. However, since a formal proof of this is not possible, we felt it appropriate to preserve all features. As a consequence of this decision, it is to be expected that some spectrum features will have perfectly coincident trends in the following.
- There are strong correlations between several spectrum features. Consequently, they cannot be considered independent of each other and the spectrum of an address in a time interval must be analyzed as a multivariate time series.

After considering the overall spectrum representing all users in the dataset, in the next subsections we examine the spectrum of the four classes of interest determined above.

Spectrum of the class "Token Contract". Given all the nodes of the class "Token contract" in the training period, we computed the minimum, maximum, mean and standard deviation of the values of the spectrum features. They are shown in Table 9.4.

Table 9.4: Minimum, maximum, mean and standard deviation of the values of the spectrum features for the class "Token Contract"

Feature	*Minimum Value*	*Maximum Value*	*Mean Value*	*Standard Deviation*
In-degree	4.65	91.40	20.52	18.60
Out-degree	0	0	0	0
In-transaction	10.80	354.44	59.24	70.76
Out-transaction	0	0	0	0
In-value	10.81	314.44	59.24	70.76
Out-value	0	0	0	0
Clustering-coefficient	$5.80 \cdot 10^{-4}$	$2.90 \cdot 10^{-2}$	$8.40 \cdot 10^{-3}$	$7.30 \cdot 10^{-3}$
PageRank	$1.61 \cdot 10^{-5}$	$9.41 \cdot 10^{-5}$	$5.97 \cdot 10^{-5}$	$2.24 \cdot 10^{-5}$

Then, in order to generate the spectrum of this class, we considered, for each feature and for each day of the training period, the average of the corresponding values for all the nodes of that class. The corresponding result is shown in Figure 9.4.

As can be seen in this figure, there are spectrum features having an identical trend, as we expected based on what we said in Section 9.1.10. These are In-transaction and In-Degree, on the one hand, and Out-transaction, Out-degree and Out-value, on the other hand. In addition, there are strong similarities between the trends of In-degree on the one hand, and In-transaction and In-value on the other hand. To quantify this fact, we computed the correlation matrix for the spectrum features of this class. It is shown in Figure 9.5. This figure also reveals another interesting correlation, i.e., a strong inverse correlation between Clustering-coefficient and PageRank.

Spectrum of the class "Exchange". The minimum, maximum, mean and standard deviation of the values of the spectrum features for the class "Exchange" are reported in Table 9.5. Figure 9.6 shows the spectrum of this class.

One interesting characteristic that can be observed in this spectrum is the absence of features with constant null value. As we will see in the next subsections, when we will examine the spectrum of the other classes, this characteristic is specific of the class "Exchange" and cannot be found in any other classes. Already from a visual analysis of this spectrum, we can observe that the trends of In-transaction, In-degree and In-value are identical. Similarly, the trends of Out-transaction

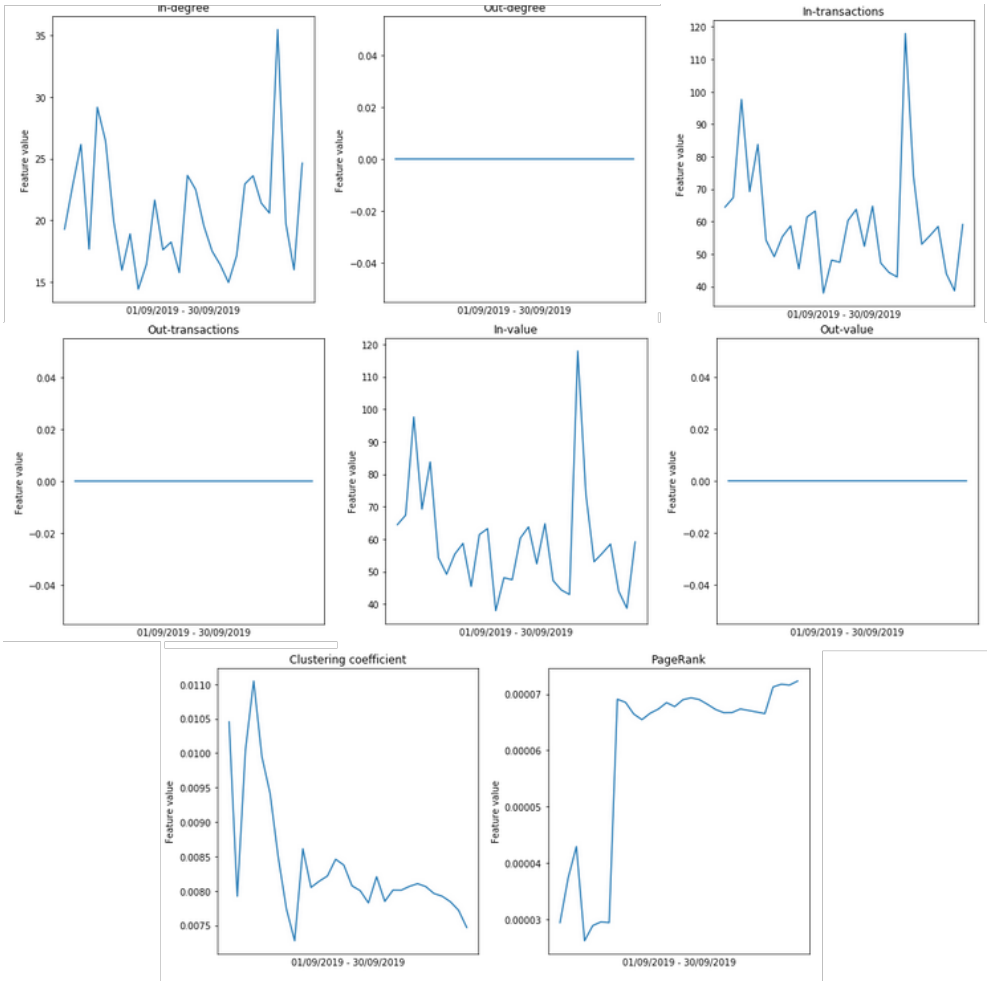


Fig. 9.4: Spectrum of the class "Token Contract"

Table 9.5: Minimum, maximum, mean and standard deviation of the values of the spectrum features for the class "Exchange"

Feature	*Minimum Value*	*Maximum Value*	*Mean Value*	*Standard Deviation*
`In-degree`	73.00	322.05	145.22	96.60
`Out-degree`	21.40	190.13	83.78	55.43
`In-transaction`	84.56	387.67	173.85	81.61
`Out-transaction`	76.37	417.83	185.35	93.10
`In-value`	84.56	387.67	173.85	81.61
`Out-value`	76.37	417.83	185.33	93.10
`Clustering-coefficient`	$5.26 \cdot 10^{-4}$	$1.99 \cdot 10^{-2}$	$4.99 \cdot 10^{-3}$	$5.02 \cdot 10^{-3}$
PageRank	$2.76 \cdot 10^{-4}$	$5.68 \cdot 10^{-4}$	$4.43 \cdot 10^{-4}$	$8.00 \cdot 10^{-5}$

and `Out-value` are identical. There is also a strong correlation between these last trends and the one of `Out-degree`.

Again, we computed the correlation matrix for the features of this class. It is reported in Figure 9.7. It shows a correlation value equal to 1 between `In-degree`, `In-transactions` and `In-value`, as well as between `Out-transactions` and `Out-`

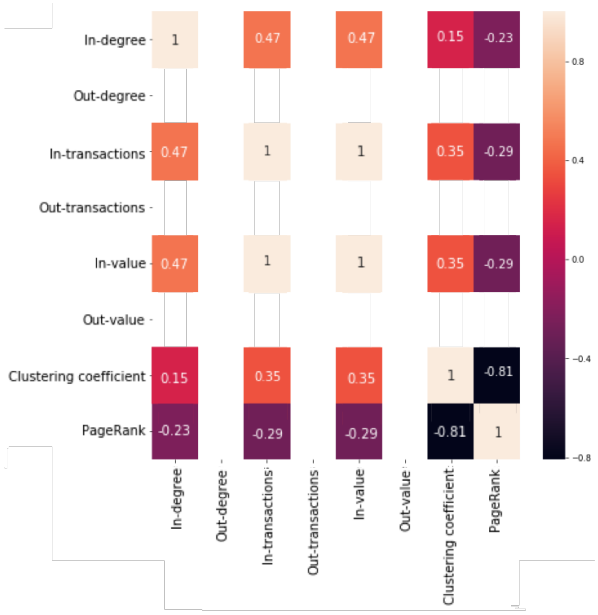


Fig. 9.5: Correlation matrix for the spectrum features of the class "Token Contract"

value. There is also a very high correlation, equal to 0.92, between Out-degree and Out-transactions and between Out-degree and Out-value. All these values fully confirm what we have deduced above from the direct observations of the trends in Figure 9.6.

Spectrum of the class "Bancor". In Table 9.6, we report the minimum, maximum, mean and standard deviation of the values of the spectrum features for the class "Bancor". In Figure 9.8, we show the spectrum of this class.

Table 9.6: Minimum, maximum, mean and standard deviation of the values of the spectrum features for the class "Bancor"

Feature	*Minimum Value*	*Maximum Value*	*Mean Value*	*Standard Deviation*
In-degree	0.42	9.63	3.10	2.23
Out-degree	0	0	0	0
In-transaction	1.57	37.40	9.47	8.04
Out-transaction	0	0	0	0
In-value	1.57	37.47	9.47	8.04
Out-value	0	0	0	0
Clustering-coefficient	$1.87 \cdot 10^{-4}$	$4.27 \cdot 10^{-3}$	$1.32 \cdot 10^{-3}$	$1.01 \cdot 10^{-3}$
PageRank	$8.99 \cdot 10^{-7}$	$3.57 \cdot 10^{-6}$	$1.49 \cdot 10^{-6}$	$6.21 \cdot 10^{-7}$

From the analysis of this spectrum we can see that the trends of Out-transaction, Out-degree and Out-value are identical. An analogous discourse is valid for the

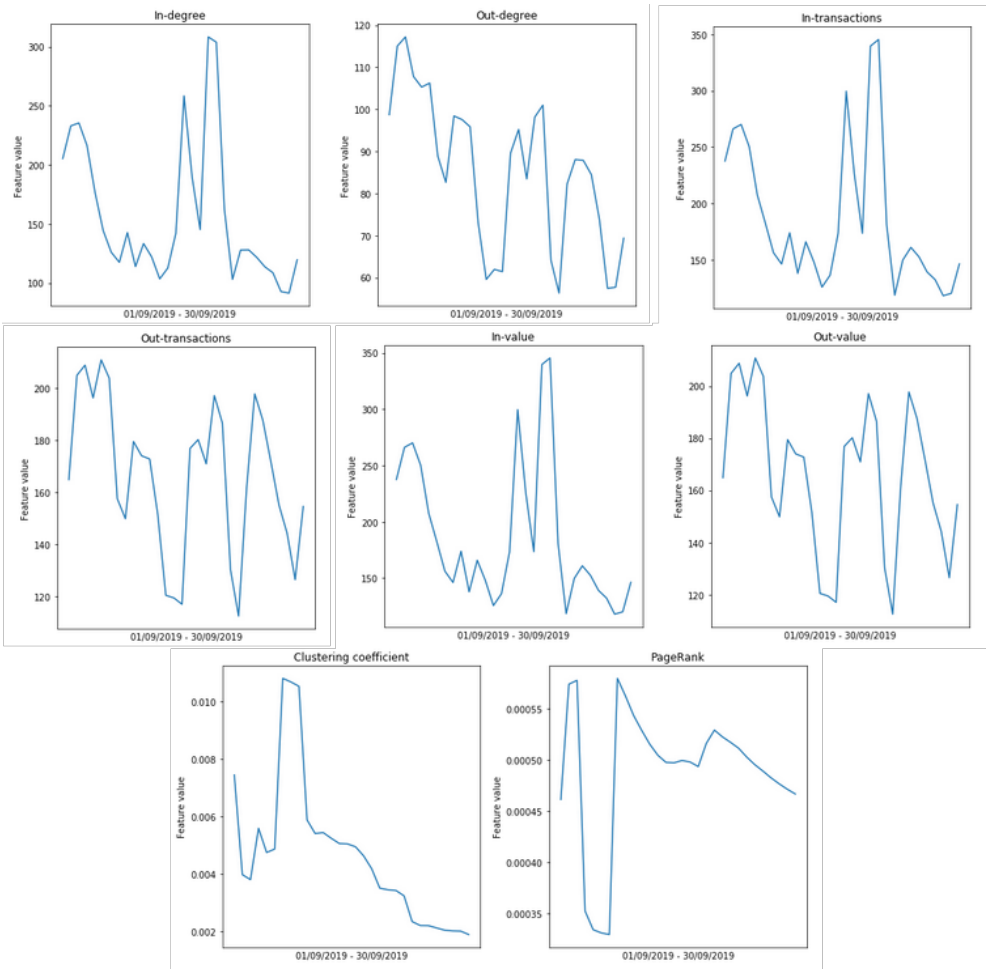


Fig. 9.6: Spectrum of the class “Exchange”

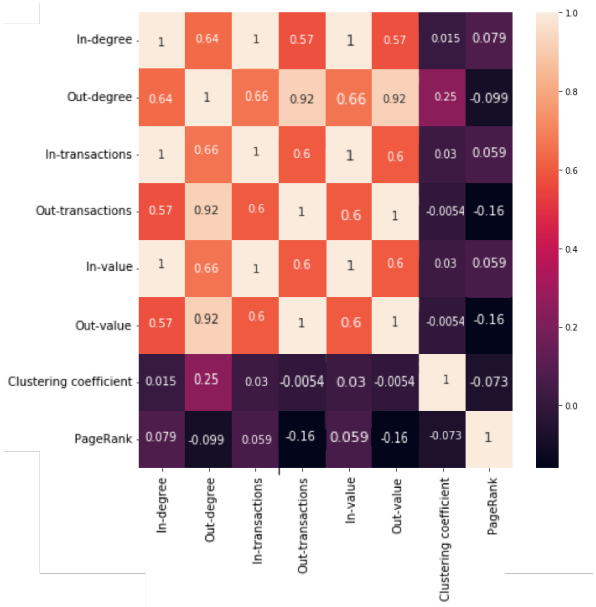


Fig. 9.7: Correlation matrix for the spectrum features of the class “Exchange”

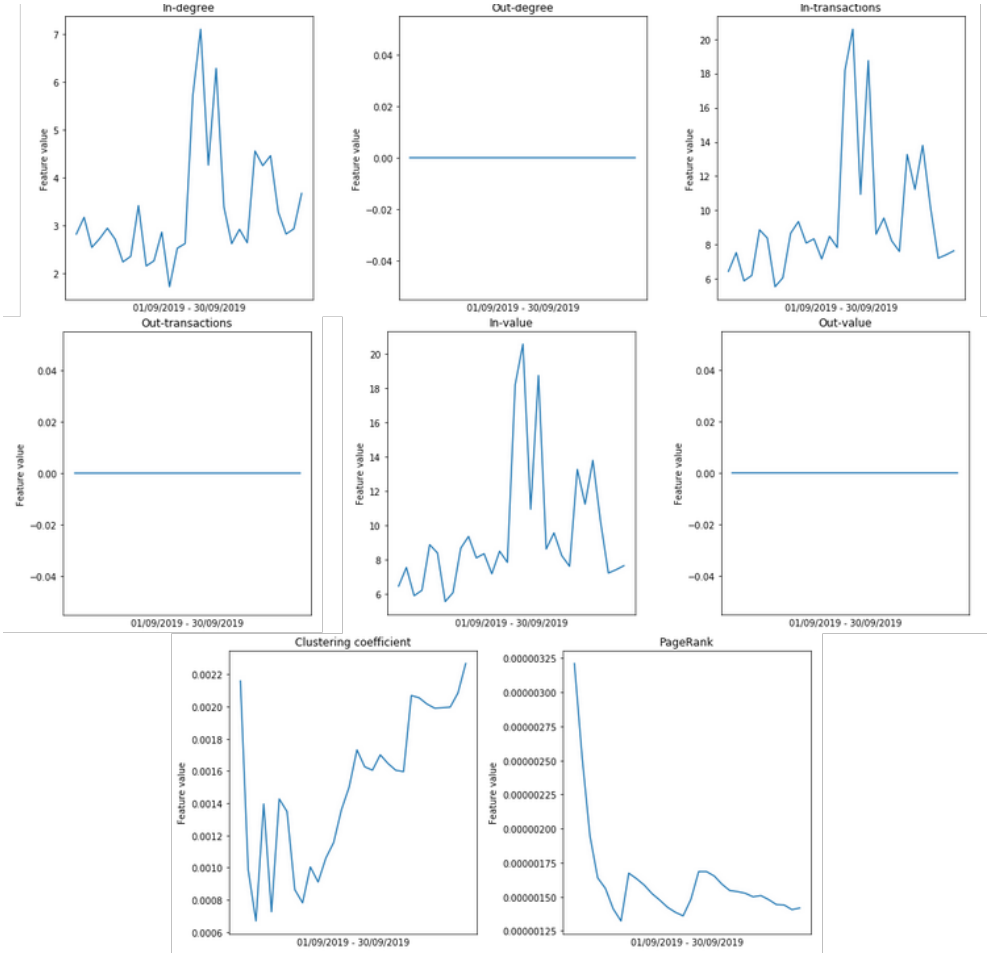


Fig. 9.8: Spectrum of the class "Bancor"

trends of `In-transaction` and `In-value`, which, in turn, show a strong correlation with the trend of `In-degree`.

Also for this class we quantified these correlations by computing the correlation matrix for the features of its spectrum. In Figure 9.9, we report such a matrix. Its analysis confirms all the previous observations and also highlights a good correlation between `Clustering-coefficient` and `In-degree`. It also reveals a strong correlation between `In-transaction`, `In-value` and `In-degree`, on one hand, and `Out-transaction`, `Out-value` and `Out-degree`, on the other hand. This is typical of this class of addresses that represents bankers.

Spectrum of the class "Uniswap". The minimum, maximum, mean and standard deviation of the values of the spectrum features for the class "Uniswap" are reported in Table 9.7. The spectrum of this class is shown in Figure 9.10.

From the analysis of this spectrum, we can see that the trends of `Out-transaction`, `Out-degree` and `Out-value` are identical. The same conclusion applies to the trends

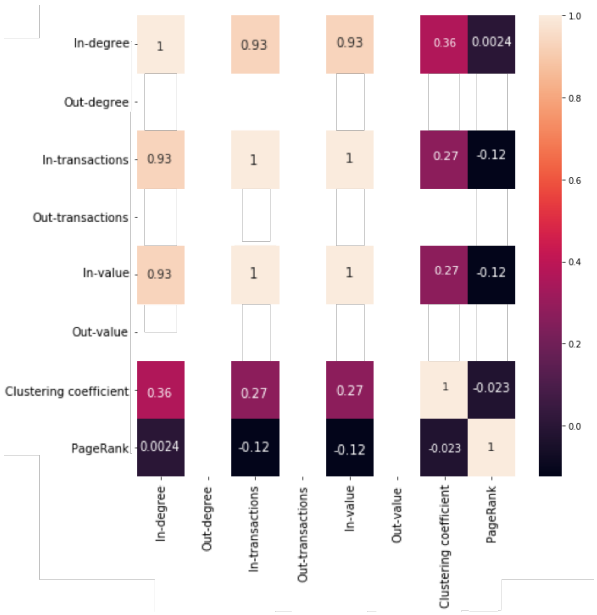


Fig. 9.9: Correlation matrix for the spectrum features of the class “Bancor”

Table 9.7: Minimum, maximum, mean and standard deviation of the values of the spectrum features for the class “Uniswap”

Feature	*Minimum Value*	*Maximum Value*	*Mean Value*	*Standard Deviation*
`In-degree`	0.42	9.63	3.10	2.23
`Out-degree`	0	0	0	0
`In-transaction`	1.57	37.40	9.47	8.04
`Out-transaction`	0	0	0	0
`In-value`	1.57	37.47	9.47	8.04
`Out-value`	0	0	0	0
`Clustering-coefficient`	$1.87 \cdot 10^{-4}$	$4.27 \cdot 10^{-3}$	$1.32 \cdot 10^{-3}$	$1.01 \cdot 10^{-3}$
`PageRank`	$8.99 \cdot 10^{-7}$	$3.57 \cdot 10^{-6}$	$1.49 \cdot 10^{-6}$	$6.21 \cdot 10^{-7}$

of `In-transaction` and `In-value`. In addition, we can observe a strong correlation between the trend of `In-degree` and the ones of `In-value` and `In-transaction`.

In Figure 9.11, we report the correlation matrix for the features of this spectrum. This figure confirms all the previous observations. As for this class, it also shows a strong correlation between `Clustering-coefficient` and `PageRank` and a good correlation between `PageRank` and `In-Degree`.

9.1.11 Weights of the Eros distance

In order to give an idea of the behavior of our heuristics for determining the weights of the Eros distance, in Table 9.8 we report the set of the weights of $\mathcal{W}_{temp}$ for the training data of our dataset. The examination of this table provides us with the following information:

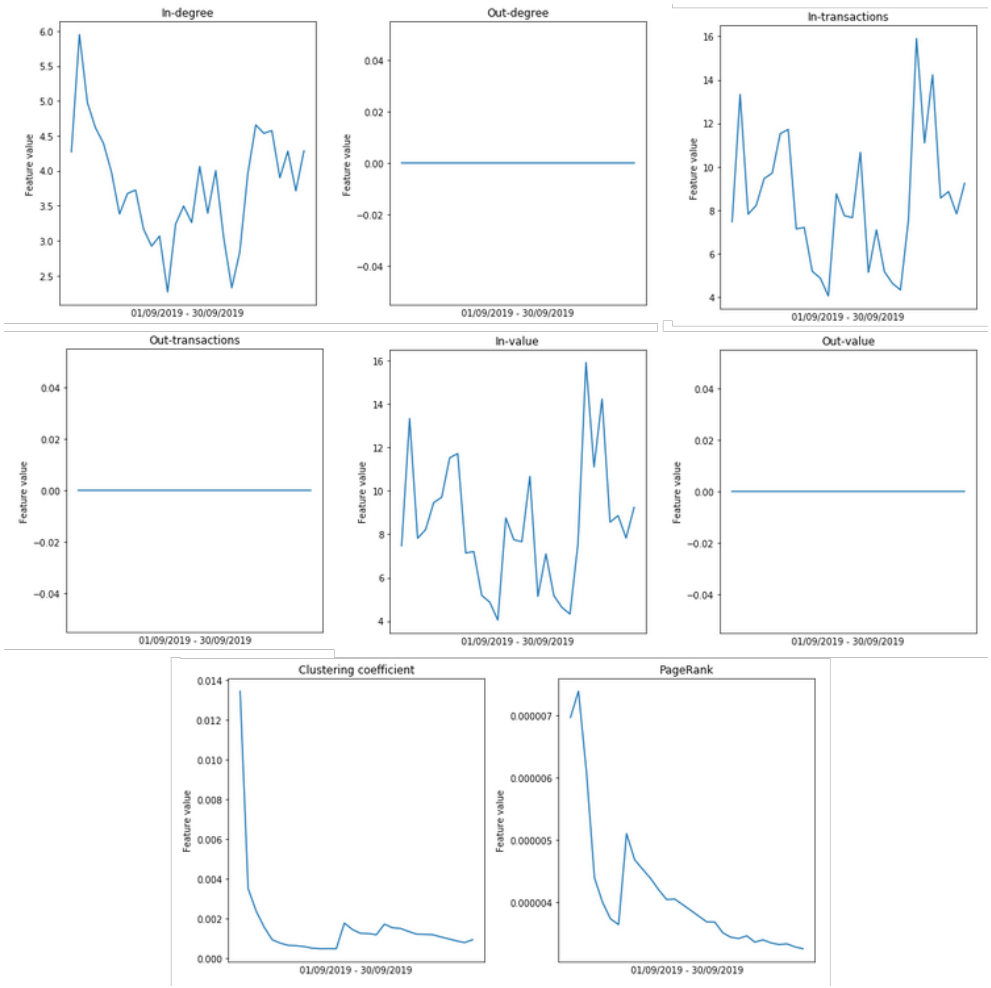


Fig. 9.10: Spectrum of the class “Uniswap”

- As for the class “Token Contract” the most important features are `In-transactions` and `In-value`. An intermediate weight is assigned to `In-degree`, `Clustering-coefficient` and PageRank. Finally, `Out-degree`, `Out-transactions` and `Out-value` have no weight.
- As far as the class “Exchange” is concerned, all features have roughly similar weights.
- Regarding the class “Bancor”, the most important features are `In-transactions`, `In-value` e `In-degree`. A fairly small weight is assigned to PageRank and `Clustering-coefficient`. Finally, the other ones have no weight.
- As far as the class “Uniswap” is concerned, the most important features are PageRank and `Clustering-coefficient`. A small to medium weight is assigned to the features `In-degree`, `In-transactions` and `In-value`. The other ones have no weight.

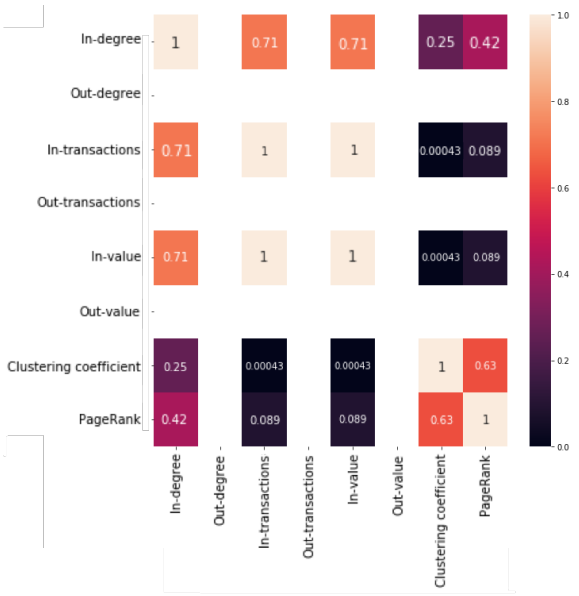


Fig. 9.11: Correlation matrix for the spectrum features of the class "Uniswap"

Comparing the weights shown in Table 9.8 with the spectra shown in Figures 9.4, 9.6, 9.8 and 9.10 and with the correlation matrices reported in Figures 9.5, 9.7, 9.9 and 9.11, the results obtained by our heuristics appear compatible with the knowledge that a human expert could derive from those figures. Clearly their actual validity must be confirmed by experiments; these will be illustrated in the next section.

9.2 Results

In this section, we present the tests we carried out to evaluate the performance of our classification approach. Specifically, in Subsection 9.2.1, we analyze our classification approach with the original Eros distance. In Subsection 9.2.2, we consider our classification approach with an exhaustive examination of all the combinations of the weights of the Eros distance. In Subsection 9.2.3, we analyze our classification approach supported with the new Eros distance with *step* set to 0.05, which proved able to guarantee an excellent tradeoff between accuracy and computation time. Finally, in Subsection 9.2.4, we give an idea of the computation times associated with the various steps of our approach.

As mentioned in Section 9.1.7, testing data in our dataset includes 20,954,629 transactions (i.e., all the transactions carried out on Ethereum from October 1^{st}, 2019 to October 31^{st}, 2019). Similarly to what we did for training data (see Section 9.1.10), we selected 2,120,834 Ethereum addresses uniformly at random from testing data

Table 9.8: Weights combination for the Eros distance relative to each class of interest

Class	*Weights*
Token Contract	In-degree: 0.18 Out-degree: 0 In-transactions: 0.26 Out-transactions: 0 In-value: 0.26 Out-value: 0 PageRank: 0.14 Clustering-coefficient: 0.16
Exchange	In-degree: 0.13 Out-degree: 0.15 In-transactions: 0.13 Out-transactions: 0.15 In-value: 0.13 Out-value: 0.15 PageRank: 0.10 Clustering-coefficient: 0.06
Bancor	In-degree: 0.27 Out-degree: 0 In-transactions: 0.27 Out-transactions: 0 In-value: 0.27 Out-value: 0 PageRank: 0.10 Clustering-coefficient: 0.09
Uniswap	In-degree: 0.12 Out-degree: 0 In-transactions: 0.12 Out-transactions: 0 In-value: 0.12 Out-value: 0 PageRank: 0.31 Clustering-coefficient: 0.33

and derived the corresponding classes from Etherscan. It was able to label 4,568 addresses whose distribution is shown in Figure 9.12.

As reported in this figure, the first four classes were “Token Contract”, “Exchange”, “Bancor” and “Uniswap”. They covered 93.73% of the Ethereum addresses labeled by Etherscan. Table 9.9 reports the number of addresses assigned by Etherscan to these classes. These assignments represent the ground truth for the experiments described in the next subsections.

9.2.1 Evaluating our approach with the original Eros distance

In this section, we evaluate our classification approach with the original version of the Eros distance for computing the similarity degree of two spectra. Recall that, in this version, the weights are obtained from the eigenvalues associated with the eigen-

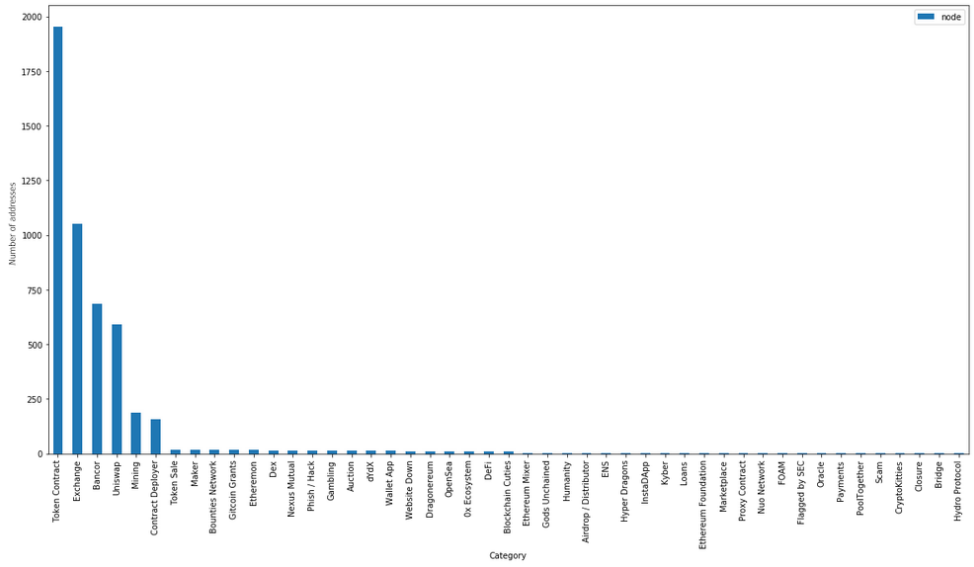


Fig. 9.12: Distribution of Ethereum testing addresses against the main categories of Etherscan

Table 9.9: Number of addresses belonging to each class of interest

Class	*Number of addresses*
Token Contract	1,954
Exchange	1,052
Bancor	684
Uniswap	592

vectors representing the time series under consideration. To perform our evaluation, we applied our classification algorithm with the original Eros distance providing as input to it the 4,568 testing addresses already labeled by Etherscan.

The computation time of this algorithm, when adopting the hardware framework described in Section 9.1.7, is equal to 21 seconds. It is acceptable if we consider that we are managing multivariate time series. However, it is still high compared to a classic classification algorithm, in which each class is represented by the value of a single parameter.

The confusion matrix we obtained is shown in Table 9.10. From the analysis of this matrix we can see that the results, albeit acceptable, are not particularly satisfactory. In order to have numerical indicators capable of quantifying the goodness of the results obtained, we computed the Micro- and Macro- Average Precision, Average Recall and Average F1-Score, as well as the overall Accuracy.

Recall that, in a multi-class classification, Micro-Average means computing Precision, Recall and F1-Score considering true positives, true negatives, false positives and false negatives together, without distinguishing between classes. On the con-

Table 9.10: Confusion matrix of our classification algorithm with the classical version of the Eros distance

	Token Contract	Exchange	Bancor	Uniswap
Token Contract	1,632	88	224	10
Exchange	62	964	54	72
Bancor	124	20	523	17
Uniswap	18	70	20	484

trary, Macro-Average means computing the metrics independently for each class and, then, computing the average of the values thus obtained. Instead, the overall Accuracy is simply defined as the ratio of the number of correctly classified instances to the total number of instances. All the seven parameters of our interest have a value ranging in the real interval $[0, 1]$; the higher the value, the higher the goodness of the approach being evaluated [310].

As for our experiment, the values of Micro- and Macro- Average parameters and the one of Accuracy are reported in Table 9.11.

Table 9.11: Values of some quality metrics obtained by applying our classification algorithm with the original Eros distance on testing data

Metric	*Value*
Accuracy	0.75
Micro Average Precision	0.74
Macro Average Precision	0.62
Micro Average Recall	0.74
Macro Average Recall	0.63
Micro Average F1-Score	0.76
Macro Average F1-Score	0.76

This table confirms, from a quantitative viewpoint, what we have qualitatively observed above, namely that the original eigenvalues-based method for computing the Eros distance is not suitable for our context.

9.2.2 Evaluating our approach with an exhaustive examination of all weight combinations for the Eros distance

In this section, we want to test whether satisfactory accuracy results are obtained with a modified version of the Eros distance. In particular, we considered all the possible combinations of weights relative to the four classes of interest and chose the best one. It is reported in Table 9.12.

Table 9.12: The best weight combination for the Eros distance obtained after an exhaustive examination of all the possible combinations on testing data

Class	*Weights*
Token Contract	In-degree: 0.15 Out-degree: 0 In-transactions: 0.30 Out-transactions: 0 In-value: 0.30 Out-value: 0 PageRank: 0.12 Clustering-coefficient: 0.13
Exchange	In-degree: 0.12 Out-degree: 0.16 In-transactions: 0.12 Out-transactions: 0.16 In-value: 0.12 Out-value: 0.16 PageRank: 0.11 Clustering-coefficient: 0.09
Bancor	In-degree: 0.30 Out-degree: 0 In-transactions: 0.30 Out-transactions: 0 In-value: 0.30 Out-value: 0 PageRank: 0.06 Clustering-coefficient: 0.04
Uniswap	In-degree: 0.10 Out-degree: 0 In-transactions: 0.10 Out-transactions: 0 In-value: 0.10 Out-value: 0 PageRank: 0.34 Clustering-coefficient: 0.36

Then, we applied our classification algorithm with the modified Eros distance and this combination of weights. In Table 9.13, we report the obtained confusion matrix, while in Table 9.14 we show the values of Accuracy and Micro- and Macro-Average Precision, Average Recall and Average F1-Score.

From the analysis of these tables, we can see that the results obtained in this case are really excellent. However, the main problem with this approach is its computation time. In fact, in order to classify 4,568 testing addresses, our algorithm required 195,641 seconds. This is a much longer time than the one required by the original version of the Eros distance. While this is still acceptable for about 4,500 testing addresses, it becomes impractical as the number of the addresseses to classify starts to increase.

Table 9.13: Confusion matrix of our classification algorithm with an exhaustive examination of all the possible weight combinations for the Eros distance

	Token Contract	Exchange	Bancor	Uniswap
Token Contract	1,896	18	32	8
Exchange	21	984	24	23
Bancor	36	15	621	12
Uniswap	12	32	16	532

Table 9.14: Values of some quality metrics obtained by applying our classification algorithm with an exhaustive examination of all the possible weight combinations for the Eros distance

Metric	*Value*
Accuracy	0.97
Micro Average Precision	0.94
Macro Average Precision	0.93
Micro Average Recall	0.94
Macro Average Recall	0.93
Micro Average F1-Score	0.94
Macro Average F1-Score	0.93

9.2.3 Evaluating our approach with our version of the Eros distance

In this section, we want to test the performance of our classification algorithm with our version of the Eros distance. Specifically, in this case, the weights to be adopted for the computation of the Eros distance are determined by means of our heuristics described in Algorithm 5. In applying it, we set the value of the parameter *step* to 0.05, which has proven to return an excellent tradeoff between accuracy and computation time.

Proceeding in this way, we obtained the weight combination shown in Table 9.15. Comparing it with the optimal one, provided in Table 9.12, we can see that the differences are very small.

Then, we applied our classification algorithm, equipped with the modified Eros distance and this weight combination. In Table 9.16, we report the confusion matrix, while in Table 9.17 we report the values of Accuracy and Micro- and Macro- Average Precision, Average Recall and Average F1-Score.

These tables show that the goodness of our algorithm slightly degrades, compared to the one obtained by an exhaustive approach. However, it continues to be very high.

In order to classify the 4,568 testing addresses, our algorithm required 1,410 seconds. This is a longer time than the one required by the original version of the

Table 9.15: The best weight combination for the Eros distance obtained by applying our heuristics on testing data

Class	*Weights*
Token Contract	In-degree: 0.17
	Out-degree: 0
	In-transactions: 0.28
	Out-transactions: 0
	In-value: 0.28
	Out-value: 0
	PageRank: 0.14
	Clustering-coefficient: 0.13
Exchange	In-degree: 0.13
	Out-degree: 0.13
	In-transactions: 0.13
	Out-transactions: 0.13
	In-value: 0.13
	Out-value: 0.13
	PageRank: 0.12
	Clustering-coefficient: 0.10
Bancor	In-degree: 0.29
	Out-degree: 0
	In-transactions: 0.29
	Out-transactions: 0
	In-value: 0.20
	Out-value: 0
	PageRank: 0.08
	Clustering-coefficient: 0.05
Uniswap	In-degree: 0.12
	Out-degree: 0
	In-transactions: 0.12
	Out-transactions: 0
	In-value: 0.12
	Out-value: 0
	PageRank: 0.31
	Clustering-coefficient: 0.33

Table 9.16: Confusion matrix of our classification algorithm with our version of the Eros distance

	Token Contract	Exchange	Bancor	Uniswap
Token Contract	1,838	44	54	18
Exchange	33	956	31	33
Bancor	42	18	608	16
Uniswap	14	46	18	514

Eros distance. However, it is much shorter than the one required by the exhaustive approach. This is already an important result, but the most relevant fact is that this computation time does not grow exponentially with the number of classes and/or the number of features, thus ensuring the scalability of our approach.

Table 9.17: Values of some quality metrics obtained by applying our classification algorithm with our version of the Eros distance

Metric	*Value*
Accuracy	0.91
Micro Average Precision	0.91
Macro Average Precision	0.90
Micro Average Recall	0.91
Macro Average Recall	0.89
Micro Average F1-Score	0.91
Macro Average F1-Score	0.89

Another very interesting characteristic is that the user can tune the tradeoff between accuracy and computation time by simply setting the value of *step*, depending on the number of classes and features she needs to consider, the accuracy degree she desires and the time she has available. In our opinion, this tuning feature represents an additional characteristic of our approach, generally not present in the related ones proposed in the literature and that can be extremely useful in real contexts.

9.2.4 Computation time analysis

In this section, we conclude the evaluation of our approach by discussing the computation time of its steps. In particular, we consider the application of our approach on the dataset we used in this paper (Section 9.1.7). With our computational resources (see Section 9.1.6 for all details on them), the time required for the tasks of our experiments are as follows:

- The time required to build the training (resp., testing) network was 2,522 (resp., 2,734) seconds.
- The time necessary to compute the spectra of the training (resp., testing) users was 9,234 (resp., 9,624) seconds. This is the largest computation time. It was necessary because, for the computation of the spectrum of a user, it is necessary to compute the clustering coefficient of the corresponding network node, which requires most of the time indicated above.
- The time required to compute the spectra of the training and testing classes from the ones of the corresponding users is negligible.
- The time required for classifying the training (resp., testing) users adopting our version of the Eros distance was 1,242 (resp., 1,410) seconds.

Regarding these times, we observe that they are acceptable. This conclusion is also reinforced by the consideration that the class of a user is invariant, or at least

varies very slowly over time. Therefore, the classification of a user must be carried out only once or, at least, very rarely.

9.2.5 Discussion

Our approach to classify Ethereum users based on their behavior has a peculiarity that differentiates it from all the other classification approaches operating on Ethereum. In fact, it is *automatic* and, at the same time, *multi-class*. Let us now take a closer look at the importance of this peculiarity. The current approaches to classify Ethereum users are based on the analysis of users' smart contracts that they voluntarily submit to a provider of this service, such as Etherscan. However, the fraction of users thus classified is extremely low (more specifically, at the time of writing, it is equal to 0.236%). To overcome this difficulty, several automatic approaches to classify Ethereum users have been proposed. However, they are all single-class. In fact, they aim to find all users belonging to a certain class [136, 666, 630, 624, 642, 153]. They certainly represent a first response to the need for approaches capable of classifying a huge number of users. However, such an answer is still limited because, as we have seen in Section 9.1.10, more than 400 classes exist on Ethereum. And, although the most important ones are few, these approaches have been targeted for a very specific class. Therefore, they cannot be easily extended to find users of another class so as to simulate multi-class behavior by calling them multiple times, once for each class of interest. Instead, our approach is automatic, multi-class and incremental; therefore, it allows the classification of all the addresses belonging to classes whose spectrum is known. From this point of view, it solves an open problem and becomes an indispensable tool for all those applications needing user classification to operate [666, 612, 168].

All the automatic multi-class approaches for classifying blockchain users that we presented in Section **??** have many differences from the one proposed in this paper. First, they were all designed for the Bitcoin blockchain, except the ones described in [312, 606]. In principle, these could be employed on any blockchain, but were tested on a very specific one, operating on stock trading. Instead, our approach is designed to operate on Ethereum, even if its guidelines are general and can be fit to other blockchains in the future.

An important difference between our approach and the related ones proposed in the past literature lies in the fact that it introduces the concept of spectrum of a user and a class of users. In this concept, a crucial role is played by the "time" variable. Instead, this variable is not taken into account by most of the approaches seen in Section **??**, more specifically by the ones described in [328, 612, 402, 691, 518, 606]. The only approach that takes time into account is the one proposed in

[312]. However, it operates on univariate time series, assuming that there is no form of correlation between features. This assumption is very strong in reality and, if not verified, would lead to a decrease in the accuracy of the results proportional to the correlation degree of features. In our approach, the concept of spectrum allows us to consider not only the temporal evolution of features but also their correlation. In fact, we measured the correlation degree of each pair of features adopted and found that some of them are totally or partially correlated (see Section 9.1.10). As a consequence, we decided to operate on multivariate time series, instead of univariate ones. Clearly, this makes our approach a bit more complex but allows it to achieve very accurate results, as we have shown in Section 9.2.

Another very important feature of our approach concerns the measure of similarity between spectra, and thus between multivariate time series. To perform this task, we start from the Eros distance [653]. This measure is very simple and easy to implement and, at the same time, outperforms other similarity measures for multivariate time series previously proposed in the literature [653]. Regarding this, our approach makes an additional contribution. Indeed, it first shows, through some experiments, that the original Eros distance does not return satisfactory results in our context. Then, it proposes a modified version of this distance which, at the price of an acceptable increase of the computational time, manages to reach very high accuracy values, as shown in Section 9.2.

10

Extracting information from posts on COVID-19

In the last two years, we have seen a huge number of debates and discussions on COVID-19 in social media. Many authors have analyzed these debates on Facebook and Twitter, while very few ones have considered Reddit. In this chapter, we focus on this social network and propose three approaches to extract information from posts on COVID-19 published in it. The first performs a semi-automatic and dynamic classification of Reddit posts. The second automatically constructs virtual subreddits, each characterized by homogeneous themes. The third automatically identifies virtual communities of users with homogeneous themes. The three approaches represent an advance over the past literature. In fact, the latter lacks studies regarding classification algorithms capable of outlining the differences among the thousands of posts on COVID-19 in Reddit. Analogously, it lacks approaches able to build virtual subreddits with homogeneous topics or virtual communities of users with common interests.

The material presented in this chapter was derived from [255].

10.1 Methods

10.1.1 Approach to classify posts based on topics

Approach description. In Reddit, the COVID-19 disease is dealt from many points of view. Therefore, it seems useful to think about defining a classification of COVID-19 posts in Reddit based on their content. This classification cannot be exclusive because a post can belong to more than one class. Furthermore, it can be hierarchical because, by adopting different abstraction levels, two or more classes of a lower level can be grouped into a class of a higher level.

Given the novelty of the COVID-19 disease and the various terms used to describe it, the definition of the initial class hierarchy can be only semi-automatic. In other words, the support of the human expert is needed to identify at least the leaf classes of the hierarchy. The human expert examines the main keywords associated with the posts as they are derived from any text mining approach (such as the ones

described in [494, 441, 528, 363, 362]). Starting from this examination, she/he identifies the leaf classes and, then, associates a set of representing keywords with each of them. Two or more classes sharing a minimum number of keywords are considered siblings and can be "merged" into a single class at the higher abstraction level. The set of keywords of the new class will be equal to the union of the sets of keywords of the starting classes. Proceeding this way, after several abstraction levels, the model will result in a single tree, if there is at least one keyword common to all classes, or a forest of trees, if not.

Once the initial hierarchy is built, the assignment of posts to the corresponding classes can be done automatically. For this purpose, it is necessary to identify a measure of similarity between the keywords of a post and those of a class, and a mechanism that, based on this measure, decides whether or not a post belongs to a certain class. As far as the measure of similarity is concerned, we thought to adopt the Jaccard coefficient taking the semantic relationships (e.g., synonymies, homonymies) between keywords into account.

In particular, if CS_i indicates the set of keywords of the class C_i, and PS_k denotes the set of keywords of the post P_k, the enhanced Jaccard coefficient J_{ik}^+ between C_i and P_k is defined as:

$$J_{ik}^+ = \frac{|CS_i \sqcap PS_k|}{|CS_i \sqcup PS_k|}$$

where $\sqcap$ (resp., $\sqcup$) denotes the enhanced intersection (resp., union) between the keywords in such a way as to take into account the synonymies and homonymies as stored in a suitable thesaurus, like Babelnet [458]. J_{ik}^+ belongs to the real interval $[0, 1]$.

We are now able to define an automatic approach for determining whether a post belongs to a class. Since multiple class memberships are allowed, i.e., a post can belong to more than one class, for leaf classes it is sufficient to define a threshold th_J and to establish that P_k belongs to C_i if $J_{ik}^+ \geq th_J$. The higher th_J, the fewer the classes which P_k will belong to. From a theoretical point of view, it is appropriate for the value of th_J to be low in order to encourage a post to belong to multiple classes. Based on this idea, we performed experiments to find the optimal value of this threshold. Due to space constraints, we do not report such experiments in detail. We only say that at the end of them we found that the optimal value of th_J is 0.25. If C_i is a non-leaf class, P_k belongs to C_i if it belongs to at least one child of C_i.

The content of a social network is very dynamic, so a classification cannot remain unchanged over time. As new posts arrive, new keywords emerge, which can stimulate the appearance of new classes. At the same time, other keywords become obsolete, which can lead to the disappearance of some classes or their inclusion into

others. Finally, two or more classes may have to be merged into one class because they have become very similar. All this led us to the definition of an incremental and automatic algorithm for updating the original classification. This algorithm is important because it is well known that one of the weak points of most hierarchical clustering or hierarchical classification algorithms is the lack of backtracking [294]. Instead, our approach is provided with some backtracking mechanisms and, therefore, is able to fix any possible classification error performed in the past and to support the evolution of the hierarchy over time.

In order to operate, our algorithm needs a parameter capable of measuring the cohesion degree of a class. Since a class is determined by its keywords, it is necessary to identify a measure of cohesion among keywords. This problem has been highly investigated in the past literature on information systems [513]. A possible solution is to associate a similarity coefficient σ_{st} with each pair of keywords (kw_s, kw_t), derived through an appropriate thesaurus such as WordNet [442] and, then, to solve a maximum weight matching problem. This maximizes the average α of the similarity coefficients of the pairs of the class keywords, with the constraint that each keyword can belong to at most one pair. We will not dwell on the formalization and technical details of this solution; the interested reader can find it in [487, 196]. Here, it is sufficient to say that, given a class C_i characterized by a set CS_i of keywords, the average α_i described above is an indicator of the cohesion degree of C_i. α_i belongs to the real interval $[0, 1]$; the higher α_i, the higher the cohesion.

We are now able to describe our (automatic) algorithm for incremental update. It receives a current classification (which consists of a hierarchy of classes and the assignments of the past posts to them) and a new post P_q to be classified and returns the updated classification. First, for each leaf class C_i of the hierarchy, it computes the enhanced Jaccard coefficient J^+_{iq} between the sets of keywords of C_i and P_q. After the computation of all the enhanced Jaccard coefficients between P_q and any leaf class of the hierarchy, three cases might happen, namely:

- $J^+_{iq} < th_J$ for each leaf class C_i. This means that P_q cannot be assigned to any class. This can happen under two very different circumstances, namely: *(i)* P_q is the first post on a new topic, in which case it is likely that, in the near future, several other posts will contain the same keywords as P_q; *(ii)* P_q is an outlier, i.e., a post totally detached from the others. To deal with both cases, our algorithm adds a new leaf class C_q to the hierarchy. The keywords of C_q will be the ones of P_q. Clearly, P_q is assigned to C_q. At this point, our algorithm activates a counter that increases each time a new post is examined. Before this counter reaches a maximum value c_{max}, if at least another post is assigned to C_q, then the latter is kept in the hierarchy and will gradually grow, giving rise to its ancestors in the

hierarchy. On the contrary, if none of the c_{max} posts following P_q is assigned to C_q, then P_q was an outlier, so C_q is removed and P_q remains unclassified.

- $J^+_{iq} \geq th_J$ for exactly one leaf class C_i. In this case, P_q is assigned to C_i and all the keywords of P_q not present in C_i are associated with that class. At this point, the cohesion coefficient α_i of C_i is re-computed. If this is less than a certain threshold $th_{\alpha_{min}}$, then we proceed to split C_i into two classes by solving an optimization problem that aims at maximizing the cohesion coefficient of the two classes thus obtained. The two classes have the same parent class, and this class will be the original parent class of C_i. This will result in the potential assignment of new keywords to it, which could lead to a decrease of its cohesion degree. If this were to happen, it would be necessary to split the parent class too. In the worst case, this process may continue until the root of the hierarchy has to be split. Note that this is a first backtracking mechanism present in our algorithm. It solves the problem regarding the existence of an excessively heterogeneous class. This could happen because of an error in the construction of the initial hierarchy or because the objects incrementally assigned to it have made its heterogeneity level greater than the maximum acceptable value.
- $J^+_{iq} \geq th_J$ for two or more classes of the hierarchy. In this case, P_q is assigned to all classes for which the above condition is true. Let C_i and $\overline{C_i}$ be the classes having the maximum and submaximum values of the enhanced Jaccard coefficient with P_q, respectively. Our algorithm verifies if C_i and $\overline{C_i}$ continue to be sufficiently distinct or must be merged into a single class. For this purpose, it computes the cohesion coefficients α_i of C_i, $\overline{\alpha_i}$ of $\overline{C_i}$ and α^* of the class C^* that would be obtained by merging C_i and $\overline{C_i}$. If $\alpha^* > \alpha_i$ and $\alpha^* > \overline{\alpha_i}$ then C_i and $\overline{C_i}$ are merged into C^*. This merge process could propagate to the parents of C_i and $\overline{C_i}$ and, gradually, to the ancestors, possibly reaching the root of the hierarchy. For each class which P_q is assigned to, it is necessary to make the check seen in the previous case to verify if that class, after the assignment of P_q to it, is sufficiently cohesive or must be split into two classes. In this last case, the same tasks described for the previous case must be performed. This is a second backtracking mechanism present in our algorithm. It is activated when there are two classes similar to each other that should be merged into a single class. This could happen because of an error in the construction of the hierarchy or because the objects incrementally assigned to the two classes have made them more and more similar to each other.

Once verified in which scenario it falls, our approach proceeds accordingly and obtains a new version of the hierarchy. In Figure 10.1, we report a flowchart that schematizes the behavior of our approach.

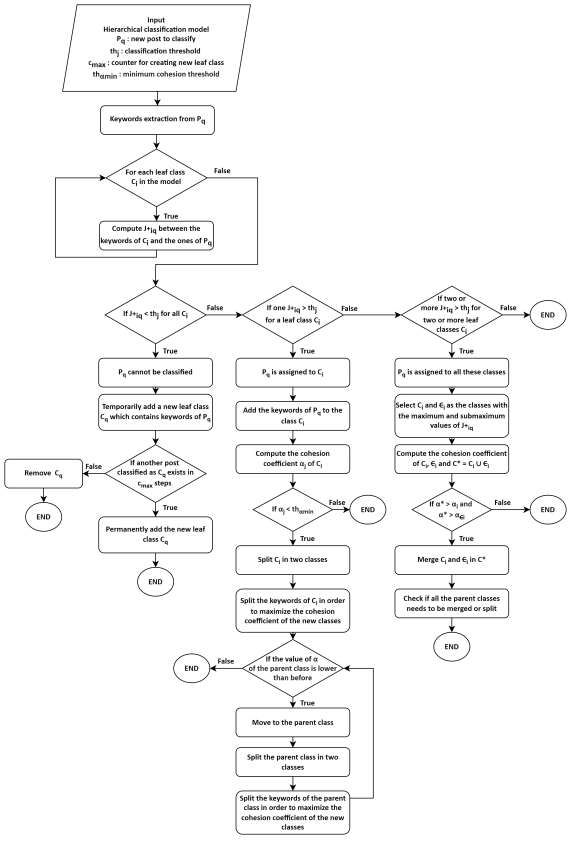


Fig. 10.1: A flowchart representing our approach to classify posts based on topics

Approach discussion. An important element of the previous algorithm is represented by the two backtracking mechanisms, which allow the correction of possible problems in the hierarchy. In principle, these problems may exist due to construction errors or, more likely, because the incremental assignment of new posts to classes has led to the need of suitably restructuring the initial hierarchy. We observe that, in the literature, there are a few rare cases of a hierarchical classification algorithm provided with backtracking mechanisms. Our approach belongs to this strand. For example, in [687], an approach for hierarchical classification of a set of documents with backtracking is proposed. It assigns a document to one or more categories of a predefined hierarchy. This approach could be applied to Reddit posts as an alternative to ours. However, in our approach, backtracking mechanisms not only allow us to repair a misclassification, as done in [687], but also to modify the hierarchy, if necessary. In our opinion, this last property is important as it allows us to correct not only errors in class assignment but also errors in the hierarchy structure. Moreover, it lets the hierarchy evolve incrementally with the evolution of the posts classified in it.

Our algorithm is very rigorous, as it provides a version of the class hierarchy and post assignments to classes in real time. However, it could be computationally expensive. To reduce its computational costs, we might consider processing a set $PSet$ of posts, instead of just one post, before making any changes to the classes. Clearly the bigger $PSet$, the greater the gain in computational resources, and the greater the information loss caused by not updating classes in real time. A good trade-off we found was setting $PSet$ to the posts on COVID-19 published each day on Reddit.

10.1.2 Approach to build virtual subreddits with homogenous topics

Approach description. Network Analysis techniques play a fundamental role in this approach. However, most of the algorithms based on Network Analysis are notoriously expensive, so we had to operate on a sample of available posts, rather than on all of them. Therefore, given a sample S_i, we constructed a suitable network $\mathcal{S}_i$ supporting our approach. In particular:

$$\mathcal{S}_i = \langle N_i, E_i \rangle$$

N_i is the set of nodes of $\mathcal{S}_i$. There is a node n_{i_j} for each post P_{i_j} of S_i. Since there is a biunivocal correspondence between the nodes of N_i and the posts of S_i, in the following of this section, we will use these two terms interchangeably. E_i represents the set of arcs of $\mathcal{S}_i$. There is one arc $(n_{i_j}, n_{i_k}, w_{jk})$ if there is at least one keyword in common between the posts P_{i_j} and P_{i_k}[1]; w_{jk} denotes the corresponding number of common keywords.

Our approach is parametric with respect to an integer number X. Given the sample S_i, it considers the set KS_i of the X keywords most present in the posts of S_i and builds (at most) X virtual subreddits, $R_1, \cdots, R_X$, one for each keyword. Given the j^{th} keyword $kw_j \in KS_i$, the corresponding virtual subreddit R_j will have associated a set RS_j of keywords (obviously including kw_j) and a set $PostS_j$ of posts. Our approach proceeds as follows:

- For each keyword $kw_j \in KS_i$:
  - It builds the subreddit R_j by initially setting $RS_j = \{kw_j\}$ and $PostS_j = \emptyset$.
  - It builds the set $\overline{KS_j}$ of the X keywords that co-occur most frequently with kw_j in the posts of S_i.
  - It sets $RS_j = RS_j \cup \overline{KS_j}$.

[1] The identification of common keywords takes synonymies and homonymies into account by following the thesaurus-based approach mentioned in Section 10.1.1.

– For each keyword $kw_{j_h} \in \overline{KS_j}$: *(i)* it builds the set $\overline{KS_{j_h}}$ of the X keywords that co-occur most frequently with kw_{j_h} in the posts of S_i; *(ii)* it sets $RS_j = RS_j \cup \overline{KS_{j_h}}$.

Note that, once we arrive at the keywords of the set $\overline{KS_{j_h}}$, we do not proceed with finding other keywords that co-occur with them. From the Network Analysis point of view, this means that we stop at the neighbors of the neighbors of kw_j. This practice of stopping at the second separation degree is very common in Network Analysis [613], as well as in the context of the derivation of semantic similarities [487, 196]. It represents an effective answer to the need of having virtual subreddits with homogeneous themes but, at the same time, wide enough to attract many users.

- Now, our approach has identified X homogeneous virtual subreddits $R_1, \cdots, R_X$, one for each keyword of KS_i. However, it could happen that two of these subreddits, say R_k and R_h, are very similar to each other, in the sense that they share most of the associated keywords (and, consequently, of the assigned posts). In this case, it would be better to merge R_k and R_h into a single subreddit R_{kh}. To make this verification and, if necessary, to merge R_k and R_h, our approach proceeds as follows:
  – Let RS_k and RS_h be the set of keywords of R_k and R_h, respectively. It computes the enhanced Jaccard Coefficient J^+_{kh} between RS_k and RS_h.
    · If $J^+_{kh} < th'_J$ then R_k and R_h are not homogeneous enough to be merged[2].
    · If $J^+_{kh} \geq th'_J$ then R_k and R_h must be merged into a single subreddit R_{kh} whose set RS_{kh} of keywords is obtained as $RS_{kh} = RS_k \cup RS_h$.
- At this point, there are at most X virtual subreddits, each with homogeneous topics sufficiently distinct from the ones of the other subreddits. The last step of our approach consists in assigning the corresponding posts to each subreddit. In this regard, we recall that a post can be assigned to more subreddits if its content is compatible with the corresponding keywords. In order to assign posts to subreddits, our approach proceeds as follows:
  – For each virtual subreddit R_k previously built:
    · For each available post P_q:
      · It computes the enhanced Jaccard Coefficient J^+_{kq} between the set RS_k of keywords associated with R_k and the set PS_q of keywords associated with P_q. If $J^+_{kq} > th_J$ then P_q is assigned to R_k.

[2] th'_J is a high threshold in such a way that if $J^+_{kh} \geq th'_J$ then RS_k and RS_h are very similar. For instance, th'_J could be set to $1 - th_J$, where th_J is the same threshold seen in Section 10.1.1.

In Figure 10.2, we report a flowchart that schematizes the behavior of our approach.

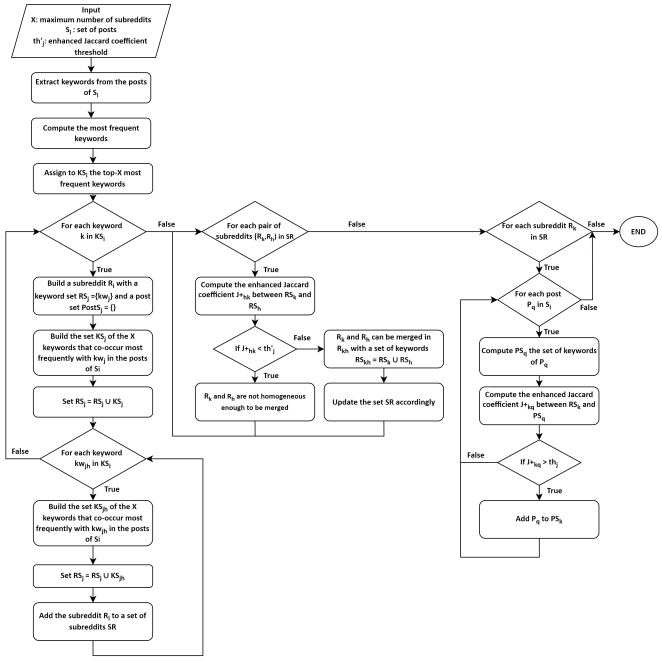


Fig. 10.2: A flowchart representing our approach to build virtual subreddits with homogeneous topics

Approach discussion. The virtual subreddits thus obtained can obviously attract users interested in finding all the posts related to a given topic in one place. Therefore, they can become very attractive not only for current Reddit users but also for new users interested in deepening a certain topic. Indeed, the former would find a new service available, the latter would find the topics of interest in Reddit in a comprehensive way and in a single place, thanks to the presence of the corresponding virtual subreddit.

It is worth pointing out that applying the approach described in Section 10.1.1 to the virtual subreddits returned by the approach described in this section could make them capable of evolving over time. Furthermore, it would be possible to build a classification hierarchy from virtual subreddits, in which these last would represent the corresponding leaf nodes.

We observe that our approach shares several similarities with document/semantic clustering methods. A discussion on these methods can be found in [536]. In this paper, the authors group them into four categories, based on Latent Semantic Analysis, lexical chains, graphs and ontologies, respectively. Our approach shares the most

similarities with graph based ones. In [536], six approaches of this family are mentioned. In the following, we give a brief description of each of them highlighting the similarities and differences with our own.

[279] describes a semi-supervised approach for clustering biomedical documents. It uses local information, derived from suitable documents, global information, derived from the MEDLINE collection, and other semantically specific information. Both the approach of [279] and ours operate by making use of keywords in similarity evaluation. However, they have some differences in that: *(i)* the approach of [279] is particularly focused on the biomedical context; *(ii)* it is semi-supervised, while ours is unsupervised; *(iii)* it imposes some constraints on the observations to be clustered. [605] defines an approach to evaluate similarities between documents in different languages. To this end, it represents multilingual documents through the concepts most commonly found in them. The clustering of documents based on concepts proposed in [605] shares some similarities with the clustering of posts based on the keywords of our approach. However, the approach in [605] was designed to analyze complex multilingual documents and to resolve translation ambiguities. Instead, our approach targets generally short texts (i.e., posts) with the goal of clustering them. Therefore, it is less general than the approach of [605] but, being more specific to a given context (i.e., Reddit posts), it can better exploit its features. [527] proposes a new approach for Multilingual Document Clustering using a tensor-based model that can handle the high dimensionality of these documents. Compared to the approach of [527], our own is more tailored to a single goal and, thus, more able to take full advantage of the characteristics of the target context. As an additional difference, the approach of [527] computes document similarities based on phrases, while our approach computes post similarities based on keywords. [526] proposes an approach that classifies a text based on the relationships present in it. To this end, it uses a graphical representation that makes the clusters easier to interpret by contextualizing their terms. In fact, the main goal of this approach is assigning a semantics to clusters. This objective is achieved by associating each cluster with its dominant topic. Both the approach of [526] and ours use keywords of the texts involved as a basis for measuring their similarity. However, they have some differences. In fact, the approach of [526] is complex, having as objective the analysis of the relationships between terms represented through graphs, which are, then, exploited to perform clustering. By contrast, our approach is tailored to posts, which can be considered very simple documents, but is capable of processing tens of thousands of them. [377] proposes an approach for extracting keywords from a text represented through a graph modeling its terms and their relationships. This approach uses a measure of centrality (e.g., PageRank) to carry out its tasks. Both the

approach of [377] and ours are designed to operate in online contexts, characterized by a large number of documents or texts to be analyzed. There are also some differences between them. Indeed, the approach of [377] uses centrality measures, which are complex to compute. Moreover, its main focus is the extraction of keywords from texts rather than the next clustering activity. [309] proposes a document clustering approach based on frequent senses. It searches for frequent subgraphs that reflect the frequent senses of a sentence. The subgraphs thus discovered are used to generate document clusters. The main difference between the approach of [309] and ours is that the former represents a sense by means of a subgraph, while the latter represents a post by means of keywords. Operating on graphs instead of on keywords takes much more time and is well suited to classify a limited number of complex documents. Instead, it is hardly applicable to our context, where there are simple, but very numerous, posts to be clustered.

In [291], the authors propose another survey for clustering semantic documents. In the following, we present the approaches described therein that shares the most similarities with our approach and, for each of them, we highlight the similarities and differences with ours. [36] proposes a clustering approach to distinguish relevant information from irrelevant one in a document. Both this approach and ours are designed to operate with many data. The main differences between them are that the approach of [36] uses ontologies and was primarily conceived for the medical field, where well defined ontologies already exist. Instead, our approach can be applied on posts about any topic, even those for which well-defined ontologies do not exist. [50] proposes a clustering approach based on frequent concepts, rather than frequent keywords. These concepts are derived from the documents through a pre-processing activity. The approach of [50] is very accurate but is suitable for a context where the number of documents to be classified is limited, which is very different from our reference context. [551] proposes an approach to classify documents based on the terms present in them and the corresponding lexical relationships. To this end, it associates a tag with each document and enriches its representation through a bag of words. Both this approach and ours are based on keywords and consider the lexical relationships involving them (in our approach this is done by using the operator J^+ instead of the operator J). The main difference between them is that the approach of [551] is designed for clustering a limited number of complex documents.

10.1.3 Approach to build virtual communities of users with homogeneous interests

Approach description. Our approach to build virtual communities of users having homogeneous interests is based on Network Analysis too. Therefore, also in this case,

we use a support social network. Specifically, given a sample S_i, we construct a social network $\mathcal{S}'_i$:

$$\mathcal{S}'_i = \langle N'_i, E'_i \rangle$$

N'_i is the set of nodes of $\mathcal{S}'_i$. There is a node n_{i_j} for each author A_{i_j} who submitted at least one post of S_i. Since there is a biunivocal correspondence between the nodes of N'_i and the authors of the posts of S_i, we will use these two terms interchangeably in this section. E'_i represents the set of arcs of $\mathcal{S}'_i$. There is an arc $(n_{i_j}, n_{i_k}, w_{jk})$ if the authors A_{i_j} and A_{i_k} used the same keyword in at least one post of S_i published by them. The weight w_{jk} of the arc indicates the number of keywords used by both A_{i_j} and A_{i_k} in some of their posts of S_i. Again, we took synonymies and homonymies between keywords into account using the same guidelines seen in Sections 10.1.1 and 10.1.2.

A first issue to address in the definition of our approach is to find a rule allowing us to identify bots (i.e., automatic Reddit users that posted news crawled from different sources). For this purpose, we analyzed the behavior of bots in Reddit and observed that they generally had a high number of keywords associated with them. Therefore, we decided to consider as bots all those authors who had more that B keywords associated with them. We carried out some tests to identify the optimal value of B and found that it is equal to 8.

Knowing the number of keywords in each arc is an important starting point to reach our goal. However, it is not sufficient. Actually, it is necessary to go in more detail considering the specific sets of keywords associated with network arcs. As a matter of fact, going to this level of detail, we observed that some sets of keywords were repeated in many arcs. This fact is important because it represents the key to construct our virtual communities of users with homogeneous interests [507]. In fact, in principle, all the nodes connected by arcs having the same set of keywords could be regarded as a community of users sharing the same set of interests.

Starting from this reasoning, our approach operates as follows:

- It identifies all the sets of keywords associated with the network arcs.
- It removes the sets of keywords consisting of less than three elements, because we considered them insignificant as indicators of common interests for a community of users.
- It removes the sets of keywords occurring less than three times because we believe that, with such a low number of occurrences, the coincidence of interests between authors expressed by them could be incidental.
- It computes the distribution of the remaining sets of keywords against the number of occurrences.

- It selects all the sets of keywords belonging to the first quartile of the distribution determined in the previous step. For each of these sets, it constructs the subnetwork consisting of only the arcs belonging to it. The nodes of this subnetwork represent a community of users with homogeneous interests defined by the keywords of the set.

In Figure 10.3, we report a flowchart that schematizes the behavior of our approach.

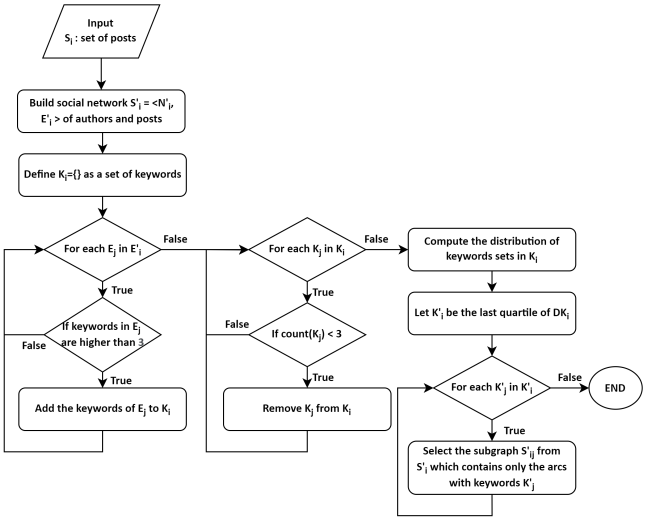


Fig. 10.3: A flowchart representing our approach to build virtual communities of users with homogeneous interests

Approach discussion. Each subnetwork represents an output of our approach and, therefore, a virtual community of users with homogeneous interests. The virtual communities thus obtained can be useful to create a collaborative filtering recommender system aiming at suggesting to a user other ones with similar interests. Moreover, our approach could be adopted by Reddit itself to propose a new functionality aiming at creating communities of users with common interests [472]. Again, we note that applying the approach described in Section 10.1.1 to the virtual communities of users returned by this approach could make returned communities able to evolve over time. Moreover, also in this case, it would be possible to build a classification hierarchy from the virtual communities. These last would represent the leaf nodes of the hierarchy.

The approach described here shares several similarities with the approaches to cluster a node-attributed network or a semantic document network.

A survey on community detection methods in node-attributed social networks can be found in [171]. Among the approaches described in this survey, the ones closest to ours are those presented in [18] and [78]. In [18], the authors propose a graph embedding approach to cluster content-enriched graphs. The idea behind this approach is to embed each node of a graph in a continuous vector space, in which structural and attributive information located at the vertices can be encoded into a unified latent representation. Analogously to our approach, the one of [18] considers the graph structure during clustering activities. In [78], the authors propose a community detection and characterization algorithm that includes the contextual attribute information of graph nodes. Its goal is to compute the context of communities and discover new ones. For this purpose, it uses a coordinate-based algorithm that updates the community label assignment of nodes. Analogously to our approach, it uses the Jaccard coefficient to evaluate the context of a node. The way the approaches of [18] and [78] operate allows them to achieve high accuracies. However, they are heavy for a context like ours characterized by very simple graphs but with a huge number of nodes and arcs. From this point of view, our approach, which considers only the structure of the graph and very little other information, is lighter and is able to process even graphs with tens of thousands of nodes, which are those of interest for our context.

In [93], the authors propose a survey on approaches to clustering the nodes of a graph with attributes. Both the approaches described in [93] and ours focus on finding homogeneous communities within the network. However, there are important differences between them. Indeed, the approaches described in [93] handle multi-dimensional graphs whose nodes and arcs can have attributes. This makes these approaches particularly suitable in handling very complex contexts where they prove to be very accurate. However, the processing times required by them are high; so, they cannot be applied in presence of large networks, such as those characterizing our scenario.

10.1.4 Dataset description

The dataset we used in the activities described in this paper was derived from the `pushshift.io` website, which is one of the main data repositories related to Reddit content. Specifically, `pushshift.io` collects Reddit posts and comments and provides a suitable website and an API for accessing them. It simplifies the query process of historical Reddit data. Furthermore, it provides several features, like a full-text search on comments and submissions. Overall, it stores all the posts and comments published on Reddit from June 2005 to today [65]. Leveraging the API provided by it, we downloaded all the posts published in Reddit from January 9^{th}, 2020

to April 30^{th}, 2020. Then, we stored them in a `.csv` file. Afterwards, we performed a set of cleaning operations, aimed to obtain a dataset ready for our analyses. Specifically, we maintained all the posts whose title contained the words "covid" and/or "coronavirus". Then, we deleted the posts consisting only of images and videos. Finally, among the remaining posts, we selected only the ones having a title written in English. To identify them, we leveraged the English corpus available in the `nltk` (i.e., Natural Language Toolkit[3]) library of Python. Specifically, we iterated over each lemma of the post title and verified if it was present in the corpus. If all the lemmas of the post title satisfied this condition, we considered the corresponding title as written in English and added it to our dataset. This last task aimed to avoid working with a multi-language dataset, which was out of our scope. At the end of these cleaning operations, our dataset consisted of 2,498,768 posts. For each post we considered the following features:

- `id`: the post's identifier;
- `author`: the post's author;
- `title`: the post's title;
- `created`: the date the post was created;
- `subreddit`: the subreddit where the post was published;
- `num_comments`: the number of comments received by the post;
- `num_crossposts`: the number of times the post was crossposted;
- `score`: the score of the post (equal to the number of upvotes minus the number of downvotes);
- `upvote_ratio`: the ratio of upvotes to the total number of votes.

The number of subreddits involved is 70,280 while the number of authors is 567,914. We note that the average number of authors per subreddit and the average number of posts per author are low, in that they are equal to 8.08 and 4.40, respectively. The average number of posts per subreddit is 35.55.

We performed our analyses on a server equipped with 16 Intel Xeon E5520 CPUs and 96 GB RAM. We used Ubuntu 18.04.3 as operating system. Moreover, we chose Python 3.6 as programming language, its Pandas Library to carry out ETL (i.e., Extraction, Transformation and Loading) tasks and its NetworkX library to perform network-based operations.

[3] `https://www.nltk.org/`

10.2 Results

10.2.1 Exploratory Data Analysis

Before carrying out our tests on the three approaches proposed in this paper, we performed an Exploratory Data Analysis (EDA, for short) on our dataset. To this end, we carried out the following tasks:

- Analysis of the distributions of `created`, `subreddit`, `num_comments`, `num_crossposts`, `score` and `upvote_ratio`.
- Analysis of the possible outliers and management of the possible missing values on all the features.
- Analysis of the possible correlations between the features.
- Detection of interesting patterns and models.

In the following of this section, we describe each of these tasks.

Analysis of feature distributions

In Figure 10.4, we report the distributions of `created`, `subreddit`, `num_comments`, `num_crossposts`, `score` and `upvote_ratio`. A first analysis of them highlights that the distribution of posts over time is irregular with the presence of two peaks. The first of them is at the end of January, the period when the COVID-19 epidemic reached its peak in China, South Korea, and other Asian countries. The second peak, much higher than the first, is around Mid-March, when the virus began to spread enormously in Europe. The distributions of posts against subreddits, comments, crossposts, score and upvote ratio follow power laws.

Analysis of possible outliers and management of missing values

As we mentioned previously, our dataset was downloaded from `pushshift.io`. Reddit data undergoes ETL activities before being stored in that repository. As a result, there are no missing or incorrect values (e.g., a negative number of comments) in `pushshift.io` and, therefore, in our dataset. Any other value assumed by one of the features of our interest (e.g., a very high value of the number of comments) cannot be considered in principle as an outlier, given the power law distribution characterizing them.

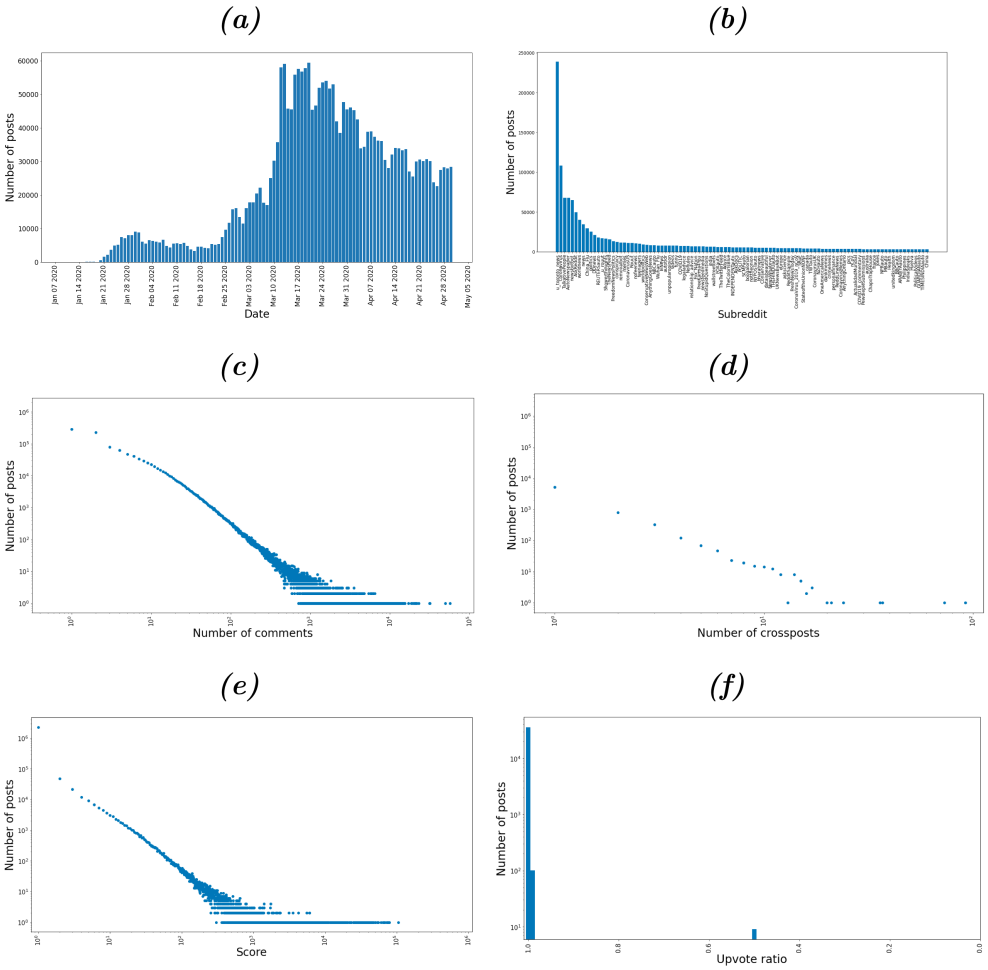


Fig. 10.4: Distribution of the features `created` (normal scale), `subreddit` (normal scale), `num_comments` (log-log scale), `num_crossposts` (log-log scale), `score` (log-log scale) and `upvote_ratio` (semi-log scale)

Analysis of the possible correlations between the features of the dataset

In Figure 12.5, we report the correlation matrix of the features of our dataset. This matrix has a row and a column for each feature. Its generic element $[i, j]$ denotes the value of the Pearson correlation between the features associated with the i^{th} row and the j^{th} column. We recall that the Pearson correlation coefficient is a parameter whose values range in the real $[-1, 1]$. When it is 1 there is a strong direct correlation; when it is -1 there is a strong inverse correlation; when it is 0 there is no correlation. From the analysis of this matrix, we can see that there is a certain correlation between `score` and `num_crossposts` and between `score` and `upvote_ratio`. The latter was expected because the percentage of upvotes influence the score of a post. Instead, the former is an unexpected information derived thanks to our analysis.

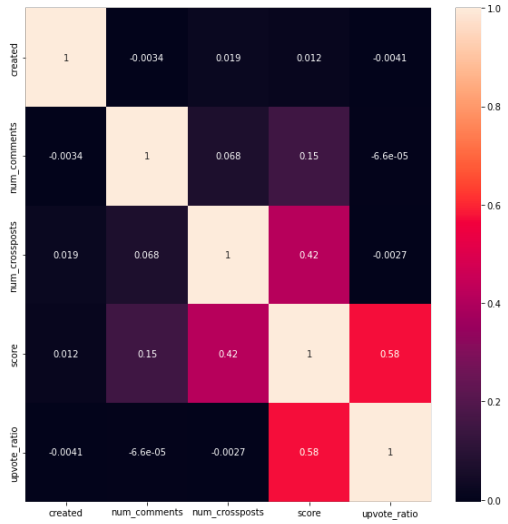


Fig. 10.5: Correlation matrix of the features of our dataset

Detection of interesting patterns and models from the dataset

As a final Exploratory Data Analysis task on our dataset, we performed a search for patterns and models that might be useful both for understanding the data available and for the next experiments.

First of all we computed the distribution of authors against posts. It is shown in Figure 10.6. This figure suggests us that it follows a power law. Recall that a quantity is said to follow a power law when the probability of measuring a particular value of it varies inversely as a power of that value. This distribution is also known as Zipf's law or Pareto Distribution [464]. It can be characterized through two parameters, namely: α, which represents the steepness of the curve, and δ, which denotes the smoothness of the slope change. In this specific case, the presence of the power law distribution means that very few authors submit a very high number of posts, while most authors submit a very little number of posts. In order to quantitatively confirm that the distribution of Figure 10.6 follows a power law, we ran two different Kolmogorov-Smirnov tests on it. The first one was based on the null hypothesis H_{01} = "The distribution is log-normal", the second one on the null hypothesis H_{02} = "The distribution is power law". We found that, given 567,914 observations, the critical value $D_{crit} = 0.025$. The first test returned a statistic $D_1 = 0.42$, with a p-value = 0.31, which led us to reject H_{01}. The second test returned a statistic $D_2 = 0.023$, with a p-value = 0.018, which confirmed H_{02}. In conclusion, we could say that our distribution follows a power law, in particular a Type 1 power law. Then, we computed its α and δ parameters and obtained that $\alpha = 2.1157$ and $\delta = 0.0201$.

By operating in the same way, we also computed: *(i)* the distribution of posts against subreddits; *(ii)* the distribution of authors against subreddits; *(iii)* the dis-

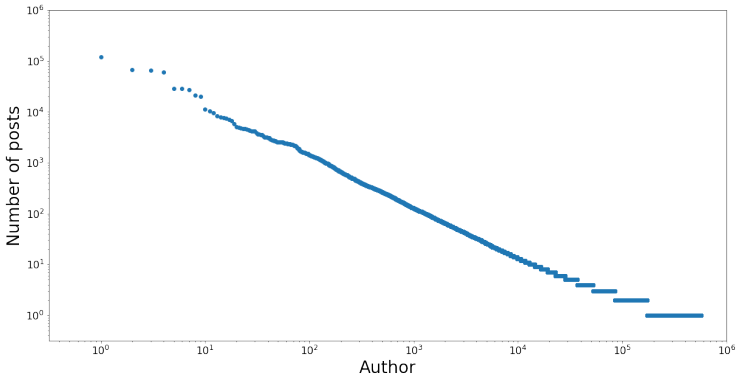


Fig. 10.6: Distribution of authors against posts (log-log scale)

tribution of posts against score; *(iv)* the distribution of comments against posts; *(v)* the distribution of crossposts against posts. We verified that all these distributions follow a power law.

The fact that the distributions of posts against score, number of comments and number of crossposts follow a power law could lead us to be pessimistic about the overall quality of published posts. Actually, this is not necessarily the case, because the power law distribution is the most common one in social networks [613]. Therefore, we decided to perform a further verification by computing the fraction of posts with an `upvote_ratio` less than 1. We saw that only 110 posts of the 2,498,768 examined ones (i.e., the 0.00044% of them) have an `upvote_ratio` less than 1. This confirms the validity of our conjecture that we should not be pessimistic about the results on posts previously obtained. All in all, the vast majority of the posts on COVID-19 were appreciated by the Reddit community.

So far we have considered four indicators of post quality and we have seen that three of them follow a power law, while the fourth one is almost always positive. We found it very interesting to check if the posts with the highest values for each of the four indicators were always the same or not. For this reason, we selected the top 500 posts for each quality parameter and computed their intersection. We could see that it contained only 13 posts. This result is very important because it tells us that there are no absolute best posts; instead, the various quality parameters capture different aspects. The only intersection worthy of attention regarded the top 500 posts with the highest score and the top 500 posts with the highest number of crossposts. In this case, we obtained that the intersection included 158 posts. This is not surprising because Figure 12.5 shows that there is a fairly high correlation between these two features.

After carrying out structural analysis, our attention focused on content. To this end, we considered the titles of the posts and carried out a lemmatization activity on

them, removing stop words and punctuation marks. After these tasks, we computed the number of occurrences for each keyword. In Figure 10.7, we report the most frequent keywords along with the corresponding number of occurrences.

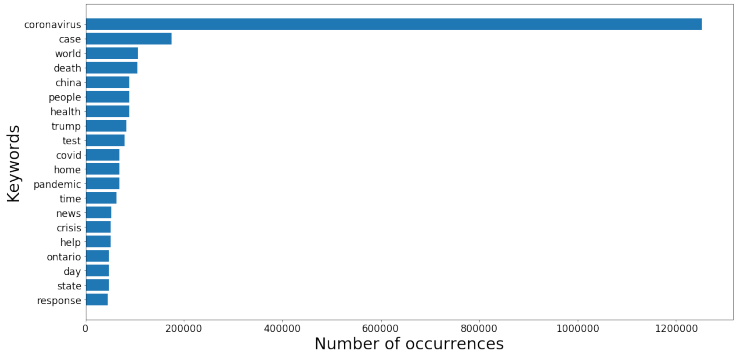


Fig. 10.7: Most frequent keywords in post titles and corresponding number of occurrences

As we can see from this figure, there is a keyword (i.e., "Coronavirus") that is by far the most frequent one. Actually, this information is quite obvious, and therefore not very significant. Instead, if we consider all the other keywords in the same figure, we can observe that most of them are characterized by a comparable and high number of occurrences. This reveals that COVID-19 is dealt within Reddit from various points of view, from health to economy, from politics to technology, and so on. This property can represent an empirical justification of the classification approach described in Section 10.1.1.

Finally, as a last task, we carried out the clustering of the keywords described above. First, we trained a FastText [327, 87] word embedding model in order to have a 100-dimensional vector representation of the keywords. Then, we used the elbow method to identify the recommended number of clusters and found that this number is equal to 5. In order to observe clusters in the bi-dimensional plane, we computed the Principal Component Analysis [640] of the word embedding vectors. We report the resulting scatter plot in Figure 10.8.

This figure is interesting because it reveals how keywords can be grouped in very homogeneous clusters. This property can represent an empirical justification of the approaches illustrated in Sections 10.1.2 and 10.1.3.

10.2.2 Approach to classify posts based on topics

The first step of our experimental campaign for evaluating this approach was the construction of the initial classification. For this purpose, we used all posts in our dataset from January 9^{th}, 2020 to March 31^{st}, 2020. To perform this classification, we

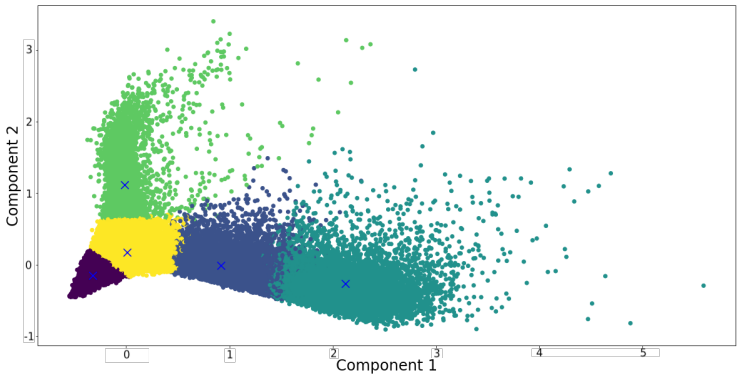


Fig. 10.8: Clustering of the keywords derived from post titles

required the support of a human expert. She was a sociologist who has been working in the field of Social Network Analysis for more than 10 years. She has been following the dynamics of information diffusion on Reddit for more than 6 years and on other popular online social networks (in particular, Facebook and Twitter) since the beginning of her work. The sociologist was supported by an epidemiologist, in interpreting technical medical terms found in some posts. We were very careful in selecting the human expert and her consultant epidemiologist, because we were aware that their decisions were very important since they would represent the ground truth in the evaluation of our approach. The initial classification is shown in Figure 10.9.

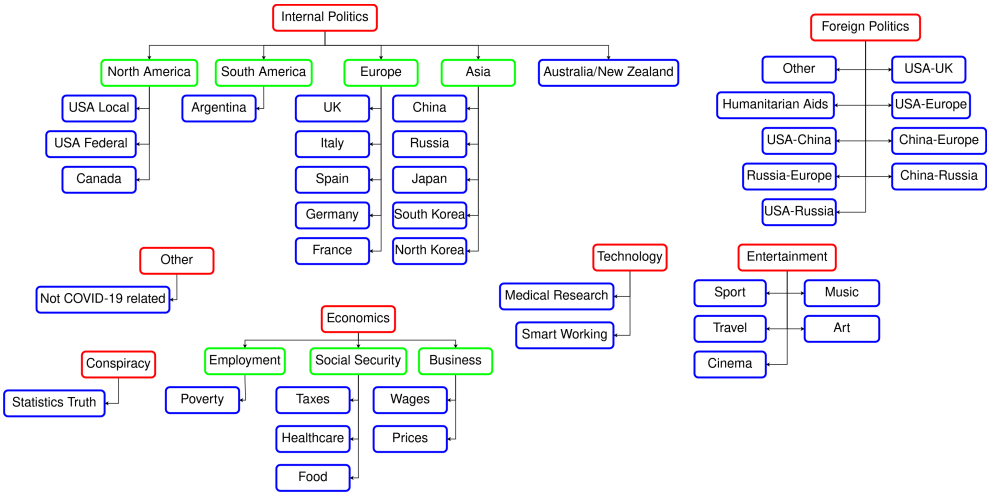


Fig. 10.9: The initial classification for the posts on COVID-19 in Reddit

With regard to it, we have the following parameter values: *(i)* number of posts available: 1,745,073; *(ii)* number of leaf classes: 40; *(iii)* number of posts assigned to at least one class: 1,605,347 (equal to 91.99% of all the posts available); *(iv)* average number of keywords associated with the leaf classes: 11.45; *(v)* average number of classes a post was assigned to: 4.07.

After this initial classification, we provided our algorithm with the posts on COVID-19 published in Reddit in April 2020. We carried out a session of the algorithm for each day of April. During each session, we gave in input the classification of the previous day and the set of posts on COVID-19 published in the current day. The classification obtained at the end of April is shown in Figure 10.10.

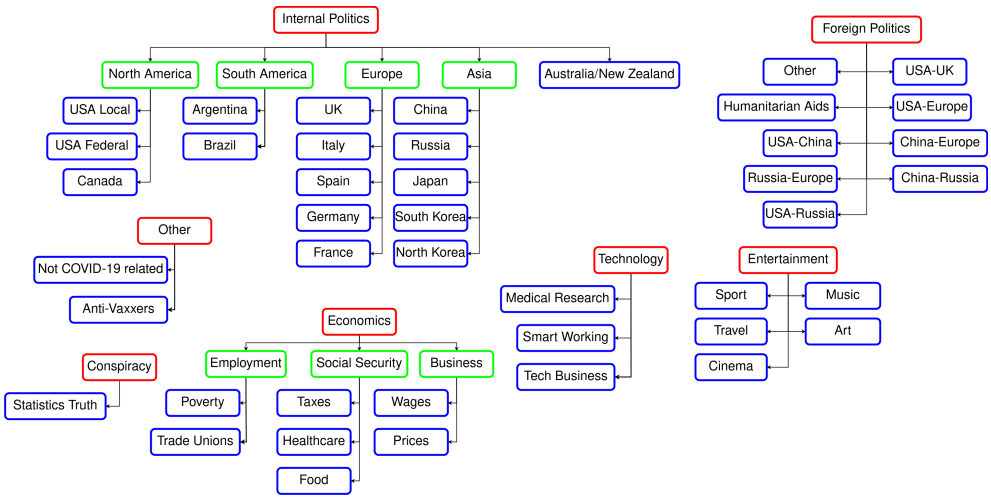


Fig. 10.10: The final classification for the posts on COVID-19 in Reddit

With regard to this final classification, we have the following parameter values: *(i)* number of posts available: 2,498,768; *(ii)* number of leaf classes: 43; *(iii)* number of posts assigned to at least one class: 2,396,744 (equal to 95.92% of all the posts available); *(iv)* average number of keywords associated with the leaf classes: 11.25; *(v)* average number of classes a post was assigned to: 4.26.

To evaluate the quality of the classification returned by the proposed algorithm we adopted the classic parameters employed in these cases, namely Precision, Recall and F-Measure [310]. In order to carry out these measurements, we used the decisions of the human expert as the ground truth. However, the processing capabilities of the human expert are limited, so it was not possible to operate on all the posts available, but only on a subset of them. Therefore, we randomly selected two samples, S_1 and S_2, each containing 500 posts of the initial classification. We also considered two samples, S_3 and S_4, each containing 500 posts of the final classification.

Given the sample S_h, $1 \leq h \leq 4$, the Precision denotes how many of the post assignments to classes made by our approach were also made by the human expert. The Recall indicates how many of the post assignments to classes made by the human expert were also made by our approach. The F-Measure is the harmonic mean

of Precision and Recall. The values of Precision, Recall and F-Measure for the four samples under consideration are shown in Table 10.1.

Sample	*Parameter*	*Value*
S_1	Precision	0.92
	Recall	0.86
	F-Measure	0.89
S_2	Precision	0.94
	Recall	0.85
	F-Measure	0.89
S_3	Precision	0.97
	Recall	0.94
	F-Measure	0.95
S_4	Precision	0.96
	Recall	0.97
	F-Measure	0.96

Table 10.1: Precision, Recall and F-Measure for the four samples under consideration

The analysis of this table reveals that:

- Our approach returns very accurate results, with both the initial and the incrementally updated classifications. Regarding these results, we observe that the values obtained using our approach are very high compared to those generally obtained when content mining techniques are adopted. In our opinion, this is caused by two reasons. The first is that our approach is not completely automatic because the leaf classes of the initial hierarchy are determined with the support of the human expert who evaluates, and possibly corrects, the results produced by the text mining algorithm. While the presence of the human expert has a negative impact on timing, there is no doubt that it can have a very positive impact on accuracy. The second reason is that, since the posts used for training were published from January 9^{th}, 2020 to March 31^{st}, 2020, while those used for testing were published in April 2020, it is plausible that there is a strong similarity between the training and testing data.
- The results obtained are stable because, if we take two different samples for each classification, they change very little. More specifically, if we consider the two samples S_1 and S_2, both derived from the initial classification, we have that: *(i)* Precision is always very high, above 0.90; its variation occurring when switching from S_1 to S_2 is 2.18%. *(ii)* Recall is always high, above 0.80; its variation occurring when switching from S_1 to S_2 is 1.17%. *(iii)* F-Measure is always high, equal to 0.89, and does not change when switching from S_1 to S_2.

We now consider the samples S_3 and S_4 both derived from the final classification. We have that the variation of Precision (resp., Recall, F-Measure) occurring when switching from S_3 to S_4 is 1.03% (resp., 3.09%, 1.04%).

As we can see, when we switch from S_1 to S_2 or from S_3 to S_4, the variations in the values of all parameters are negligible.

- Incremental updates allow our approach to obtain even more accurate results, especially for Recall. This last fact is not surprising because updates are made on the basis of the posts published. Indeed, if we compare the values of the parameters before and after classification, we can see that they always show an improvement. In particular: *(i)* Precision increases by 3.76%, passing from an average value of 0.930 to an average value of 0.965; *(ii)* Recall increases by 11.70%, passing from an average value of 0.855 to an average value of 0.955; *(iii)* F-Measure increases by 7.30%, passing from an average value of 0.890 to an average value of 0.955.

Everything we have seen in this section allows us to conclude that our approach is really capable of classifying posts related to COVID-19 and of keeping this classification updated.

10.2.3 Approach to build virtual subreddits with homogeneous topics

Analogously to what we performed for the experiments related to the previous approach, we decided to select two samples randomly, in order to verify whether the results we will obtain are stable. In particular, we considered two samples, S_1 and S_2, each including 52,352 randomly selected posts. Their main characteristics are reported in Table 10.2. Figures 10.11 and 10.12 illustrate the distribution of posts against authors and comments, whereas Figure 10.13 reports the trend of the number of posts over time. As we can see, despite the total randomness they were built with, the differences between the two samples are very low. Therefore, it was reasonable assuming that the results we obtained from them would have been stable. In any case, we did not trust this hypothesis alone but, for each result obtained, we made the appropriate stability check to see if it was very similar in the two samples.

After building the two networks, we computed some basic parameters of them. These are shown in Table 10.3.

The analysis of the values of these basic parameters provides us with valuable information. In fact, we can see that the density of $\mathcal{S}_1$ and $\mathcal{S}_2$ is low, while the corresponding average clustering coefficient is high. This kind of configuration for these two parameters is not very common in Network Analysis. In fact, usually, both of them are low or both are high. Instead, in this case, the presence of a low density

Parameter	*Value in S_1*	*Value in S_2*
Number of posts	52,352	52,352
Number of authors	23,874	23,807
Number of subreddits	7,820	7,825
Timestamp of the first post	2020-01-09 05:35:31	2020-01-09 04:59:13
Timestamp of the last post	2020-04-30 23:59:55	2020-04-30 23:58:25
Average number of comments per post	9.210	9.702
Average score of posts	5.168	4.401
Average number of keywords per post	3.031	3.053

Table 10.2: Main characteristics of the two samples S_1 and S_2

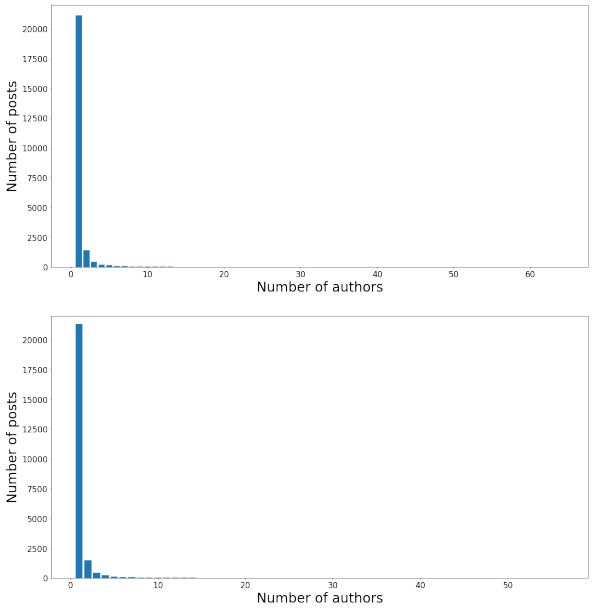


Fig. 10.11: Distribution of posts against authors for S_1 (on top) and S_2 (on bottom)

Parameter	*Value in $\mathcal{S}_1$*	*Value in $\mathcal{S}_2$*
Number of nodes	52,352	52,352
Number of arcs	29,498,151	29,332,207
Density	0.0215	0.0214
Average clustering coefficient	0.702	0.699
Average weight of arcs	1.035	1.035

Table 10.3: Some basic parameters of the networks $\mathcal{S}_1$ and $\mathcal{S}_2$

indicates that each post shares keywords with only few other ones. This can be justified considering that the topics covered in the COVID-19 posts are various, as we have seen in Section 10.2.2. The presence of a high clustering coefficient is an indicator of closed triads [613]. This implies that, if the post P_{i_j} shares keywords with the post P_{i_k}, and P_{i_k} shares keywords with the post P_{i_h}, then P_{i_j} and P_{i_h} will also share keywords [463, 229]. This suggests that, actually, there may be groups of keywords in common among a "cluster" of posts. These keywords are exactly the reference

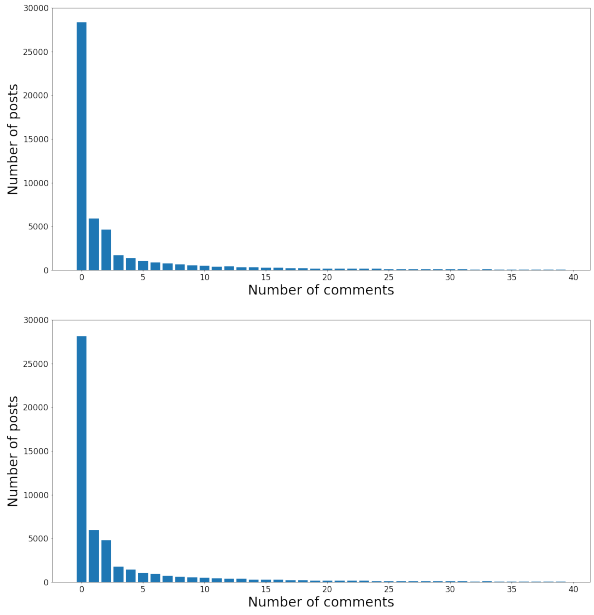


Fig. 10.12: Distribution of posts against comments for S_1 (on top) and S_2 (on bottom)

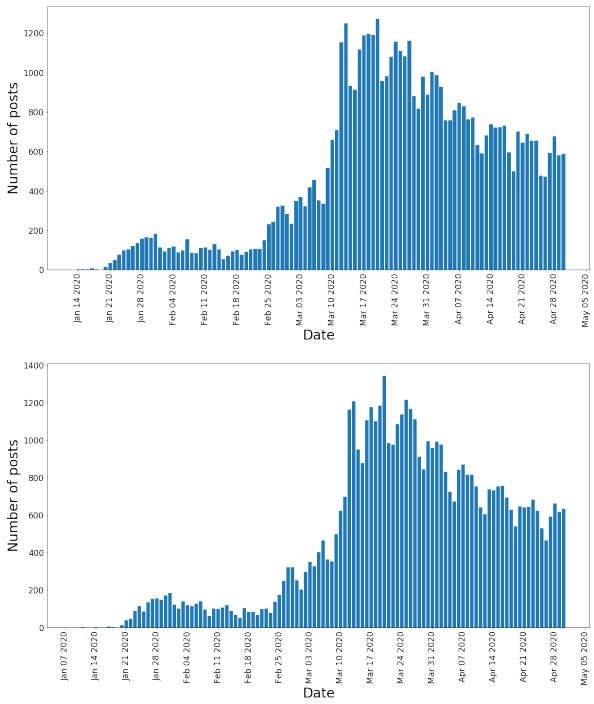


Fig. 10.13: Trend of the number of posts over time for S_1 (on top) and S_2 (on bottom)

point for the construction of virtual subreddits with homogeneous themes. In fact, the cluster of posts with the keywords in common represents the core of the virtual subreddit.

As a starting point in the definition of our approach, we determined the distribution of the keywords in the posts of the samples S_1 and S_2[4]. Indeed, our general idea is to use each of the most common keywords in the sample as an aggregation point for attracting new homogeneous keywords, together with the corresponding posts where they are present.

We applied our approach to the two samples S_1 and S_2 presented at the beginning of this section. We set $X = 10$ because, due to the steep distribution followed by the keywords characterizing the posts of the samples, the first 10 keywords already "cover" 73.33 % of the posts of S_1 and 73.03 % of the posts of S_2. The 10 keywords identified for S_1 and S_2, sorted by the number of posts in which they occur, are shown in Table 10.4.

S_1	S_2
case (6073)	case (5910)
world (5432)	world (5514)
people (4836)	health (4862)
trump (4793)	death (4811)
health (4774)	people (4784)
death (4750)	trump (4752)
china (4525)	china (4488)
ontario (4491)	ontario (4451)
test (4422)	test (4348)
home (4296)	home (4312)

Table 10.4: The 10 keywords identified for S_1 and S_2

Finally, Tables 10.5 and 10.6 report the subreddits derived from these keywords. For each subreddit, they report the set of the corresponding keywords and the number of posts assigned to it. Observe that, in Table 10.5, the subreddits R_1 and R_6 were found very similar and, according to the rules of our approach, were merged into a unique subreddit $R_{1,6}$.

We observe that many of the keywords present in Tables 10.5 and 10.6 belong to two or more virtual subreddits. In other words, each virtual subreddit in one of these tables shares keywords with one or more of the other virtual subreddits. This is due to the high clustering coefficient characterizing the networks $\mathcal{S}_1$ and $\mathcal{S}_2$ and discussed in our comments to Table 10.3. In that part, we pointed out that a high clustering coefficient implies that posts tend to be connected forming closed triads and that the posts in a triad share groups of common keywords. All this is reflected by the fact that virtual subreddits, which are ultimately sets of posts, share several

[4] Also in the computation of this distribution we removed the word "Coronavirus" (for the reasons discussed in Section 10.2.1) and took the synonymies and homonymies into account.

Virtual subreddit	*Keyword(s) from which it originated*	*Set of keywords associated with it*	*Number of assigned posts*
$R_{1,6}$	case, death	world, ontario, city, death, number, health, toronto, week, report, worker, rate, case, total, china, rise, country, home, resident, official, people, patient, test, day, toll, york	34,071
R_2	world	world, china, case, death, people, report, trump, health, virus, ontario, number, day, test, home, toronto, response, organization, time, country	18,205
R_3	people	people, health, case, death, world, worker, test, ontario, report, home, toronto, number, day, china, trump, country, virus, help, spread	18,816
R_4	trump	trump, president, news, state, house, health, world, china, case, death, report, country, people, response, government, ontario, toronto, claim, administration, test, virus	18,280
R_5	health	health, case, ontario, death, report, number, total, world, day, city, official, toronto, trump, home, minister, test, china, people, worker, help	18,278
R_7	china	china, world, case, health, death, trump, report, country, people, ontario, number, day, toll, home, toronto, virus, test, news, flight, wuhan	18,036
R_8	ontario	ontario, case, death, report, number, health, total, world, day, home, toronto, nursing, people, test, worker, hospital, patient, icu, week	15,492
R_9	test	test, ontario, case, death, home, total, health, worker, minister, world, report, people, employee, kit, china, mask, help, result, time, day, hospital, member, staff	17,833
R_{10}	home	home, death, case, report, ontario, health, number, toronto, world, nursing, resident, retirement, test, stay, life, total, worker, help, city, people, day, work	16,272

Table 10.5: The virtual subreddits constructed for S_1

Virtual subreddit	*Keyword(s) from which it originated*	*Set of keywords associated with it*	*Number of assigned posts*
R_1	case	case, death, report, ontario, home, health, world, rate, number, toronto, total, test, worker, china, state, york, people, country, organization, trump	17,499
R_2	world	world, china, case, death, people, health, report, trump, response, ontario, organization, test, number, country, state, home, rate, toronto, outbreak, time, york, day	18,871
R_3	health	health, official, case, death, trump, state, ontario, toronto, number, report, total, world, country, china, organization, time, people, home, rate, minister, test, patient, worker, outbreak	18.728
R_4	death	death, case, ontario, report, number, health, total, world, country, state, china, toll, rise, home, city, toronto, outbreak, test, worker, resident, people, organization, rate, day	17,028
R_5	people	people, health, case, world, death, ontario, organization, test, report, home, toronto, staff, china, country, time, trump, number, help, day	18,245
R_6	trump	trump, president, news, test, world, house, response, ontario, china, health, case, death, organization, report, people, call, administration, state, claim	16,762
R_7	china	china, world, health, case, death, organization, country, time, report, trump, people, ontario, number, state, home, toronto, test, virus, response, flight	18,244
R_8	ontario	ontario, case, death, report, number, health, total, world, state, home, toronto, work, people, staff, hospital, worker, test, patient, day, week, province	16,414
R_9	test	test, ontario, case, death, home, total, health, worker, world, minister, organization, report, employee, work, hospital, kit, china, toronto, result, day, time, people, staff, member, country	17,632
R_{10}	home	home, death, case, report, ontario, health, world, number, toronto, stay, life, country, response, nursing, resident, staff, total, test, worker, week, outbreak, spread, help, work, people, day, china	18,902

Table 10.6: The virtual subreddits constructed for S_2

keywords with each other. This is further amplified by the fact that our approach allows a post to belong to multiple virtual subreddits. Each virtual subreddit obtained through our approach often differs from the others not so much for the exclusivity of topics or posts but for the greater or smaller emphasis that it assigns to one or more topics with respect to others.

10.2.4 Approach to build virtual communities of users with homogeneous interests

In the experiments to evaluate this approach, we decided to work on the samples S_1 and S_2 described in Section 10.2.3.

After building the two networks, we computed some basic parameters of them. They are shown in the second and third column of Table 10.7. Their examination immediately revealed a problem, namely the great variance in the number of keywords per author. Analyzing in more detail the average values and the ones associated with the quartiles, it emerged that this variance was due to the presence of some outlier authors. Examining them carefully, we realized that they were bots (i.e., automatic Reddit users that posted news crawled from different sources), so they were not of interest for the goal we were pursuing. Therefore, we decided to remove them. The basic parameters of the new networks $\mathcal{S}_1''$ and $\mathcal{S}_2''$, obtained after the removal of bots from $\mathcal{S}_1'$ and $\mathcal{S}_2'$, are shown in the fourth and fifth columns of Table 10.7. The distribution of the arcs of $\mathcal{S}_1''$ and $\mathcal{S}_2''$ against the number of associated keywords is reported in Table 10.8.

Parameter	*Value in* $\mathcal{S}_1'$	*Value in* $\mathcal{S}_2'$	*Value in* $\mathcal{S}_1''$	*Value in* $\mathcal{S}_2''$
Number of nodes	23,835	24,084	22,204	22,457
Number of arcs	6,956,916	6,972,620	3,326,119	3,392,481
Density	0.0245	0.0240	0.0130	0.0135
Average clustering coefficient	0.7374	0.7332	0.7010	0.6960
Average weight of arcs	1.20	1.19	1.02	1.02
Average number of keywords per author	6.640	6.618	2.763	2.780
Standard deviation of the number of keywords per author	159.5453	159.8500	1.7472	1.7425
Maximum number of keywords per author	21,185	21,293	8	8
Minimum number of keywords belonging to the first quartile	4	4	4	4
Minimum number of keywords belonging to the second quartile	2	3	2	2
Minimum number of keywords belonging to the third quartile	1	1	1	1
Minimum number of keywords per authors	1	1	1	1

Table 10.7: Some basic parameters of the networks $\mathcal{S}_1'$, $\mathcal{S}_2'$, $\mathcal{S}_1''$ and $\mathcal{S}_2''$

Number of keywords	*Number of arcs in* $\mathcal{S}_1''$	*Number of arcs in* $\mathcal{S}_2''$
1	3,270,276	3,331,012
2	54,292	59,693
3	1,400	1,606
4	107	122
5	35	29
6	6	10
7	2	5
8	1	4

Table 10.8: Distribution of the arcs of $\mathcal{S}_1''$ and $\mathcal{S}_2''$ against the number of associated keywords

In order to give an idea of how our approach works, we describe its application to the networks $\mathcal{S}_1''$ and $\mathcal{S}_2''$. The sets of at least 3 keywords occurring most frequently

in the arcs of $\mathcal{S}_1''$ and $\mathcal{S}_2''$, along with the corresponding number of occurrences, is shown in Figure 10.14. Some fundamental parameters about them are reported in Table 10.9. In Figure 10.15 (resp., 10.16), we show four communities derived from $\mathcal{S}_1''$ (resp., $\mathcal{S}_2''$) to give an idea of them. In Table 10.10, we report the density and clustering coefficient of $\mathcal{S}_1''$ and $\mathcal{S}_2''$, as well as the average values of these parameters for the networks associated with the communities returned by our approach. As we can see, both the average density and the average clustering coefficient of the networks returned by our approach are higher, or much higher, than the ones of $\mathcal{S}_1''$ and $\mathcal{S}_2''$. This is an indicator that our approach is really capable of finding new user communities with homogeneous interests.

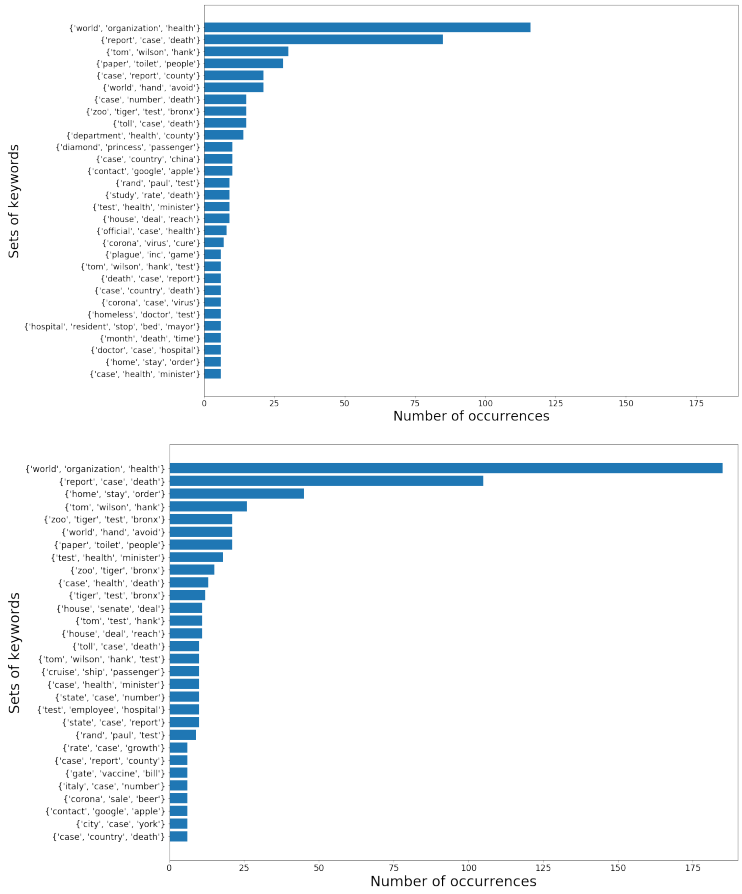


Fig. 10.14: Most frequent sets of at least 3 keywords occurring at least 3 times in the arcs of $\mathcal{S}_1''$ (on top) and $\mathcal{S}_2''$ (on bottom) and corresponding number of occurrences

Parameter	*Value in* $\mathcal{S}_1''$	*Value in* $\mathcal{S}_2''$
Average number of occurrences of the sets of keywords	7.02	6.98
Standard deviation of the number of occurrences of the sets of keywords	13.95	17.27
Maximum number of occurrences of the sets of keywords	116	185
Minimum number of occurrences of the sets of keywords belonging to the first quartile	6	6
Minimum number of occurrences of the sets of keywords belonging to the second quartile	3	3
Minimum number of occurrences of the sets of keywords belonging to the third quartile	3	3
Minimum number of occurrences of the sets of keywords	3	3

Table 10.9: Some fundamental parameters of the sets of at least 3 keywords occurring at least 3 times in the arcs of $\mathcal{S}_1''$ and $\mathcal{S}_2''$

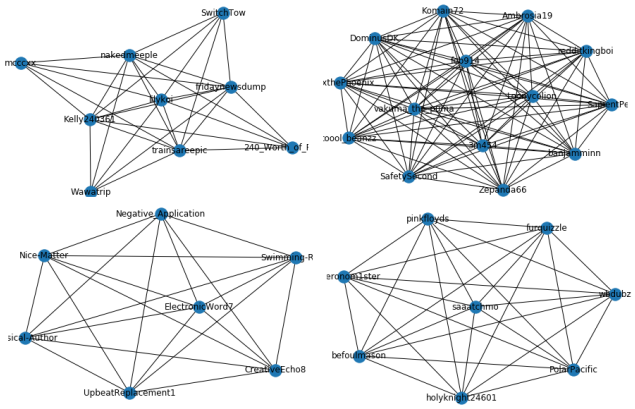


Fig. 10.15: Four communities of authors with homogeneous interests derived from $\mathcal{S}_1''$

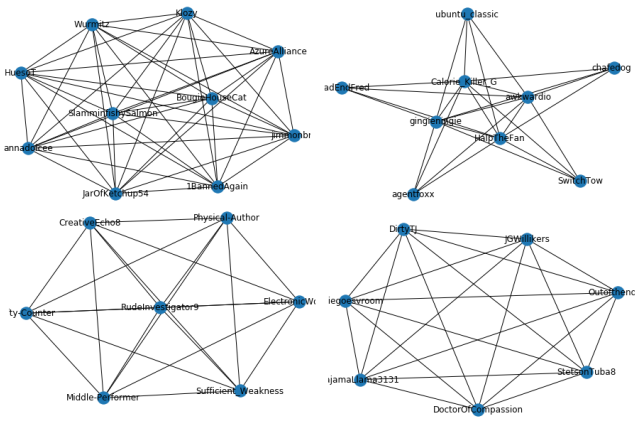


Fig. 10.16: Four communities of authors with homogeneous interests derived from $\mathcal{S}_2''$

Networks	(Average) Density	(Average) Clustering Coefficient
$\mathcal{S}_1''$	0.0130	0.7010
$\mathcal{S}_2''$	0.0135	0.6960
Networks representing author communities derived from $\mathcal{S}_1''$	0.9498	0.9267
Networks representing author communities derived from $\mathcal{S}_2''$	0.9382	0.9114

Table 10.10: Values of (average) density and (average) clustering coefficients for $\mathcal{S}_1''$ and $\mathcal{S}_2''$ and the networks associated with the communities obtained by applying our approach

11

Extracting time patterns from the lifespan of TikTok challenges

One of the key aspects that distinguish TikTok from other social media is the presence of challenges. A challenge is a kind of competition that starts when a user posts a video with certain actions and a certain hashtag and invites other users to replicate the same video in their own way. Most challenges are fun and harmless, but sometimes dangerous challenges are launched as well. The authors of these challenges use various tricks to bypass TikTok's controls. In this paper, we analyze the lifespans of some TikTok challenges and show how they are very different for non-dangerous and dangerous ones. Then, we deepen our analysis by identifying some time patterns that characterize the two types of challenges. Finally, we test the accuracy of the results obtained on a large set of challenges different from those used during the detection of time patterns. The focus of this paper is the detection of time patterns allowing the classification of challenges in dangerous and non-dangerous ones. This could represent a first step towards an approach for the early detection of dangerous challenges in TikTok.

The material presented in this chapter was derived from [3].

11.1 Methods

11.1.1 Dataset Description

As specified in the Introduction, the first step of our research consisted in building the dataset for our experiments. Indeed, to the best of our knowledge, there was no dataset of TikTok challenges already available and suitable for our goals.

To construct this dataset, we first considered a period of interest for our challenge analysis. The choice fell on the period January 2018 - April 2021 as it encompassed the most recent challenges and was sufficiently extensive. Among the challenges whose lifespan spanned this period, we considered those mentioned most frequently on Google News. From them, we had to exclude the extremely dangerous ones, already removed by TikTok, since it would have been impossible to recover their data (see below for details). Finally, among the challenges still available, we

chose some that we could assume had been highly recommended in TikTok. With regard to this, as seen in the Introduction, the recommender system underlying a user's FYP in TikTok depends heavily on her past behavior and returns highly personalized results, which vary rapidly over time. All this makes it impossible to determine with certainty which challenges have been most recommended. Furthermore, TikTok does not publicly provide detailed information about them (e.g., how many times a challenge has been recommended, its level of popularity in various countries, etc.). However, we assumed that if a challenge has many views, receives many likes and comments and has many videos associated, then it has been seen by many users and is popular. We chose challenges based on this assumption. Clearly, ours is an assumption and not an objective and incontrovertible criterion. Therefore, it is prone to sample bias. However, we believe that, with the limitations on the information made available by TikTok mentioned above, any sampling choice we made would not have eliminated this risk. Our choice was aimed at reducing it by adopting criteria and indicators that seemed reasonable to us.

Among the available challenges, we selected seven "non-dangerous" and seven "dangerous" ones. These last challenges, besides complying with all the previous constraints, meet an additional one, that is the fact that all the news that mentioned them judged them "dangerous". Before continuing with the discussion, some considerations on the concept of "dangerous challenges" are in order. First of all, as specified in the Introduction, dangerous challenges can be considered as a particular case, related to TikTok, of harmful or dangerous behaviors in social media. This is a topic much debated by researchers who study human behavior in social platforms. In the past, these authors have identified a wide range of "dangerous behaviors", such as: *(i)* harassing, discriminating [381, 250], doxing [641] and socially disenfranchising vulnerable individuals [485, 403, 355]; *(ii)* stimulating suicidal tendencies and depressive symptoms among adolescents and young adults [355, 387, 281]; *(iii)* stimulating adolescents and young adults to engage in self-harming behavior [689, 492, 525]; *(iv)* stimulating social and aggressive behaviors; *(v)* stimulating online non-suicidal self-injury; *(vi)* discussing acts of self-harm and of cyber-suicide [493, 452, 347]. Such behaviors have been inherited by what we call "dangerous challenges" in TikTok. Regarding this concept, we must however point out that the definition of "dangerous" is not necessarily objective, nor can it be taken-for-granted as widely accepted. It is also adult-centric since the people who talk about it are almost always adults. Clearly, the aim of this paper is not to propose a scientific and systematic treatment of dangerous behavior in social media. It is up to experts of human behavior, some of whom have been cited above. Our goal is the definition of computer science-based

approaches to search for "dangerous" challenges, considering the latter based on the "mainstream" views of a general public.

We are aware that media and journalistic analyses can be politically, ideologically and geographically/culturally biased. However, we want to point out that our approach, from a technical point of view, can work with any definition of "dangerous" challenge. Therefore, if a user wants to apply it considering a different definition of "dangerous" challenge, our approach still works. The only condition for it to work is that the user provides (perhaps with the support of experts on human behavior) training data that reflect the definition of "dangerous" challenge that she wants to consider.

As pointed out in the Introduction, a challenge is identified by the hashtag used to post a video related to it.

The seven non-dangerous challenges we selected are the following:

- `#bussitchallenge`: it consists of a change of clothes following the song "Buss It" by Erica Banks.
- `#copinesdancechallenge`: it consists of a series of dance movements following the song "Fly" by Aya Nakamura.
- `#emojichallenge`: it consists of imitating several emoji; it does not have an associated song.
- `#colpiditesta`: it consists of virtually hitting a soccer ball with the head; it does not have an associated song.
- `#boredinthehouse`: it consists of filming a subject, mostly an animal, in different parts of the house. The associated song is "Board in the house" by Curtis Roach.
- `#itookanap`: it consists of filming a subject, mostly an animal, sleeping. The associated song is "I Took A Nap" published by the user "gunnarolla".
- `#plankchallenge`: it consists of performing dance movements based on physical training exercises to the rhythm of a song, which is not unique.

The seven dangerous challenges we selected are the following:

- `#silhouttechallenge`: it consists of exposing the body covered by a red light filter following the song "Put Your Head On My Shoulder" by Giulia di Nicolantonio. It is considered dangerous because often the body of the author of the video is naked and the filter, being digital, can be easily removed.
- `#bugsbunny`: the authors of the corresponding videos lie down on their stomach and lift their legs upwards to show their feet sticking out of their head like the ears of a rabbit; at this point they start to move their feet to the rhythm of a song.

It is considered dangerous because it has an explicit variant in which the authors show parts of their bodies "inappropriate" for young people aged 0-18[1].

- `#strippatok`: it consists of publishing videos related to strippers (both men and women). It is considered dangerous because it deals with subjects "inappropriate" for young people aged 0-18.
- `#firewroks`: it consists of posting videos with fireworks for which the authors risk their own safety. The apparently wrong hashtag is a trick of the authors of videos to bypass TikTok's controls.
- `#fightchallenge`: it consists of publishing videos with fights organized by the authors themselves. It is considered dangerous because it can lead to the injury of the author or other participants.
- `#sugarbaby`: it consists of videos regarding "sugar babies", i.e., young people having sex with older ones for economic reasons only. It is considered dangerous because it deals with topics "inappropriate" for young people aged 0-18.
- `#updownchallenge`: it consists of moving intimate parts of the bodies to the rhythm of a song. It is considered dangerous because it deals with issues "inappropriate" for young people aged 0-18.

We point out that challenges much more dangerous than the seven ones selected by us were spread on TikTok in the past, such as those mentioned in Section **??**. They were promptly blocked by TikTok and, therefore, the recovery of the corresponding data was impossible.

Regarding the choice to consider seven non-dangerous and seven dangerous challenges, some discussions are in order. Indeed, the classification problem we are dealing with is a typical "rare class problem" [100]. It arises when there is a strong imbalance of the two classes to predict, and the class of greatest interest (which we call "positive") is precisely the rare one. In this scenario, a false negative (which, in our case, would imply classifying a dangerous challenge as non-dangerous) is much more serious than a false positive. Paradoxically, in a case like this, the most accurate classification model might be the one that simply classifies all classes as non-dangerous. However, such a model would be useless. In our context, it is better to have a model that is less accurate but is able to detect as many dangerous challenges as possible, even if it were to misclassify some non-dangerous challenges along the way [100]. It is precisely this reasoning that led us to use the same number of dangerous and non-dangerous challenges in the sample.

In practice, it is very difficult to find data on dangerous challenges because they are rare and are removed from TikTok as soon as they are recognized as dangerous.

[1] Note that the judgement of appropriateness refers not only to viewing the content but also to emulating it, since we are investigating TikTok challenges.

For this reason, in order to have a balanced dataset, we had to undersample the non-dangerous challenges. As pointed out above, this way of proceeding can lead to a worsening of the overall accuracy of our approach, but allows us to obtain very high values of sensitivity (i.e., recall). The latter allows our approach to correctly classify the maximum possible number of dangerous challenges.

Finally, we observe that, in any case, the number of challenges considered in the dataset is low. This is due in part to the rarity of the dangerous challenges and in part to the way of proceeding typical of the analyses on TikTok. In fact, these analyses often take into consideration few challenges, each characterized by many videos. For example, [467] analyzes 12 challenges, [28] examines 8 challenges, [101] considers 8 challenges and a total of 100 videos, [240] studies only one challenge characterized by 1,495 videos; finally, [553] and [511] each analyze two challenges. As we will see below, our 14 challenges still led us to examine 6,005 videos, which represent a significant number in the TikTok analyses scenario.

After the choice of the challenges, we developed a crawler capable of obtaining public data about the videos associated with a given challenge identified by its hashtag. Our crawler was written in Python and uses several libraries of this language, such as Pandas. The DBMS used to store the corresponding data is MongoDB. Our crawler is primarily a web scraper that, given in input the hashtag of a challenge, returns the list of all videos related to that hashtag. For each video so identified, it gets the list of its likes. For each like, it determines *(i)* the user who put it; *(ii)* whether this user has her privacy policy set to "public" or not; *(iii)* a video (if it exists) about the same challenge published by her. All these data are handled by means of a Pandas dataframe.

The choice to implement a web crawler using a web scraper is motivated by the fact that TikTok does not provide an API to fetch its data. On the other hand, the need of creating a web scraper due to the lack of an API means that out crawler does not suffer from time or rate limitations set by TikTok.

The data downloaded by our crawler are those publicly visible in TikTok. In other words, they are the same that any user would see when opening this app. In fact, our crawler can operate only with users who have set their privacy policy to "public" and comply with the Terms and Conditions of TikTok. Thanks to this and to the fact that it does not take any data from users who have their privacy policy set to "private", we can say that the use of our crawler does not pose ethical problems.

Our crawler suffers from some technical limitations due to its nature of a web scraper. In fact, the time to download the data for an experimental campaign is very large. The number of videos available for a challenge could be very high and the web scraper has to download and process the data of each video and its correspond-

ing author. Moreover, for each video, it has to find the data and the videos of all the users who liked it to check if they, in turn, published a video in the same challenge. Clearly, this is time consuming. For example, it took more than one week to download the data we used for the training activities of our experimental campaign (which involved 14 challenges). Instead, to download the data for the testing activities (involving 175 challenges) we took about two months.

We had to perform some pre-processing and cleaning activities on our data. First of all, we had to immediately verify the privacy settings of the user whose data we wanted to download. If that privacy setting was set to “private" we had to discard that user. This happened for about 30% of the users considered. For the remaining ones we carried out the classic ETL operations on their data. In particular, we removed all rows with null fields or inconsistencies. Next, we performed aggregations of numeric values. In particular, we had to transform the likes given to a certain video from a list of nicknames to an overall value. More generally, wherever possible, we had to convert lists and non-numeric values to numeric ones, because they are easier to process and much more suitable for data analyses.

After downloading data through our crawler, and after performing some pre-processing tasks, we obtained a record for each video. This record contains the following fields:

- `challenge_id`: the hashtag of the challenge which the video belongs to;
- `createTime`: the publication date of the video;
- `video_id`: the identifier of the video;
- `video_duration`: the video duration, expressed in seconds;
- `author_id`: the identifier of the author of the video;
- `author_verified`: it indicates whether the user is verified[2];
- `music_id`: the identifier of the music track or sound used in the video;
- `music_title`: the title of the music track or sound used in the video;
- `stats_diggCount`: the number of likes obtained by the video;
- `stats_playCount`: the number of views of the video;
- `authorStats_diggCount`: the total number of likes expressed by the author of the video for other videos;
- `authorStats_followingCount`: the number of users followed by the author of the video;
- `authorStats_followerCount`: the number of users following the author of the video;

[2] In TikTok, a verified user denotes a notable person.

- `authorStats_heartCount`: the total number of likes received by the author of the video;
- `originalVideo`: it is set to 1 if the video began the challenge it belongs to; otherwise, it is set to 0.
- `likedBy_ids`: the list of identifiers of the users, who put a like to the video and have their privacy policy set to "public".

Table 11.1 displays the number of videos we collected for each challenge, along with the date of the first and last one.

Challenge	*Number of Videos*	*Date of the first video*	*Date of the last video*
Non-dangerous Challenges			
#bussitchallenge	803	2020-06-11	2021-03-28
#copinesdancechallenge	250	2020-12-10	2021-03-24
#emojichallenge	663	2018-09-25	2021-03-27
#colpiditesta	1086	2018-01-21	2021-04-08
#boredinthehouse	359	2019-11-12	2021-04-06
#itookanap	206	2018-09-16	2021-03-22
#plankchallenge	380	2018-06-22	2021-04-08
Dangerous Challenge			
#silhouttechallenge	266	2018-08-15	2021-03-24
#bugsbunny	252	2018-01-05	2021-04-09
#strippatok	756	2019-02-16	2021-04-19
#firewroks	118	2018-02-03	2021-04-14
#fightchallenge	381	2018-08-08	2021-04-20
#sugarbaby	174	2018-09-11	2021-04-22
#updownchallenge	311	2018-06-17	2021-04-25

Table 11.1: Number of videos, date of the first and last one for each challenge

It is worth pointing out that, in the period in which we conducted our experimental campaign (June 2021 - August 2021), the lifespan of all the challenges we considered in the dataset could be considered concluded. In fact, these challenges, while continuing to exist, no longer generated significant interactions with users.

Finally, a consideration about the completeness of the dataset is due. In fact, as we said before, TikTok does not make available in an official way the data of the videos published. Since our data were not officially provided by TikTok, we cannot guarantee the completeness of our dataset. However, we can guarantee that, for each challenge, our crawler extracted all the information about its videos that were detectable on TikTok.

11.1.2 A Social Network-based model representing TikTok challenges

The second step of our research activity consists in the construction of a social network for each challenge. Specifically, let $\mathcal{C}$ be the set of challenges considered in the

dataset and let $\mathcal{C}'$ (resp., $\mathcal{C}''$) be the set of non-dangerous (resp., dangerous) challenges. Let C_i be a challenge of $\mathcal{C}$; a social network $\mathcal{N}_i = \langle N_i, A_i \rangle$ can be associated with it.

N_i is the set of nodes of $\mathcal{N}_i$. There is a node n_{i_j} for each author a_{i_j} who posted at least one video for C_i. A label l_{i_j} can be associated with n_{i_j}; it indicates the publication timestamp of the first video on C_i posted by a_{i_j}[3]. Since there is a biunivocal correspondence between a node $n_{i_j} \in N_i$ and the corresponding author a_{i_j}, in the following we will use these two terms interchangeably.

A_i is the set of arcs of $\mathcal{N}_i$. An arc (n_{i_j}, n_{i_k}) indicates that the author a_{i_k} put a like to a video posted by the author a_{i_j} and that the timestamp corresponding to l_{i_j} precedes the one corresponding to l_{i_k}. Intuitively, the presence of this arc indicates a form of propagation of the challenge C_i towards new users. In fact, it denotes that a_{i_j} published a video for C_i, a_{i_k} liked it and decided to publish her own video, thus participating to C_i.

To give an idea of the structure of the networks thus obtained, in Figure 12.1 (resp., Figure 12.2) we report the structure of the non-dangerous (resp., dangerous) networks. The more internal a node, the older the corresponding label and the most senior the associated author in the community of C_i.

In both figures there are nodes of different colors. In particular, we can find red, black and yellow nodes. The red node, if present, represents the author of the original video of the challenge, i.e., the author who started it. The yellow nodes represent the leaf nodes of the network, i.e., authors who have been stimulated to publish a video but have not been able to stimulate other authors to do so. Black nodes are all the other nodes in the network; they represent authors who were stimulated to publish a video and in turn were able to stimulate other authors to do so.

11.1.3 Analysis of the structure of the social networks associated with the challenges

In this section, we begin by analyzing the structure of the networks associated with the non-dangerous and dangerous challenges of our dataset to verify if there are structural differences between the networks corresponding to the two types of challenges. Tables 12.5 and 12.6 show the basic structural characteristics of the two types of networks. From the analysis of these tables, we can draw the following conclusions: *(i)* the networks associated with non-dangerous challenges are on average larger than those associated with dangerous challenges; *(ii)* there is no significant difference for the average degree and the clustering coefficient of the two types of

[3] Observe that a_{i_j} may post more videos on C_i over time.

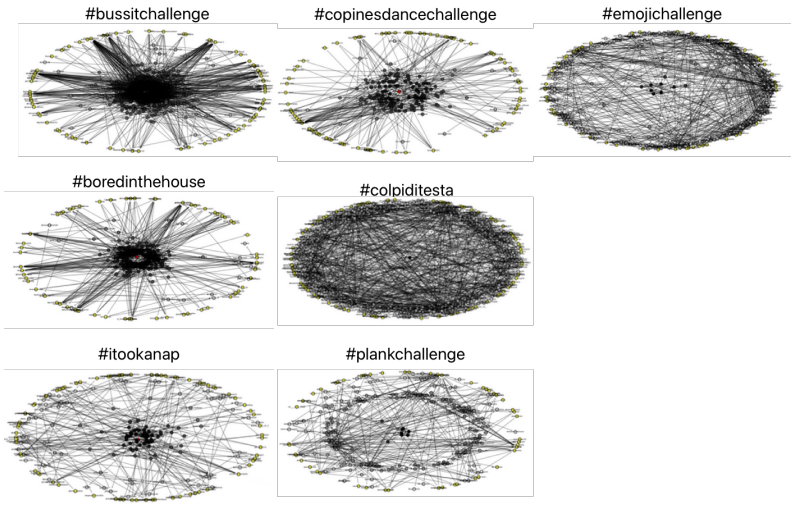


Fig. 11.1: Structure of non-dangerous networks

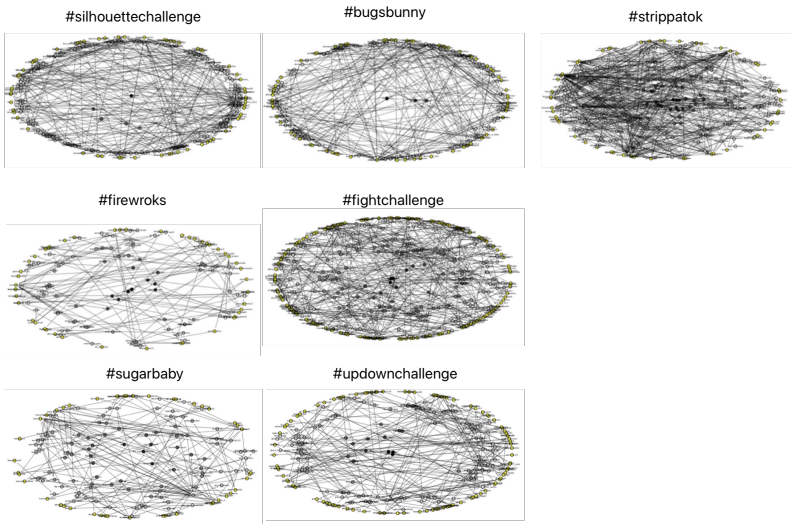


Fig. 11.2: Structure of dangerous networks

networks; *(iii)* the networks associated with dangerous challenges have a density higher than the ones associated with non-dangerous challenges.

In the next analysis, we focused on the characteristics of the videos for the two types of challenges. The main basic characteristics are shown in Table 12.7. From the analysis of this table we can observe that: *(i)* the average duration of the videos is similar in the two types of challenges; *(ii)* the average number of music tracks is higher in the non-dangerous challenges than in the dangerous ones; *(iii)* the average number of likes, comments, shares and views is higher for the dangerous challenges than for the non-dangerous ones.

After examining videos, we focused on the main basic characteristics of their authors. These characteristics are reported in Table 12.8. From the analysis of this table we can observe that: *(i)* there is a slight difference in the average number of follow-

Challenge	*Number of nodes*	*Number of arcs*	*Average degree*	*Average clustering coefficient*	*Density*
#bussitchallenge	618	708	1.14	0.0047	0.0019
#copinesdancechallenge	237	226	0.96	0	0.0040
#emojichallenge	440	498	1.13	0.0053	0.0026
#colpiditesta	691	843	1.22	0.0015	0.0018
#boredinthehouse	306	309	1.01	0.0018	0.0033
#itookanap	219	201	0.92	0	0.0042
#plankchallenge	271	266	0.98	0.0079	0.0036
Average Value	397.429	435.857	1.051	0.0030	0.0031

Table 11.2: Basic structural characteristics of the networks associated with non-dangerous challenges

Challenge	*Number of nodes*	*Number of arcs*	*Average degree*	*Average clustering coefficient*	*Density*
#silhouettechallenge	262	259	0.98	0	0.0037
#bugsbunny	212	239	1.13	0	0.0053
#strippatok	297	519	1.74	0.0025	0.0059
#firewroks	141	111	0.79	0.0083	0.0056
#fightchallenge	409	339	0.83	0.0009	0.0020
#sugarbaby	151	143	0.94	0.0035	0.0061
#updownchallenge	243	199	0.81	0.010	0.0033
Average Value	245	258.429	1.031	0.0036	0.0046

Table 11.3: Basic structural characteristics of the networks associated with dangerous challenges

Parameter	*Non-dangerous challenges*	*Dangerous challenges*
Average video duration (seconds)	21.39	20.38
Average number of music tracks used in a challenge	208.44	126.20
Average number of likes	178,104.13	249,152.12
Average number of comments	1,970.03	2,559.98
Average number of shares	5,456.83	6,990.26
Average number of views	1,471,020.16	2,070,632.01

Table 11.4: Differences between the main basic characteristics of the videos for non-dangerous and dangerous challenges

ers for the two types of authors; *(ii)* the authors of non-dangerous challenges tend to put more likes, follow many more authors and have many more videos published than the authors of dangerous challenges; *(iii)* the authors of dangerous challenges receive many more likes than the authors of non-dangerous ones.

Parameter	*Non-dangerous challenges*	*Dangerous challenges*
Average number of likes put by an author	17,730.52	11,998.711
Average number of likes received by an author	7,033,150.71	12,080,102.18
Average number of users followed by an author	1,357.08	670.24
Average number of users following an author	400,593.58	447,762.28
Average number of videos published	384.05	263.13

Table 11.5: Differences between the main basic characteristics of the authors of videos for non-dangerous and dangerous challenges

The last structural analysis we performed regarded the evolution of the network structure over time during the challenge lifespan. It is also a starting point for the next analyses that represent the core of our paper. In particular, this analysis focused on the average duration of the lifespan and the growth of the network size over time. The results obtained are reported in Table 12.9. From the analysis of this table we can observe important differences between non-dangerous and dangerous challenges. First of all, the average lifespan of dangerous challenges is longer than that of non-dangerous ones. Furthermore, the growth of non-dangerous challenges is much more gradual than that of dangerous ones. In fact, in the latter case, the growth is very limited up to about 75% of the lifespan, while it becomes "explosive" later. The investigation of the detailed differences concerning challenge lifespans represents the main topic of the research described in this paper.

11.1.4 Definition of the lifespan intervals of a challenge

In the last experiment of the previous section we have seen that the growth of non-dangerous networks seems to show a totally different trend from the one characterizing dangerous networks. In this section, we explore this aspect more deeply.

As a first step, we considered the variation of the size of each network during its lifespan. Clearly, the functions thus obtained would be broken lines, whatever the sampling frequency. Actually, we chose a very high sampling frequency, equal to 1% of the lifespan. However, for motivations we will see later, we wanted to have continuous curves, rather than broken lines. For this reason, we interpolated the points using a univariate spline. Given the high sampling frequency we chose, we assumed

Parameter	*Non-dangerous challenges*	*Dangerous challenges*
Average challenge lifespan (days)	405	550.17
Average number of network nodes at 5% of lifespan	8.6	2.2
Average number of network nodes at 25% of lifespan	140.4	7.6
Average number of network nodes at 50% of lifespan	172	22.4
Average number of network nodes at 75% of lifespan	179.4	58.8
Average number of current network nodes (100% of lifespan)	397.43	245.67

Table 11.6: Differences between the main basic characteristics of the lifespan for non-dangerous and dangerous challenges

that the difference between the broken line and the curve obtained by interpolation was minimal. To test this hypothesis, we computed the Mean Absolute Error (i.e., MAE) between them, considering 100 additional equidistant points for each interval (thus considering 10,000 points for each lifespan). Afterwards, for each point, we normalized this value against the corresponding one of the broken line. The obtained results are reported in Table 12.10. From the analysis of this table we can observe that the average normalized differences are very low. Therefore, the interpolation we made can be considered acceptable.

Non-dangerous Challenge	*Normalized MAE*	*Dangerous Challenge*	*Normalized MAE*
#bussitchallenge	0.013	#silhouttechallenge	0.018
#copinesdancechallenge	0.014	#bugsbunny	0.016
#emojichallenge	0.022	#strippatok	0.024
#colpiditesta	0.024	#firewroks	0.025
#boredinthehouse	0.012	#fightchallenge	0.015
#itookanap	0.016	#sugarbaby	0.022
#plankchallenge	0.017	#updownchallenge	0.025

Table 11.7: Normalized MAE between the continuous function returned by the univariate spline interpolation and the real values for non-dangerous challenges (at left) and dangerous ones (at right)

The reason we wanted to have a continuous curve is that it allows the computation of the first derivative and, then, the identification of the points of the lifespan where the curve slope inverts.

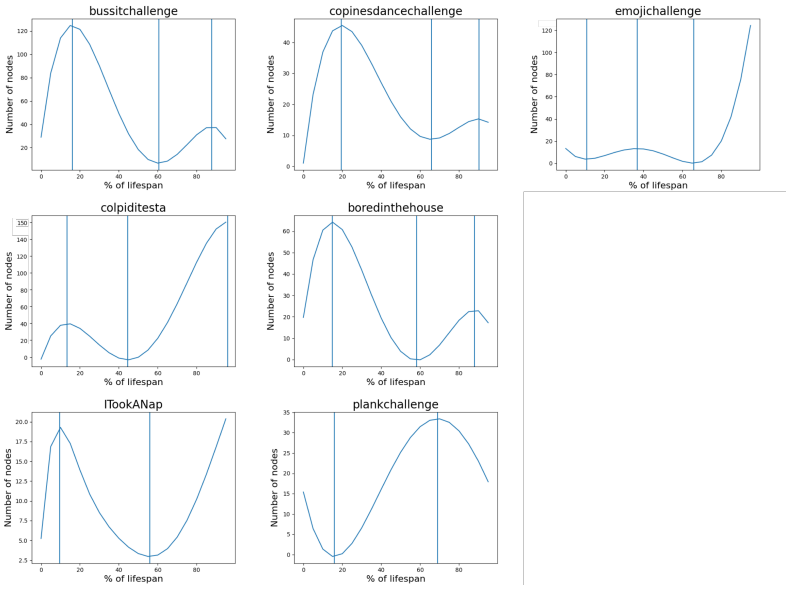


Fig. 11.3: Trend of the function $v_i(\cdot)$ and corresponding intervals for non-dangerous challenges

Let C_i be a challenge, let $\mathcal{N}_i$ be the corresponding network and let $v_i(\cdot)$ be the function representing the variation of the number of nodes of $\mathcal{N}_i$ during the lifespan of C_i. $v_i(\cdot)$ is obtained by applying the univariate spline on the data of C_i. Let $X = \{x_1, x_2, \cdots, x_N\}$ be the set of points for which the first derivative of $v_i(\cdot)$ is null. The lifespan of C_i can be divided into $N-1$ intervals (x_q, x_{q+1}), $1 \leq q \leq N-1$, such that $v_i(\cdot)$ is always increasing or always decreasing within each interval. As we will see later, such intervals play a key role in our approach.

In Figure 12.3 (resp., 12.4) we show the trend of the function $v_i(\cdot)$ and the corresponding intervals for non-dangerous (resp., dangerous) challenges. Already from the examination of these figures we can see how the two types of challenges show very different trends of $v_i(\cdot)$. Capturing such differences is the next goal of this paper.

Definition of features to characterize lifespan intervals. As the next step of our approach, we determined a set of features capable of characterizing an interval of a challenge lifespan. To this end, we tried to maximize the number of features to consider taking all those available from the dataset plus several others derived from Social Network Analysis. The latter were possible thanks to the Social Network-based model for the representation of a challenge described in Section 11.1.2. Proceeding in this way, given a challenge C_i, the corresponding social network $\mathcal{N}_i$, and an interval $\mathcal{I}$, we identified the following 26 features characterizing it:

- `video_number`: number of videos of C_i posted during $\mathcal{I}$;

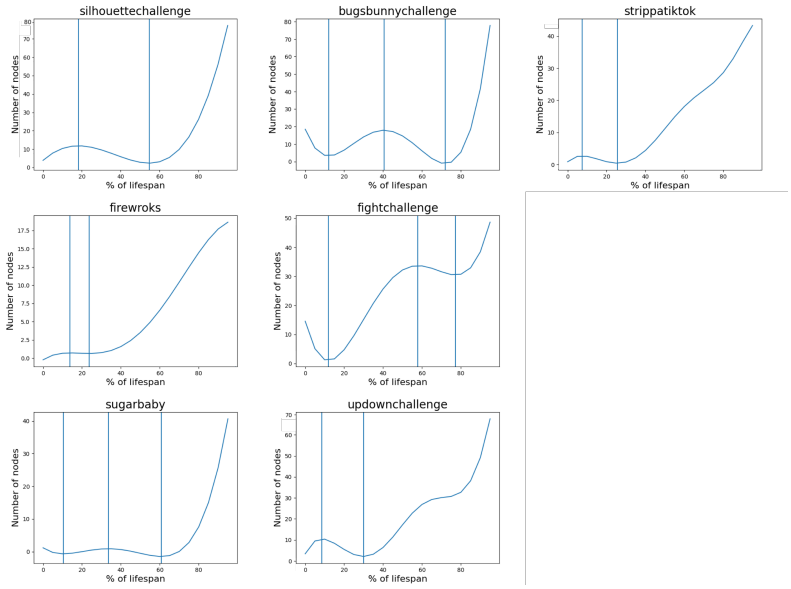


Fig. 11.4: Trend of the function $\nu_i(\cdot)$ and corresponding intervals for dangerous challenges

- `video_difference`: difference between the number of videos posted during $\mathcal{I}$ and the number of videos posted in the previous interval;
- `begin_percentage`: percentage of the lifespan at which $\mathcal{I}$ begins;
- `end_percentage`: percentage of the lifespan at which $\mathcal{I}$ ends;
- `duration`: duration of $\mathcal{I}$ (expressed in days);
- `average_hours_between`: average number of hours elapsed between the posting of two videos during $\mathcal{I}$;
- `likes`: total number of likes obtained by C_i during $\mathcal{I}$;
- `average_likes`: average number of likes obtained by C_i during $\mathcal{I}$;
- `average_comments`: average number of comments obtained by C_i during $\mathcal{I}$;
- `average_shares`: average number of shares obtained by C_i during $\mathcal{I}$;
- `average_views`: average number of views obtained by C_i during $\mathcal{I}$;
- `average_followers`: average number of followers of the authors of the videos posted during $\mathcal{I}$;
- `average_following`: average number of users followed by the authors of the videos posted during $\mathcal{I}$;
- `average_likes_authors`: average number of likes received by the authors of the videos posted during $\mathcal{I}$;
- `verified_authors`: number of verified authors (see Section 11.1.1) posting videos during $\mathcal{I}$;
- `number_nodes`: number of nodes of $\mathcal{N}_i$;
- `number_arcs`: number of arcs of $\mathcal{N}_i$;

- network_density: density of $\mathcal{N}_i$;
- connected_components: number of connected components of $\mathcal{N}_i$;
- maximum_size_components: number of nodes of the maximum connected component of $\mathcal{N}_i$;
- average_degree_centrality: average degree centrality of the nodes of $\mathcal{N}_i$;
- average_eigenvector_centrality: average eigenvector centrality of the nodes of $\mathcal{N}_i$;
- average_pagerank: average PageRank of the nodes of $\mathcal{N}_i$;
- average_closeness_centrality: average closeness centrality of the nodes of $\mathcal{N}_i$;
- average_betweenness_centrality: average betweenness centrality of the nodes of $\mathcal{N}_i$;
- average_clustering_coefficient: average clustering coefficient of the nodes of $\mathcal{N}_i$.

However, such a large number of features is difficult to manage. Therefore, we decided to carry out a study of their correlations to see if some of them could be filtered out. In Figure 12.5, we show the correlation matrix thus obtained. This figure shows several valuable information that can help us to better understand the mutual inter-relationships between the features, as well as the inter-relationships between the features and the structure of the underlying network.

In particular, some interesting information that can be derived and that help us to select a manageable number of features to characterize lifespans are the following:

- There is a high direct correlation between video_number, video_difference, number_nodes, number_edges, maximum_size_component and average_degree_centrality. Therefore, to characterize lifespans, it is sufficient to keep only one of them and discard the others. We decided to keep video_number.
- There is a high direct correlation between login_percentage and end_percentage. For this reason, we decided to keep begin_percentage and discard end_percentage.
- There is a low correlation between duration and all the other features. Therefore, we decided to keep this feature. A similar reasoning applies to average_hours_between, average_following and average_betweenness_centrality.
- There is a high direct correlation between like, average_likes, average_comments, average_shares, average_views and verified_authors. For this reason, it is sufficient to keep only one of them and discard all the others. We decided to keep average_likes.

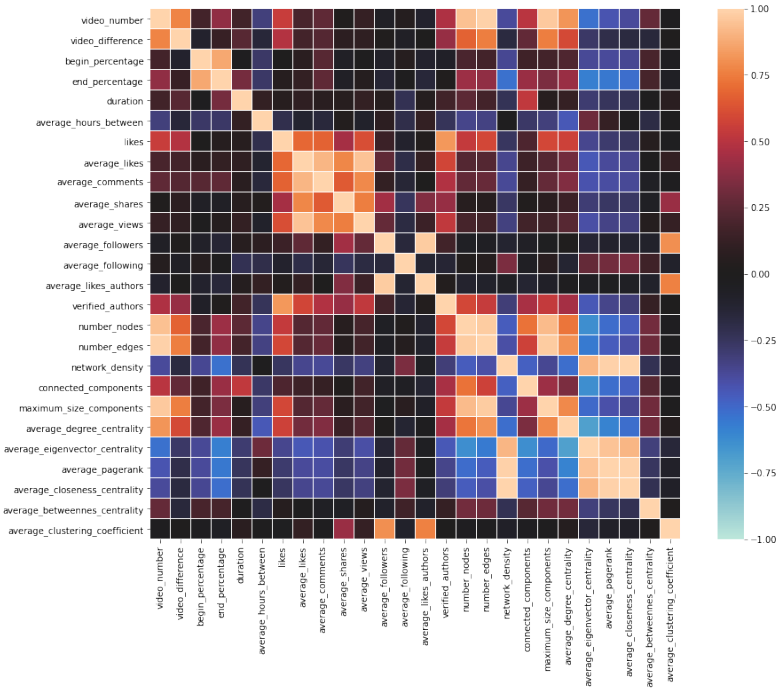


Fig. 11.5: Correlation matrix for the 26 features selected for characterizing lifespan intervals

- The features `average_followers` and `average_like_authors` have a high direct correlation with each other. Furthermore, each of them has also a high direct correlation with `average_clustering_coefficient`. Therefore, we decided to keep the latter feature and discard the first two.
- The feature `connected_component` has both direct and inverse medium-high correlations with many features. Therefore, we decided to discard it. A similar reasoning applies to `maximum_size_component` and `average_degree_centrality`.
- The features `average_eigenvector_centrality`, and `average_closeness_centrality` have a high direct correlation with each other and with `network_density`. As a consequence, one might think of keeping only one of these features and discarding the others. However, we observe that all of them have a high inverse correlation with several other ones, for example with `video_number`, which we have already kept, and `end_percentage`. In turn, the latter has a very high correlation with `begin_percentage`, which we have already kept. For this reason, we decided to discard all these features.

Summarizing, at the conclusion of this examination, we decided to select the following eight features for characterizing lifespan intervals:

- `average_likes`;

- average_following;
- video_number;
- duration;
- average_betweenness_centrality;
- average_clustering_coefficient;
- average_hours_between;
- begin_percentage.

Characterizing the intervals of challenge lifespans. In the previous sections, we determined, through the function $\nu(\cdot)$, the lifespan of the 14 challenges of our interest. Afterwards, through the computation of the first derivative of $\nu(\cdot)$, we divided each lifespan into intervals. In this section, we illustrate our approach for characterizing these intervals. Roughly speaking, it consists of grouping them into homogeneous clusters, based on the eight features identified above, and, then, determining the characteristics of each cluster.

As a first task of this activity, we considered a new dataset consisting of one table whose rows were associated with intervals and whose columns corresponded to the eight features. Each row of the table reported the values of the eight features for the corresponding interval.

Afterwards, we applied the Principal Component Analysis (hereafter, PCA) [294] to this dataset and reduced the number of dimensions from 8 to 2. This allowed us to represent the intervals in a plane, in order to favor a visual representation of the clusters obtained.

After this task, we applied Autoclass [143], a classical algorithm that uses Naive-Bayes in combination with Expectation-Maximization to find the probability distribution parameters best fitting the data. We chose Autoclass because, among the various strengths characterizing it, there is also the capability of automatically determining the number of clusters [294]. In fact, it was not possible to make any preliminary conjecture about this number, and the elbow method performed with k-means returned no results. Autoclass allowed us to group the intervals into five clusters. Thanks to the preliminary application of PCA, these clusters can be represented in a plane whose coordinates correspond to the two dimensions returned by PCA. The five clusters thus obtained are shown in Figure 12.6.

From the analysis of this figure we can observe that these clusters actually appear quite homogeneous. However, in order for them to be useful for our analysis, it is necessary to understand what type of intervals each cluster represents. By carefully examining the features of the intervals belonging to each cluster, we were able to draw the following characterizations:

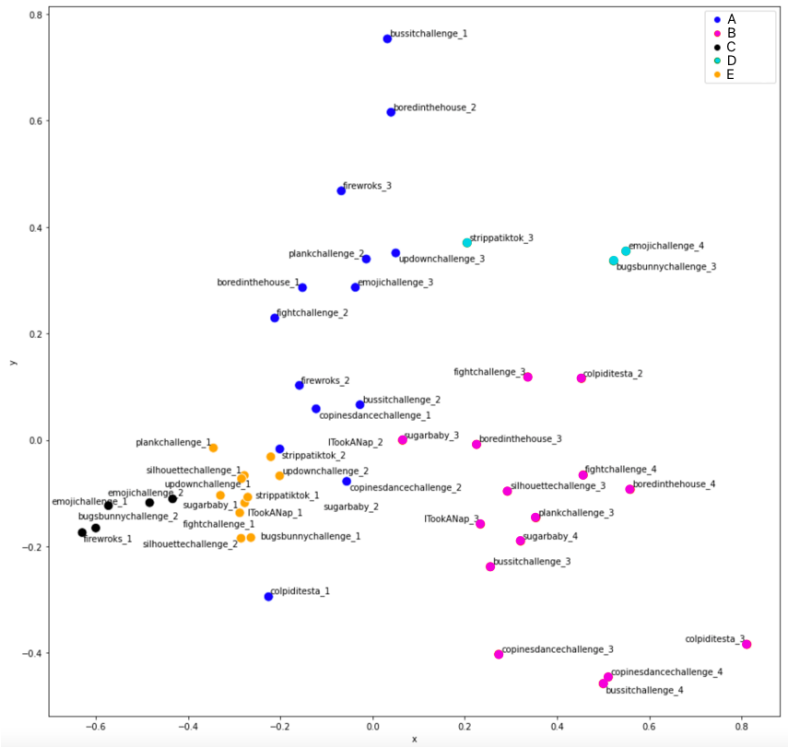


Fig. 11.6: The five clusters of intervals returned by Autoclass

- *Cluster A*: it includes final intervals of lifespans, when the challenge has less attraction on users. When compared to the other intervals of the same challenge, the ones of Cluster A are characterized by: *(i)* a lower average number of likes; *(ii)* the presence of less important authors (in fact, verified authors are few and most of them have few followers). Finally, the time interval between the publication of two consecutive videos is longer. The networks associated with these intervals are more connected and have higher centrality values than the ones corresponding to other intervals. This represents a further evidence that we are in a well-established phase of the challenge.
- *Cluster B*: it includes intervals belonging to a peak phase of the challenge. In fact, they are characterized by a very high number of likes and videos published. There are many verified authors, as well as many authors with many followers. The time interval between the publication of two consecutive videos is short.
- *Cluster C*: it includes initial intervals of lifespans. The number of likes is less than that characterizing the intervals of Cluster B. However, it is quite high, and this means that the challenge is arousing curiosity and will probably have a peak in a later interval. The users are generally not verified but have a high number of followers. This makes the number of views and shares very high. The time interval between the publication of two consecutive videos is quite long. The networks associated with these intervals are poorly connected. This indicates

that, in these intervals, video postings are made by people still unconnected to each other. This represents a further evidence that we are in an initial phase of the challenge.

- *Cluster D*: it includes lifespan intervals that follow a challenge peak. Intervals belonging to this cluster are characterized by a low number of likes. Most of the authors of the videos are verified and have many followers and interactions. The average number of videos posted is high. The time elapsed between the posting of two consecutive videos is short, although it tends to increase as we move towards the end of the intervals. The network associated with these intervals is fairly connected.
- *Cluster E*: it includes initial intervals of lifespans. They are characterized by a high number of likes and published videos. There are many views but few comments and few shares. This implies that the interaction level between users is low. The time elapsed between the publication of two consecutive videos is long. The network associated with intervals is quite disconnected.

To give also a quantitative idea of the characteristics of the clusters, in Table 11.8 we report the average values assumed in each cluster by the eight features that we selected to represent the lifespan intervals.

Features	*Cluster A*	*Cluster B*	*Cluster C*	*Cluster D*	*Cluster E*
average_likes	49,235	253,521	164,872	55,964	172,454
average_following	778	894	1,074	795	1,089
video_number	722	891	128	742	105
duration	235	174	14	224	28
average_betweenness_centrality	0.74	0.54	0.12	0.68	0.05
average_clustering_coefficient	0.0282	0.0443	0.0011	0.0323	0.0008
average_hours_between	42	0.51	153	34	122
begin_percentage	90%	46%	0%	88%	5%

Table 11.8: Average values assumed in each cluster by the features representing lifespan intervals

11.2 Results

11.2.1 Searching for time patterns in the challenge lifespans

After grouping the intervals into homogeneous clusters, we were able to perform the second main investigation of this paper, namely the extraction of time patterns allowing us to distinguish non-dangerous challenges from dangerous ones.

As a first step, we considered the lifespan of the 14 challenges under examination and verified to which cluster the corresponding intervals belonged. If two consecutive intervals belonged to the same cluster we considered them as if they were a single one. At the end of this activity, we obtained the following sequences of intervals for non-dangerous challenges:

- #boredinthehouse: B, A
- #bussitchallenge: B, A
- #colpiditesta: B, A
- #copinesdancechallenge: B, A
- #emojichallenge: C, B, D
- #ITookANap: E, B, A
- #plankchallenge: E, B, A

Instead, the sequences of intervals characterizing the dangerous challenges were as follows:

- #bugsbunnychallenge: E, C, D
- #fightchallenge: C, B, A
- #firewroks: C, B
- #silhouettechallenge: E, A
- #strippatok: E, D
- #sugarbaby: E, A
- #updownchallenge: E, B

From the examination of the previous sequences, we drew some interesting information. In particular, we observed that:

- In non-dangerous challenges, the pattern B, A tends to repeat often. In any case, an interval belonging to the cluster B is always present. However, it is always followed by an interval belonging to the clusters A or D.
- In dangerous challenges there is no dominant pattern. However, the presence of an interval belonging to the cluster E is often observed.

We noticed that the intervals of type D generally followed the peak of a challenge and that the ones of type A generally were the final ones of a challenge. By analyzing the data for these intervals in a detailed and comparative way, we observed that:

- The intervals of type A were characterized by few interactions (i.e., views, likes, comments and shares) with videos. The number of videos posted during them was high, albeit the number of likes received by them was small. Their duration was long and the associated networks were very dense.

- The intervals of type D were also characterized by few interactions with videos. The number of videos posted is high, while the number of likes received by these videos and the duration of the intervals were low. The associated networks were very dense.

These characteristics led us to hypothesize that the intervals of types A and D represented the same reality, i.e., the conclusion of a challenge. More precisely, they represented two slightly different ways of challenge conclusion. In fact, the intervals of type A described a faster conclusion, while those of type D represented a slower one.

To deepen this hypothesis we decided to perform a t-test based on the following null hypothesis H0: "The means of the samples for the intervals of types A and D are equal". The metrics we used to perform this test are the eight features we selected to characterize the intervals of the challenge lifespans, namely: `average_likes`, `average_following`, `video_number`, `duration`, `average_betweenness_centrality`, `average_clustering_coefficient`, `average_hours_between`, `begin_percentage` (see Section 11.1.4 for all details).

Actually, in order to apply the classical t-test it is necessary that the elements of the two samples have equal variance; otherwise, it is necessary to use the Welch's t-test [100].

In order to decide what kind of t-test was appropriate, we applied the Bartlett's t-test [60] to the intervals of types A and D; also for this test we applied the same metrics used for t-test. The Bartlett's t-test is used to know if two samples with different numbers of elements have the same variance or not. In our application of it, we considered the following null hypothesis H0: "The variances of the samples for the intervals of types A and D are equal". At this point, we computed the corresponding p-value and obtained that it is equal to 0.003. Since this value is smaller than 0.05, we concluded that the null hypothesis was rejected and, therefore, it was necessary to apply the Welch's t-test, instead of the classical one, to test the hypothesis H0: "The means of the samples for the intervals of types A and D are equal". Applying this test, we obtained a p-value of 0.67, which was much greater than 0.05. Therefore, the null hypothesis cannot be rejected.

As a consequence, deepening through t-test did not invalidate our hypothesis that the intervals of type A and D represent the conclusion of a challenge. Despite their minor differences, for the purpose of our research, we can assume that A and D are equivalent.

Based on this assumption, the sequences of intervals for non-dangerous challenges were the following:

- #boredinthehouse: B, A
- #bussitchallenge: B, A
- #colpiditesta: B, A
- #copinesdancechallenge: B, A
- #emojichallenge: C, B, A
- #ITookANap: E, B, A
- #plankchallenge: E, B, A

Instead, the sequences of intervals for dangerous challenges were the following:

- #bugsbunnychallenge: E, C, A
- #fightchallenge: C, B, A
- #firewroks: C, B
- #silhouettechallenge: E, A
- #strippatok: E, A
- #sugarbaby: E, A
- #updownchallenge: E, B

After this, we considered the intervals of types C and E. The description given above allowed us to hypothesize that both of them were initial lifespan intervals. Also, the number of likes and the number of videos posted during them were comparable. The properties of the networks associated with them were also similar. Analogously to what we performed for A and D, we carried out a statistical analysis to deepen our hypothesis. In this case, the Bartlett's t-test with the null hypothesis H0: "the variances of the samples for the intervals of types C and E are equal", and with the same metrics used for the previous t-test, gave us a value of 0.55, which is much greater than 0.05. Therefore, we could conclude that the null hypothesis cannot be rejected. Consequently, we could apply the classical t-test with the following null hypothesis H0: "The means of the samples for the intervals of types C and E are equal" and with the metrics used for all the previous t-tests. In this case, the computation of the p-value returned 0.91. Therefore, the null hypothesis cannot be rejected.

As a consequence, also for the intervals of type C and E, the further investigation through t-test did not invalidate our hypothesis, namely that both intervals represent the beginning of a challenge, albeit with some minor specificities. Despite them, for the purposes of our research, we can assume that C and E are equivalent.

Based on this assumption, the interval sequences for non-dangerous challenges were the following:

- #boredinthehouse: B, A
- #bussitchallenge: B, A

- #colpiditesta: B, A
- #copinesdancechallenge: B, A
- #emojichallenge: C, B, A
- #ITookANap: C, B, A
- #plankchallenge: C, B, A

Instead, the interval sequences for dangerous challenges were the following:

- #bugsbunnychallenge: C, A
- #fightchallenge: C, B, A
- #firewroks: C, B
- #silhouettechallenge: C, A
- #strippatok: C, A
- #sugarbaby: C, A
- #updownchallenge: C, B

Thanks to this result, we were able to identify some time patterns characterizing non-dangerous and dangerous challenges. As we will see below, since these time patterns are different in the two cases, they are also able to differentiate one type of challenge from the other.

Let us first examine non-dangerous challenges. In this case, we always have the presence of a sequence of intervals of type B, A. This sequence is very often preceded by an interval of type C, so that we have a time pattern of type C, B, A. Recall that: *(i)* the intervals of type C are initial ones in a challenge lifespan; *(ii)* the intervals of type B correspond to a peak of a challenge; *(iii)* the intervals of type A indicate the end of a challenge. We argued that the typical time pattern of a non-dangerous sequence is C, B, A. In fact, the challenges showing a B, A time pattern already existed when our research on them began although the interactions with users that they were able to elicit were almost negligible.

Let us now examine dangerous challenges. In this case, unlike the previous one, there is no single sequence of intervals characterizing all of them. Instead, we identified three dominant sequences that correspond to three different "fates" generally characterizing the challenges of this type. In particular, the three time patterns are:

- C, B: these challenges had a standard initial phase with an interval of type C; then, they reached a peak phase. Finally, they almost suddenly ceased to have meaningful interactions with users. This may have happened because they ran out of steam very quickly or they were recognized by TikTok as dangerous and were stopped or removed from the social network.

- C, A: these challenges had an initial phase, which was followed by a decay one. In other words, they never reached the peak. They were born, survived for a certain period on the social network, and then died.
- C, B, A: as we will see below, these challenges are a small minority among the dangerous ones. They behaved like the non-dangerous ones, in that they were born, had a peak and, finally, decayed.

In order to verify the goodness of our approach, we decided to test it on a new dataset, larger than the previous one. It stores data on 175 challenges; 150 of them are non-dangerous while 25 are dangerous. Due to space limitations, we cannot detail these challenges as we did for the 14 challenges defined in Section 11.1.1. However, in Table 11.9, we report the aggregate values of some fields that refer to them and whose meaning we had illustrated in Section 11.1.1.

Parameter	*Non-dangerous challenges*	*Dangerous challenges*
Publication month of the first video	From 2018-01 to 2019-12	From 2017-01 to 2020-12
Publication month of the last video	From 2018-03 to 2021-02	From 2017-02 to 2021-04
Average lifespan in days	523.45	364.73
Average number of videos	542.54	366.55
Average number of likes received	184,234.52	247,325.48
Average number of comments received	1,984.05	2,654.03
Average number of shares	5,548.72	7,002.44
Average number of views	1,475,042.16	2,084,544.06

Table 11.9: Aggregate values of some fields that refer to non-dangerous and dangerous challenges

The results obtained are the following:

- As for non-dangerous challenges:
  - 134 (i.e., 89.33% of them) followed the time pattern C, B, A. This is the only one we identified as significant for this type of challenges.
  - 16 (i.e, 10.67% of them) followed several other sequences of intervals.
- As for dangerous challenges:
  - 10 (i.e., 40.00% of them) followed the time pattern C, B;
  - 11 (i.e., 44.00% of them) followed the time pattern C, A;
  - 2 (i.e., 8.00% of them) followed the time pattern C, B, A;
  - 2 (i.e., 8.00% of them) followed other sequences of intervals.

As a further analysis, having trained our model on a balanced dataset, we decided to create a third dataset of 300 challenges (150 non-dangerous and 150 dangerous

ones). The 150 non-dangerous challenges are those of the previous dataset. As for the dangerous challenges, since they are very rare, they have been obtained from the 25 challenges of the previous dataset using the oversampling technique implemented through bootstrap [100]. The results obtained by applying our approach to the new dataset are the following:

- As for non-dangerous challenges:
  - 132 (i.e., 88.00% of them) followed the time pattern C, B, A.
  - 18 (i.e,, 12.00% of them) followed a variety of other sequences of intervals; these were partially different from the ones found in the previous dataset, because they were influenced by the new composition of the dataset.
- As for dangerous challenges:
  - 65 (i.e., 43.33% of them) followed the time pattern C, B;
  - 69 (i.e., 46.00% of them) followed the time pattern C, A;
  - 7 (i.e., 4.67% of them) followed the time pattern C, B, A;
  - 9 (i.e., 6.00% of them) followed a variety of other sequences of intervals.

The results obtained from both these datasets represent a confirmation that the time patterns we detected actually exist for the two types of challenges into consideration and are capable of discriminating them. In addition, they show that the patterns we found are really able to capture almost all the behaviors of TikTok challenges.

Note that with both datasets the sensitivity of our approach is very high. In fact, it is equal to 92.00% in the case of the second dataset (i.e., the one containing only real challenges), while it raises to 94.00% in the case of the third dataset (i.e., the one balanced through the oversampling of dangerous challenges).

12

Investigating community evolutions in TikTok

In just few years, TikTok has become a major player in the social media environment, especially with regards to teenagers. One of the key factors of this success is the idea of challenges. Unfortunately there are users who launch challenges that are dangerous, or at least suitable only for an adult audience (and TikTok is the most popular social network for teenagers). This paper focuses primarily on this kind of challenge. In particular, it investigates an aspect not yet studied in the literature, that is the different characteristics and evolutionary dynamics of the communities of users participating in non-dangerous and dangerous challenges. The final goal is the identification of evolutionary patterns that distinguish the communities of users participating in the two types of challenges. In this way, it provides a new tool to identify dangerous challenges, which is very robust against the tricks generally used to bypass the current TikTok controls.

The material presented in this chapter was derived from [256].

12.1 Methods

12.1.1 Dataset construction

In order to perform our research, we needed a dataset recording a set of data and metadata related to non-dangerous and dangerous challenges in TikTok. To the best of our knowledge, there was no dataset with such characteristics already available and we decided to build it from scratch. In identifying the challenges to be considered in such a dataset, we focused on some of them that were very common in TikTok at the time of data extraction. Specifically, we considered seven non-dangerous challenges and seven dangerous ones. To this end, we assumed as dangerous a challenge that had received several criticisms in the media about the problems it could cause to the people participating in it. As it usually happens in TikTok, we identify each challenge through the hashtag used to post the corresponding videos. In Table 12.1, we report the seven non-dangerous challenges, while in Table 12.2 we show the seven dangerous ones. Actually, in the past, much more dangerous challenges than

those shown in Table 12.2 have been published on TikTok. Some of them, such as the Benadryl challenge and the Blackout challenge mentioned in the Introduction, have even caused the death of participants. These challenges, and other ones equally disrupting, were promptly blocked by TikTok and the access to the corresponding data was impossible.

Challenge	*Description*
#bussitchallenge	Participants show themselves changing clothes.
#copinesdancechallenge	Participants perform a series of dance movements.
#emojichallenge	Participants imitate different emoji.
#colpiditesta	Participants virtually hit a soccer ball with their heads.
#boredinthehouse	Participants film a subject, often an animal, in different parts of the house.
#itookanap	Participants film a subject, often an animal, sleeping.
#plankchallenge	Participants perform dance movements based on training excercises.

Table 12.1: The seven non-dangerous challenges of our dataset

After choosing the challenges, we developed a crawler to obtain public data about them and the corresponding videos. Our crawler anonymizes information about the authors of the videos. More specifically, for each challenge, it records the identifier of the video originating it and the identifiers of the other videos referring to it. For each of these videos, our crawler derives its list of likes. Finally, for each like, it determines: *(i)* the user who posted it; *(ii)* her privacy policy; *(iii)* any possible video that she posted in the same challenge[1].

After downloading the data for each video and performing some pre-processing tasks, we obtained a record for each video. It contains the fields shown in Table 12.3.

In Table 12.4, for each challenge, we report the number of videos registered in our dataset.

12.1.2 Model definition

After illustrating the dataset on which we perform our analyses, we want to define a model to represent a challenge. Specifically, our model is a social network-based ones. In particular, let $\mathcal{C}'$ (resp., $\mathcal{C}''$) be the set of non-dangerous (resp., dangerous) challenges and let $\mathcal{C}$ be the union of $\mathcal{C}'$ and $\mathcal{C}''$. Let C_i be a challenge of $\mathcal{C}$; a social network $\mathcal{N}_i = \langle N_i, A_i \rangle$ can be associated with it.

[1] In TikTok, a user can post more videos for the same challenge.

Challenge	*Description*
`#silhouttechallenge`	Participants expose their bodies covered by a red filter. Participants are often naked and the filter, being digital, can be easily removed.
`#bugsbunny`	Participants lie on their stomachs and lift their legs upward to show their feet sticking out of their heads like the ears of a rabbit. Then they begin to move their feet to the beat of a song. Participants often show intimate parts of their bodies.
`#strippatok`	Participants post videos related to strippers (both males and females). Clearly it regards topics not suitable for a young audience.
`#firewroks`	Participants post videos with fireworks risking their safety. The seemingly wrong hashtag is a trick by participants to bypass TikTok's controls.
`#fightchallenge`	Participants post videos with battles that they organize. It is judged dangerous because it can lead to fighters getting injured.
`#sugarbaby`	Participants post videos about "sugar babies", i.e., young people having sex with older people for money.
`#updownchallenge`	Participants move intimate parts of their bodies to the beat of a song.

Table 12.2: The seven dangerous challenges of our dataset

N_i is the set of nodes of $\mathcal{N}_i$. There is a node n_{i_j} for each author a_{i_j} who posted at least one video for C_i. Each node n_{i_j} has associated a label l_{i_j} that registers the publication timestamp of the first video that a_{i_j} posted for C_i[2]. Since there is a biunivocal correspondence between a node $n_{i_j} \in N_i$ and the corresponding author a_{i_j}, in the following we will use these two terms interchangeably.

A_i is the set of arcs of $\mathcal{N}_i$. An arc $(n_{i_j}, n_{i_k}) \in Ai$ denotes that the author a_{i_k} posted a like on a video published by a_{i_j} and that the timestamp recorded in l_{i_j} precedes the one recorded in l_{i_k}. Intuitively, the arc (n_{i_j}, n_{i_k}) denotes that the challenge C_i propagated from a_{i_j} to a_{i_k}. In fact, a_{i_j} posted a video for C_i; this was liked by a_{i_k}, who, in turn, posted a video of her own for the same challenge.

To give an idea of the variety of the obtained social networks (and, therefore, of the corresponding challenges), in Figure 12.1 (resp., 12.2), we show a representation of those associated with non-dangerous (resp., dangerous) challenges. In it, the more

[2] Note that a_{i_j} could post more videos for C_i over time.

Feature	*Description*
challenge_id	the hashtag of the challenge which the video belongs to;
createTime	the publication date of the video;
video_id	the identifier of the video;
video_duration	the video duration, expressed in seconds;
author_id	the identifier of the author of the video;
author_verified	it indicates whether the user is verified (in TikTok, a verified user denotes a notable person);
music_id	the identifier of the music track or sound used in the video;
music_title	the title of the music track or sound used in the video;
stats_diggCount	the number of likes obtained by the video;
stats_playCount	the number of views of the video;
authorStats_diggCount	the total number of likes expressed by the author of the video for other videos;
authorStats_followingCount	the number of users followed by the author of the video;
authorStats_followerCount	the number of users following the author of the video;
authorStats_heartCount	the total number of likes received by the author of the video;
originalVideo	it is set to 1 if the video began the challenge it belongs to; otherwise, it is set to 0;
likedBy_ids	the list of identifiers of the users, who put a like to the video and have their privacy policy set to "public" (our crawler can operate only with users adopting this policy; it cannot derive information from users having their privacy policy set to "private".).

Table 12.3: The record associated with each challenge video

Non-dangerous Challenge	*Number of Videos*	*Dangerous Challenge*	*Number of Videos*
#bussitchallenge	803	#silhouttechallenge	266
#copinesdancechallenge	250	#bugsbunny	252
#emojichallenge	663	#strippatok	756
#colpiditesta	1086	#firewroks	118
#boredinthehouse	359	#fightchallenge	381
#itookanap	206	#sugarbaby	174
#plankchallenge	380	#updownchallenge	311

Table 12.4: Number of videos we collected for non-dangerous challenges (at left) and dangerous ones (at right)

internal a node, the older its label and the more senior the associated user for the challenge.

12.2 Results

12.2.1 A preliminary analysis of challenges

In this section, we begin with a preliminary analysis of the networks associated with the challenges in our dataset. It serves a dual purpose, namely: *(i)* verifying if there are structural differences between the networks associated with the two types of challenges; *(ii)* identifying interesting insights to investigate whether the user communities related to the two types of challenges have different evolutions or not,

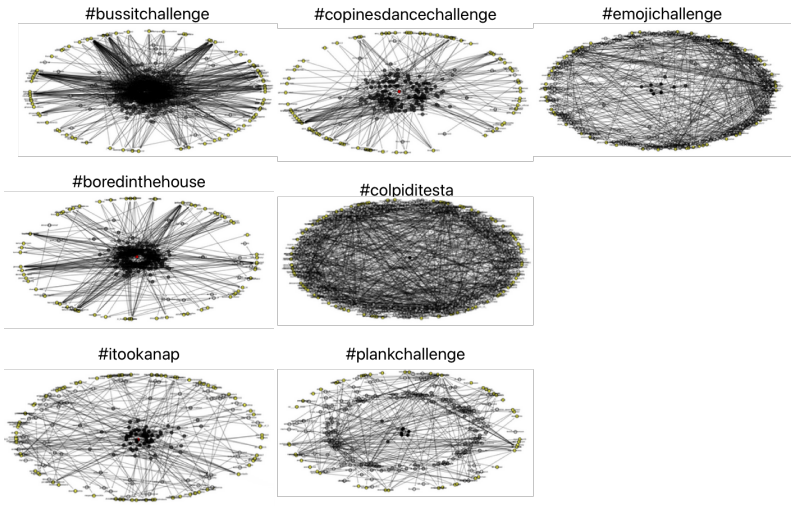


Fig. 12.1: Structure of non-dangerous networks

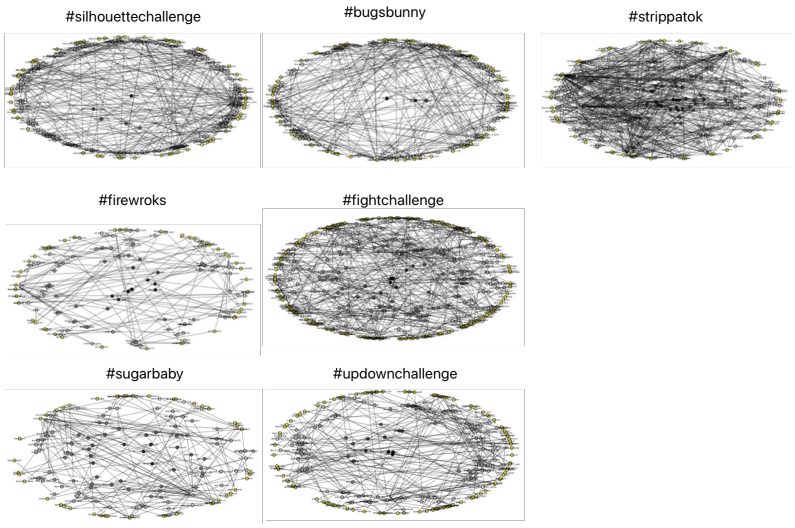


Fig. 12.2: Structure of dangerous networks

which is the core of our paper. In Tables 12.5 and 12.6, we report the values of the basic structural parameters for the two types of networks. The analysis of these tables allows us to draw the following conclusions: *(i)* the size of the networks representing non-dangerous challenges is generally greater than that of the networks associated with dangerous challenges; *(ii)* the average degree and the clustering coefficient of the two kinds of network are comparable; *(iii)* the density of the networks associated with dangerous challenges is higher than the one of the networks associated with non-dangerous challenges.

After examining the characteristics of the networks associated with the two types of challenges, we proceeded to examine their corresponding videos. Their main characteristics are shown in Table 12.7. From the analysis of this table we can deduce that: *(i)* the two types of challenges have videos with similar duration; *(ii)* non-

Challenge	*Number of nodes*	*Number of arcs*	*Average degree*	*Average clustering coefficient*	*Density*
#bussitchallenge	618	708	1.14	0.0047	0.0019
#copinesdancechallenge	237	226	0.96	0	0.0040
#emojichallenge	440	498	1.13	0.0053	0.0026
#colpiditesta	691	843	1.22	0.0015	0.0018
#boredinthehouse	306	309	1.01	0.0018	0.0033
#itookanap	219	201	0.92	0	0.0042
#plankchallenge	271	266	0.98	0.0079	0.0036
Average Value	397.429	435.857	1.051	0.0030	0.0031

Table 12.5: Basic structural characteristics of non-dangerous networks

Challenge	*Number of nodes*	*Number of arcs*	*Average degree*	*Average clustering coefficient*	*Density*
#silhouettechallenge	262	259	0.98	0	0.0037
#bugsbunny	212	239	1.13	0	0.0053
#strippatok	297	519	1.74	0.0025	0.0059
#firewroks	141	111	0.79	0.0083	0.0056
#fightchallenge	409	339	0.83	0.0009	0.0020
#sugarbaby	151	143	0.94	0.0035	0.0061
#updownchallenge	243	199	0.81	0.010	0.0033
Average Value	245	258.429	1.031	0.0036	0.0046

Table 12.6: Basic structural characteristics of dangerous networks

dangerous challenges have a higher average number of music tracks than dangerous challenges; *(iii)* dangerous challenges have a higher average number of likes, comments, shares and views than non-dangerous challenges.

At this point, we looked at the authors of the videos posted for the two types of challenges and examined their main characteristics. These are shown in Table 12.8. From this table we can deduce that: *(i)* the average number of followers is comparable for the two types of authors; *(ii)* the authors of the non-dangerous challenges tend to put more likes, follow many more authors and post many more videos than the ones of the dangerous challenges; *(iii)* the authors of the dangerous challenges receive many more likes than the ones of the non-dangerous challenges.

Finally, we considered the evolution of user communities associated with non-dangerous and dangerous challenges over time. In this preliminary analysis, we focused only on the variation in the number of users. The results obtained are shown in Table 12.9. Examining this table, we can see important differences between non-

Parameter	*Non-dangerous challenges*	*Dangerous challenges*
Average video duration (seconds)	21.39	20.38
Average number of music tracks used in a challenge	208	126.20
Average number of likes	178,104.13	249,152.12
Average number of comments	1,970.03	2,559.98
Average number of shares	5,456.83	6,990.26
Average number of views	1,471,020.16	2,070,632.01

Table 12.7: Differences between the main basic characteristics of videos for non-dangerous and dangerous challenges

Parameter	*Non-dangerous challenges*	*Dangerous challenges*
Average number of likes put by an author	17,730.52	11,998.711
Average number of likes received by an author	7,033,150.71	12,080,102.18
Average number of users followed by an author	1,357.08	670.24
Average number of users following an author	400,593.58	447,762.28
Average number of videos published	384.05	263.13

Table 12.8: Differences between the main basic characteristics of the authors of videos for non-dangerous and dangerous challenges

dangerous and dangerous challenges. First, the average lifespan of dangerous challenges is longer than that of non-dangerous ones. Also, the growth of the number of users in non-dangerous challenges is much more gradual than in dangerous ones. In fact, the latter show a very limited growth up to about 75% of the lifespan. After this limit, the growth becomes explosive.

This preliminary analysis seems to suggest that the communities of users associated with the two types of challenges have very different growth dynamics. Finding out whether this conjecture is true and, if so, investigating these differences in detail and finding evolutionary patterns characterizing them is the core of this paper.

12.2.2 Analysis of the evolution of user communities for non-dangerous and dangerous challenges

In this section, we want to address the core issue of this paper, that is the identification of possible evolutionary patterns that characterize the communities of users

Parameter	*Non-dangerous challenges*	*Dangerous challenges*
Average challenge lifespan (days)	405	550.17
Average number of network nodes at 5% of lifespan	8.6	2.2
Average number of network nodes at 25% of lifespan	140.4	7.6
Average number of network nodes at 50% of lifespan	172	22.4
Average number of network nodes at 75% of lifespan	179.4	58.8
Average number of current network nodes (100% of lifespan)	397.43	245.67

Table 12.9: Differences between the growth of user communities associated with non-dangerous and dangerous challenges

related to TikTok challenges and allow the distinction of the non-dangerous challenges from the dangerous ones.

The first step of this research consists in analyzing the temporal evolution of the 14 challenges stored in our dataset. In particular, we want to determine if the lifespans of the various challenges contain common typical intervals. Examples of such intervals might be: *(i)* the interval in which the challenge is born and a very first community of users begins to develop; *(ii)* the interval in which the challenge is enormously successful and becomes viral; *(iii)* the interval in which the challenge's popularity begins to decline; *(iv)* the interval in which the challenge has become obsolete and is abandoned. In addition, we want to test whether each interval is characterized by very different behaviors from the user communities associated with challenges. Finally, behavioral differences among user communities could occur not only based on the type of intervals, but also, and perhaps most importantly, based on the type (i.e., non-dangerous and dangerous) of challenge.

To begin our research, we considered how the size of each community evolved during the lifespan of the corresponding challenge. As seen in Section 12.1.2, the community associated with each challenge can be modeled as a social network and there is a biunivocal correspondence between the users of a community and the nodes of the corresponding social network.

We now consider a graph whose x-axis represents the lifespan of a challenge and whose y-axis denotes the number of members of the community associated with it or, equivalently, the number of nodes of the corresponding social network. If we subdivide the lifespan into suitable intervals (also very small), consider the number

of social network nodes in correspondence to each interval, find the corresponding points in the diagram and join them, we obtain a broken line. This denotes the variation of the size of the community during the challenge lifespan. We chose a very fine granularity and, in fact, we divided the lifespan into 100 intervals. With this choice, the broken line becomes very detailed, providing a very accurate representation of how the community size varies over time. However, for reasons that will become clear later, we need a continuous function, instead of a broken line. To obtain such a function, we interpolated the points of the broken line using a univariate spline.

To test whether the difference between the broken lines and the curves obtained from the interpolation is acceptable, we computed the Mean Absolute Error (MAE) by considering 100 additional equidistant points for each interval (and, thus, 10,000 points for each lifespan). Then, we normalized the MAE value at each point to the value of the broken line at that point. Table 12.10 shows the results obtained. The analysis of this table reveals that the mean values of the normalized MAE are very low. This allows us to conclude that the interpolation performed by us is acceptable.

Non-dangerous Challenge	*Normalized MAE*	*Dangerous Challenge*	*Normalized MAE*
#bussitchallenge	0.012	#silhouttechallenge	0.017
#copinesdancechallenge	0.015	#bugsbunny	0.017
#emojichallenge	0.021	#strippatok	0.023
#colpiditesta	0.025	#firewroks	0.026
#boredinthehouse	0.011	#fightchallenge	0.014
#itookanap	0.015	#sugarbaby	0.021
#plankchallenge	0.018	#updownchallenge	0.026

Table 12.10: Normalized MAE between the continuous function returned by the univariate spline interpolation and the real values for non-dangerous challenges (at left) and dangerous ones (at right)

To analyze how the communities associated with challenges evolve over time, we found it useful to identify the points of the lifespan where their characteristics change. Since, up to this point, the most important characteristic we know of is community size, this implies considering the points at which the broken line or the corresponding interpolation curve inverts. This is the reason why we chose to use the interpolation curve with the univariate spline. In fact, in this way, we have a continuous function and the points where it inverts are given by the ones where it reaches a maximum or a minimum.

More formally, let C_i be a challenge, let $\mathcal{N}_i$ be the corresponding social network, and let $\nu_i(\cdot)$ be the function representing the change in the number of nodes of $\mathcal{N}_i$ during the lifespan of C_i; in other words, $\nu_i(\cdot)$ is the interpolation curve described above. To identify the points in the lifespan where $\nu_i(\cdot)$ has a maximum or a min-

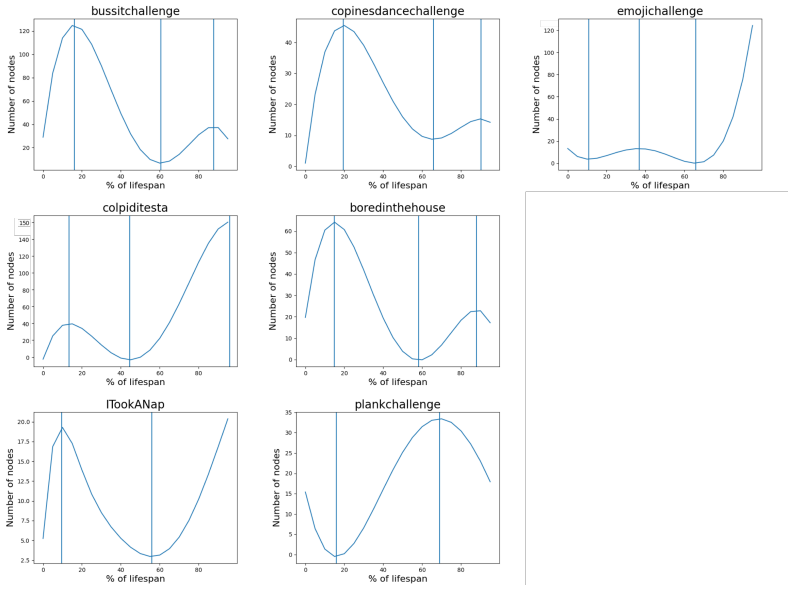


Fig. 12.3: Trends and intervals of $\nu_i(\cdot)$ for non-dangerous challenges

imum, we compute the first derivative $\nu_i'(\cdot)$ of $\nu_i(\cdot)$ and check the points where it becomes null. Let $X_i = \{x_{i_1}, x_{i_2}, \cdots, x_{i_N}\}$ be the set of such points; we can split the lifespan of C_i into $N-1$ intervals (x_q, x_{q+1}), $1 \leq q \leq N-1$, such that $\nu_i(\cdot)$ is always increasing or always decreasing within each of them. As we will see in the following, these intervals represent an essential tool of our analysis because it is from them that we will look for the evolutionary patterns of communities capable of distinguishing non-dangerous challenges from dangerous ones.

Figures 12.3 and 12.4 show the trends of the function $\nu_i(\cdot)$ for each non-dangerous and dangerous challenge, respectively. They also show the corresponding intervals. Already from this first visual analysis, we can observe that, in the two kinds of challenge, the corresponding communities show completely different dynamics. Capturing and formalizing such dynamics represent the objective of the next sections.

Capturing community evolution during a challenge lifespan. In order to capture the evolution of communities during the lifespan of a challenge, it is first necessary to identify features capable of representing this evolution in detail and from multiple perspectives. To this end, we are helped by the social network-based model that we introduced in Section 12.1.2. Thanks to this model, given a challenge C_i, the social network $\mathcal{N}_i$ that represents its community at a given interval $\mathcal{I}$, during which the trend of $\nu_i(\cdot)$ is always increasing or always decreasing, it is possible to identify 19 features of interest. These are:

- `node_number`: number of nodes of $\mathcal{N}_i$;
- `arc_number`: number of arcs of $\mathcal{N}_i$;

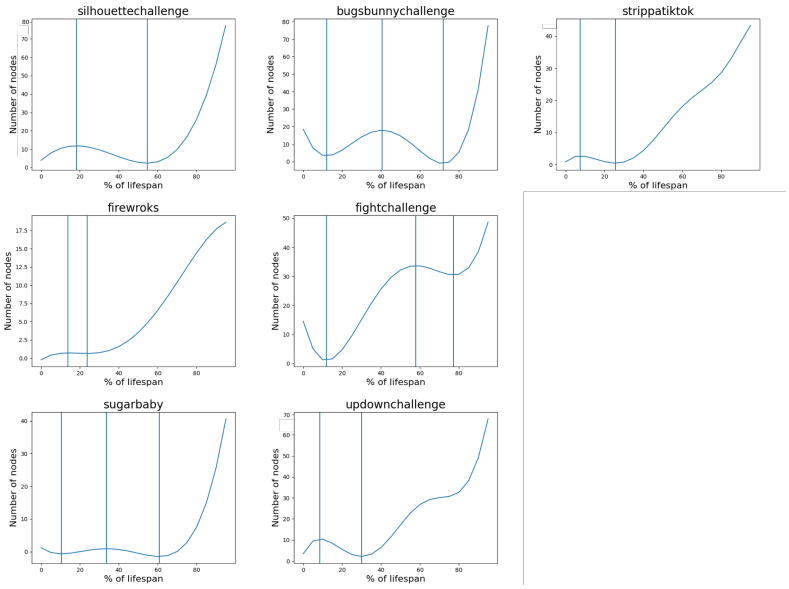


Fig. 12.4: Trends and intervals of $v_i(\cdot)$ for dangerous challenges

- `density`: density of $\mathcal{N}_i$;
- `conn_components_number`: number of connected components of $\mathcal{N}_i$;
- `max_conn_comp_node_number`: number of nodes of the maximum connected component of $\mathcal{N}_i$;
- `avg_indegree_centrality`: average indegree centrality of the nodes of $\mathcal{N}_i$;
- `avg_outdegree_centrality`: average outdegree centrality of the nodes of $\mathcal{N}_i$;
- `avg_eigenvector_centrality`: average eigenvector centrality of the nodes of $\mathcal{N}_i$;
- `avg_pagerank`: average PageRank of the nodes of $\mathcal{N}_i$;
- `avg_closeness_centrality`: average closeness centrality of the nodes of $\mathcal{N}_i$;
- `avg_clustering_coefficient`: average clustering coefficient of the nodes of $\mathcal{N}_i$.
- `radius_max_conn_comp`: radius of the maximum connected component of $\mathcal{N}_i$;
- `diameter_max_conn_comp`: diameter of the maximum connected component of $\mathcal{N}_i$;
- `perc_nodes_in_max_conn_comp`: percentage of nodes of $\mathcal{N}_i$ belonging to its maximum connected component;
- `avg_eccentricity`: average eccentricy of the nodes of $\mathcal{N}_i$;
- `avg_path_length`: average length of the paths of $\mathcal{N}_i$;
- `max_ego_network_node_number`: number of nodes present in the ego-network with the maximum size in $\mathcal{N}_i$;
- `avg_ego_network_node_number`: average number of nodes in the ego-networks of $\mathcal{N}_i$.

As we can see, we have a lot of available features, and managing all of them can be complex. Therefore, we decided to check for possible correlations between them. In fact, if a group of features is correlated, we can consider only one of them and filter out the others. Figure 12.5 shows the correlation matrix we obtained by applying the Pearson's correlation [100] to the pairs of features identified above.

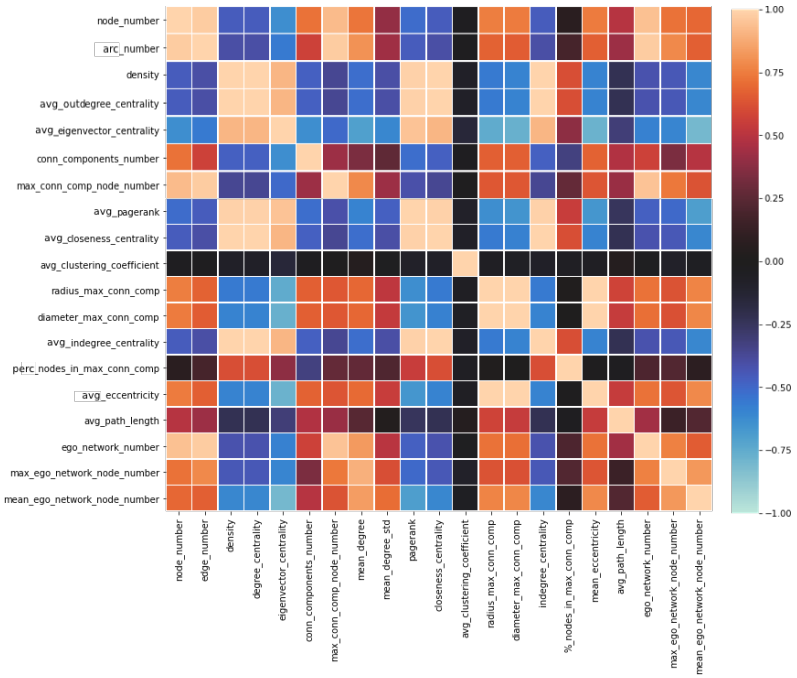


Fig. 12.5: Correlation matrix for the 20 features representing the behavior of the communities during a challenge

Considering the various groups of correlated features and choosing one for each group, we identified the following features to maintain for the next analyses:

- `conn_components_number`;
- `avg_indegree_centrality`;
- `avg_outdegree_centrality`;
- `avg_clustering_coefficient`;
- `perc_nodes_in_max_conn_comp`;
- `avg_path_length`;
- `avg_ego_network_node_number`.

Detecting the similarities and differences of the evolutionary dynamics of communities. In the previous section, we have identified a list of features able to describe the behavior of the community of users associated with a challenge during a time interval. In this section, we want to use these features to group the intervals

related to the lifespan of the 14 challenges of our dataset into clusters being homogeneous from the perspective of the evolutionary dynamics of the communities involved.

First of all, we considered a new dataset formed by a single table whose rows represent the interval of the 14 challenges under consideration and whose columns are associated to the 7 selected features. The element (h,k) of this table indicates the value assumed by the k^{th} feature in the h^{th} interval. The presence of 7 features (and, therefore, of 7 dimensions) with a limited number of elements to cluster made this last task very difficult to carry out. To address this issue, we applied the Principal Component Analysis (hereafter, PCA) [294] to the dataset. In this way, we were able to reduce the dimensions from 7 to 2. This gave us a further advantage, as it allowed us to visualize the elements to be clustered and the corresponding clusters in a bidimensional plane.

After the application of PCA, we applied a clustering technique to group the segments into homogeneous clusters from the user community behavior perspective. Specifically, we chose the Expectation Maximization (hereafter, EM) clustering algorithm. The reason for this choice lies in the fact that this algorithm, among the various positive properties characterizing it, also has that of being able to automatically determine the number of clusters [294]. This property was particularly important in our case because it was not possible to make any a priori conjecture on this number, and the application of the elbow method carried out with k-means returned no results. Applying Expectation Maximization to our dataset we obtained four clusters, shown in Figure 12.6. This representation refers to a bidimensional plane whose axes correspond to the two dimensions returned by PCA. Once clusters were identified, we tried to understand what each of them represented in terms of the behavior and the dynamics of the challenge communities during the time intervals belonging to it. At the end of this activity, we drew the following characterizations:

- *Cluster A*: during the intervals belonging to this cluster, the networks are characterized by a quite high number of nodes and a high number of connected components. The nodes of each connected component have a high average indegree and average outdegree. This implies that the corresponding communities consist of highly connected users. Confirming the latter property, the average size of the ego networks is large and the average clustering coefficient is high.
- *Cluster B*: during the intervals belonging to this cluster, the networks are characterized by a very high number of nodes (more than twice the number of nodes in Cluster A) and a rather high number of connected components (although less than in Cluster A). The maximum connected component includes most of the nodes, while the other ones are all made up of few nodes, albeit their number

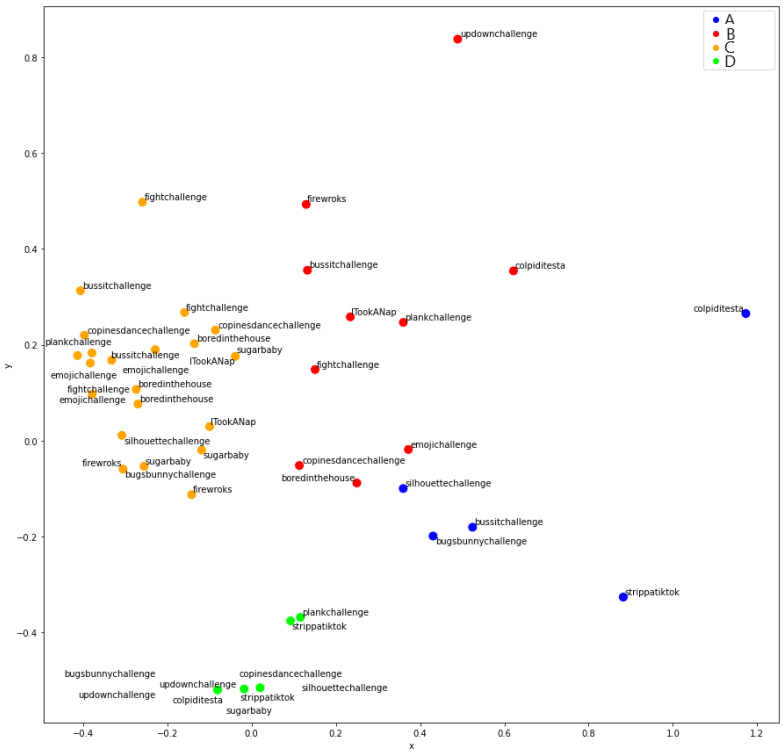


Fig. 12.6: The four clusters of intervals returned by Expectation Maximization

is still high. The average clustering coefficient and the average size of the ego networks remain very high, even if this is mainly due to the contribution of the nodes of the maximum connected component.

- *Cluster C*: during the intervals belonging to this cluster, the networks are characterized by a limited number of nodes and a certain number of connected components. The nodes of each connected component have a small-medium average indegree and average outdegree. The average size of the ego networks is small and the average clustering coefficient is medium-small.
- *Cluster D*: during the intervals belonging to this cluster, the networks have a high number of nodes and a high number of connected components. The nodes of each connected component have a medium average indegree and average outdegree. The average size of the ego networks is medium-high and the average clustering coefficient in medium-high.

In Figure 12.7, we show an example of the structure of a user community associated with a challenge for each cluster.

12.2.3 Searching for evolutionary patterns in the challenge lifespans

After grouping the intervals into clusters, and after identifying the characteristics of each cluster, we tested whether there were evolutionary patterns characterizing the

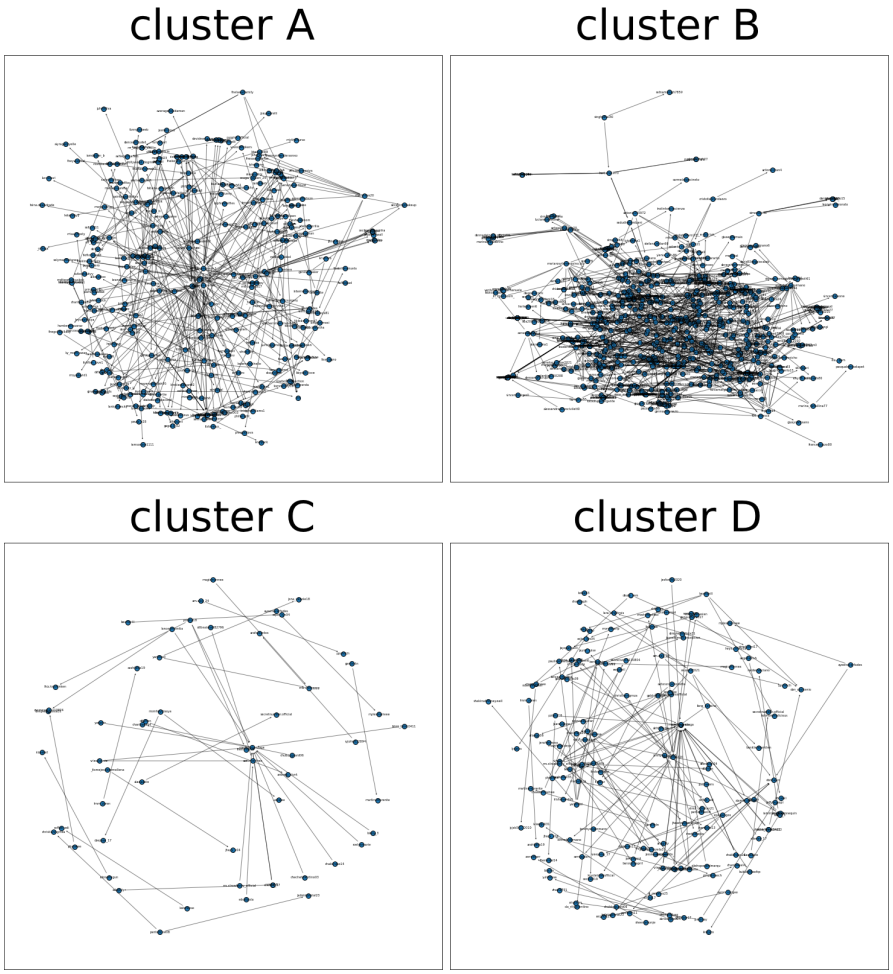


Fig. 12.7: Example of the structure of a user community associated with a challenge for each cluster

communities of non-dangerous and dangerous challenges while also allowing us to distinguish one from the other. To this end, we considered the lifespans of the 14 challenges of the dataset and, for each of the corresponding intervals, we recorded the cluster to which it belonged. If two consecutive intervals belonged to the same cluster, we recorded them only once. A the end of this process, we obtained the sequences of intervals shown in Table 12.11.

Examining these sequences, we can draw some observations. In particular:

- In the non-dangerous challenges, there is no dominant pattern although intervals of type C and D are frequent. Specifically, an interval of type D is present in each non-dangerous challenge.
- Dangerous challenges always begin with an interval of type C whereas they end with intervals of type A, B or D.

Examining the description of clusters in Section 12.2.2, we can note that the user communities during the intervals belonging to clusters A and B have similar features.

Non-dangerous Challenge	*Evolutionary Paths*	*Dangerous Challenge*	*Evolutionary Paths*
#bussitchallenge	C, B, D	#silhouttechallenge	C, A
#copinesdancechallenge	C, A, B, D	#bugsbunny	C, D
#emojichallenge	A, B, D	#strippatok	C, D
#colpiditesta	C, A, D	#firewroks	C, A, B
#boredinthehouse	A, D	#fightchallenge	C, A
#itookanap	C, A, D	#sugarbaby	C, A, D
#plankchallenge	C, B, D	#updownchallenge	C, B

Table 12.11: Sequences of intervals for non-dangerous and dangerous challenges

Also examining Figure 12.6 we can see that cluster B can be seen as an extension of cluster A. Therefore, we decided to examine the data corresponding to the intervals of these clusters in more detail. We have previously seen that:

- The intervals of cluster A are characterized by networks with a high number of connected components. The average indegree and outdegree of the network nodes are high. As a result, during these intervals, there are many connections between users. This is also witnessed by the average clustering coefficient that is very high.
- The intervals of type B are characterized by network with a rather high number of connected components and high average indegree and outdegree of the network nodes. The main difference with the intervals of type A is that, in this case, the maximum connected component contains most of the network nodes. In fact, the other connected components generally consist of pair of nodes.

Despite the main difference mentioned above, and other small existing ones, we can hypothesize that the two clusters of intervals A and B represent the same reality. In particular, given the high average indegree, average outdegree, average clustering coefficient and the large size of ego networks, we can hypothesize that these intervals represent the peak of the evolution of a challenge.

In order to confirm or reject our hypothesis, we performed a t-test [100], based on the following null hypothesis: H0: "There is no statistically significant difference between the intervals of clusters A and B". Prior to performing it, we had to test whether the items in the two samples had comparable variances or not. In fact, this step is necessary to choose whether to perform the classical t-test (used when the two samples have comparable variances) or the Welch's t-test (used otherwise) [100]. In order to decide on the comparability of the variances of the intervals of the clusters A and B, we performed the Bartlett's t-test [60]. It allows us to determine whether two samples with different numbers of items have the same variance or not. More formally, we applied the Bartlett's t-test with the following null hypothesis: H0: "The sets of intervals belonging to clusters A and B have the same variance". We com-

puted the corresponding p-value and saw that it was equal to 0.52, which is much higher than the classical threshold of 0.05 generally considered for this parameter. Therefore, the null hypothesis was confirmed. As a consequence of this fact, in order to test whether the difference between the intervals of clusters A and B was statistically significant, we had to adopt the classic t-test and not the Welch's one.

Applying the classic t-test on the null hypothesis H0: "There is no statistically significant difference between the intervals of clusters A and B", we obtained a p-value of 0.63. This is much greater than 0.05 and allowed us to conclude that the null hypothesis was confirmed. In turn, this implied that the clusters A and B were statistically equivalent and represented two very similar scenarios, despite the previously highlighted differences.

Thanks to this result, it was possible to substitute A for B in all the interval sequences of the challenges under consideration.

Observe that, after determining the equivalence betweeen the intervals of A and B, we have three kinds of interval, namely: *(i)* intervals of type A, whose characteristics described above suggest that they correspond to the peak of a challenge; *(ii)* intervals of type C, whose characteristics suggest that they are the initial ones in a challenge; *(iii)* intervals of type D, whose characteristics suggest that they are the ones relating to the end of the lifecycle of a challenge.

Now, after the substitution of B with A, and recalling that our evolutionary pattern model states that two consecutive intervals of the same cluster are represented only once, the sequences of intervals that characterize non-dangerous challenges are shown in Table 12.12.

Non-dangerous Challenge	*Evolutionary Paths*	*Dangerous Challenge*	*Evolutionary Paths*
#bussitchallenge	C, A, D	#silhouttechallenge	C, A
#copinesdancechallenge	C, A, D	#bugsbunny	C, D
#emojichallenge	A, D	#strippatok	C, D
#colpiditesta	C, A, D	#firewroks	C, A
#boredinthehouse	A, D	#fightchallenge	C, A
#itookanap	C, A, D	#sugarbaby	C, A, D
#plankchallenge	C, A, D	#updownchallenge	C, A

Table 12.12: Sequences of intervals for non-dangerous and dangerous challenges after the verification of the hypothesis that A and B are equivalent

Thanks to this result, we were able to identify some evolutionary patterns characterizing non-dangerous and dangerous challenges. Furthermore, since these evolutionary patterns are different in the two cases, they also allow us to distinguish non-dangerous challenges from dangerous ones.

Let us first examine non-dangerous challenges. In this case, we always have the presence of a sequence of intervals of type A, D. This sequence is very often preceded by an interval of type C, so that we have an evolutionary pattern of type C, A, D. We argued that the typical evolutionary pattern of a non-dangerous sequence is C, A, D. In fact, the challenges showing only the sequence A, D already existed when we started to collect their data in our dataset. Therefore, we assumed that our investigation of them started too late to also capture the initial interval of type C. This is also confirmed by the fact that all TikTok challenges (both non-dangerous and dangerous ones), which we observed since their inception, always exhibited a startup phase before reaching their peak or their decline.

Let us now examine dangerous challenges. In this case, unlike the previous one, there is no single sequence of intervals characterizing all of them. Instead, we identified three dominant sequences that correspond to three different "fates" generally characterizing the challenges of this type. In particular, the three evolutionary patterns are:

- C, A: these challenges had a standard initial phase with an interval of type C; then, they reached a peak phase and suddenly stop. This was probably due to the fact that, being dangerous, they were suppressed by TikTok itself.
- C, D: these challenges had an initial phase, which was followed by a decay one. In other words, they never reached the peak. They were born, survived for a certain period on the network, and then died.
- C, A, D: as we will see below, these challenges are a small minority among the dangerous ones. They behaved like the non-dangerous ones, in that they were born, had a peak and, finally, decayed.

In order to verify the goodness of our results, we decided to test them on a set of new challenges, different from the previous ones. In particular, we considered 300 challenges (150 non-dangerous and 150 dangerous ones). The results obtained are the following:

- As for non-dangerous challenges:
  - 134 (i.e., 89.33% of them) followed the evolutionary pattern B, C, A. This is the only significant one we identified for this type of challenges.
  - 16 (i.e,, 10.67% of them) followed a variety of other sequences of intervals.
- As for dangerous challenges:
  - 65 (i.e., 43.33% of them) followed the evolutionary pattern C, A;
  - 64 (i.e., 42.67% of them) followed the evolutionary pattern C, D;
  - 11 (i.e., 7.33% of them) followed the evolutionary pattern C, A, D;
  - 10 (i.e., 6.67% of them) followed a variety of other sequences of intervals.

The results obtained represent a confirmation that the evolutionary patterns we detected actually exist for the two types of challenges into consideration and are capable of discriminating them. In addition, these results show that the patterns we found are really able to capture almost all the behaviors of the communities of TikTok challenges.

Part II

Networking things

In this part, we apply Social Network Analysis concepts, parameters and approaches to smart objects. In particular, we investigate: (i) the usage of connected smart objects for fall detection in a workplace in Chapter 13; (ii) anomalies in Multiple IoT scenarios in Chapter 14; (iii) protection and autonomy of smart objects in the IoT in Chapter 15; (iv) saliency map and gaze prediction in an Industry 4.0 scenario in Chapter 16.

13

Networking wearable devices for fall detection in a workplace

In the last few decades, we have witnessed an increasing focus on safety in the workplace. ICT has always played a leading role in this context. One ICT sector that is increasingly important in ensuring safety at work is the Internet of Things and, in particular, the new architectures referring to it, such as SIoT, MIoT and Sentient Multimedia Systems. All these architectures handle huge amounts of data to extract predictive and prescriptive information. For this purpose, they often make use of Machine Learning. In this chapter, we propose a framework that uses both Sentient Multimedia Systems and Machine Learning to support safety in the workplace. After the general presentation of the framework, we describe its specialization to a particular case, i.e., fall detection. As for this application scenario, we describe a Machine Learning based wearable device for fall detection that we designed, built and tested. Moreover, we illustrate a safety coordination platform for monitoring the work environment, activating alarms in case of falls, and sending appropriate advices to help workers involved in falls.

The material presented in this chapter was derived from [218].

13.1 Framework description

The overall architecture of the proposed framework is shown in Figure 13.1. It assumes that the global working environment is partitioned in several areas where workers operate. These areas, along with their corresponding smart objects, are fixed. Instead, workers, with their wearable smart objects, move from one area to another over time.

As can be seen from Figure 13.1, this architecture consists of three distinct layers, namely:

- *Personal Devices*: these are smart objects worn by workers. They have a twofold objective, i.e., supporting a worker in her/his work activities and guaranteeing the maximum safety at all times.

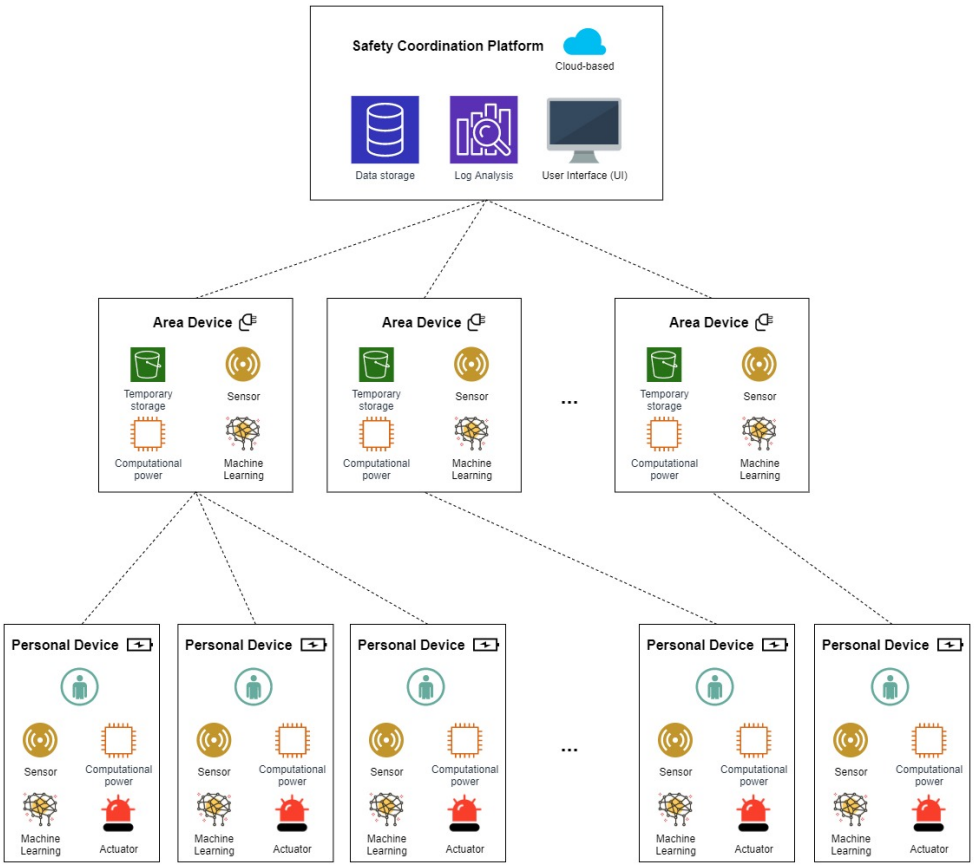


Fig. 13.1: The overall architecture of the proposed framework

- *Area Devices*: these are fixed smart objects, each associated with a specific area. They aim at constantly monitor the area which they belong to, in order to support safety. For this purpose, they process both the data produced by themselves and the ones coming from the Personal Devices of the workers present in the area at that moment.
- *Safety Coordination Platform*: it represents the highest layer of our framework. It receives data from all the Area Devices of the working environment and is responsible for processing these data to ensure the overall safety of the whole environment.

The communication between the Area Devices and the Safety Coordination Platform is point-to-point, while the communication between Area Devices is broadcast. The same happens for the communication between Personal Devices, as well as for the one between Area Devices and Personal Devices.

In a certain area, there is only one Area Device of a given type, while there may be several Area Devices of different types. At a certain time, a Personal Device can communicate with the Area Devices of the area where it is located. As a result, it can exchange data with multiple Area Devices, each having a type different from

the ones of the other Area Devices. However, given a certain type, it can communicate only with the Area Device of that type, located in the area where the operator wearing it is working at that moment.

Obviously, a Personal Device worn by the operator moves with her/him. Therefore, when she/he moves from one area to another, the Device Areas with which her/his Personal Device communicates, also change.

In the next subsections, we provide a more detailed description of the devices that make up our framework.

13.1.1 Personal Devices

The focus of Personal Devices is the individual worker. Their goal is supporting the operator wearing them in all her/his activities, ensuring her/his safety at all time. For example, at this layer of our framework, we can find devices for augmented reality, which aim at showing the user how to perform certain operations, devices for taking the minimum path to evacuate an area or to reach an injured colleague, or devices for the detection of falls, to promptly report any injuries from tripping, slipping, etc. Figure 13.2 provides an overview of these devices.

Fig. 13.2: An overview of Personal Devices available for a worker

A smart object belonging to the Personal Device layer of our framework must meet some requirements. More specifically:

- It must be able to collect data that allows the derivation of information about the worker who is wearing it and the area where she/he is operating. For example,

such a smart object could have accelerometric and gyroscopic sensors, to determine the motion or activity that the worker is performing, and/or a camera, to monitor her/him while she/he is performing some dangerous activities.

- It must have enough computational power to ensure an initial analysis of data retrieved by it. This feature makes this first layer of our framework an edge-computing network, and therefore a network of smart objects capable of performing real time analysis without the aid of a cloud service. Many of the data analyses performed by smart objects is based on Machine Learning. An example of how Machine Learning can be used to analyze data and make decisions within these smart objects is shown in Section 13.2.1.
- It must be able to carry out "actions" helping the operator in her/his activities or signaling a danger or, even, sending an alarm (for instance, in the event of an accident to the worker). For example, a smart object that monitors the position of the operator must be able to issue vibrations, sounds or activate appropriate LEDs, to report her/his possible entry into a restricted access area or the presence of another operator at a distance less than that required by safety regulations against COVID-19. This implies that each of these smart objects must be equipped with one or more actuators.
- It must have the ability to communicate with other smart objects, as well as with other kinds of device.
- It must be powered by a battery and it must be able to continuously monitor the corresponding charge, in order to alert the operator if a recharge is needed.
- It must have a low power consumption, avoiding as much as possible the need to recharge the battery during a work shift of its operator.
- It must be wearable and, if possible, non-invasive.

13.1.2 Area Devices

The focus of Area Devices is the monitoring of a specific area in a working environment. To achieve this goal, the smart objects belonging to this category leverage both the data produced by them and the one sent by the Personal Devices present in the area. The ultimate goals of the monitoring performed by them are the prevention of accidents, the optimization of environmental parameters to improve the quality of the operators' work, the control of access to specific reserved areas, and so forth.

An example of an Area Device is represented by the fixed smart objects for the detection of falls through video that, as we will see, can be used in parallel with Personal Devices for fall detection. A second example could be a smart object for the analysis of vibrations in structures and plants, able to detect an imminent failure of

the structure or an imminent danger and to warn immediately the workers who are in the area.

A smart object belonging to the Area Device layer of our architecture must meet certain requirements. More specifically:

- It must contain some sensors that are able to measure environmental parameters, such as temperature, brightness, presence of specific gases, etc.
- It should communicate with other smart objects and with the Safety Coordination Platform through various modes and protocols, like Bluetooth, WiFi, UWB (Ultra Wide Band), etc.
- It must have a computational power allowing it to carry out real-time data analyses and to support the execution of Machine Learning and other Artificial Intelligence techniques in order to perform real-time predictions.
- It must be able to temporarily save some data in order to keep track of communications with other Area Devices and Personal Devices. For example, it must need to store that, at a certain time, there was a gas leak in the area of interest or that, in the same area, the temperature was above a certain threshold.
- It must be able to be connected to the power grid. Furthermore, it should be equipped with a backup battery in case of a power failure.

Area Devices represent the intermediate layer of the proposed framework and, therefore, play a key role in guaranteeing the communication between the other two layers, i.e., Personal Devices and Safety Coordination Platform.

13.1.3 Safety Coordination Platform

The Safety Coordination Platform represents the highest layer of our framework. It aims at monitoring the situation of the whole working environment based on the data provided by the Area Devices. For this purpose, it carries out the appropriate data analysis and makes the suitable decisions regarding any alarms and/or requirements. This is possible thanks to its cloud-based nature, which allows it to be accessed at any time while ensuring an excellent level of scalability and availability.

In order to be able to perform all the activities required, the Safety Coordination Platform must be capable of saving large amounts of data in the form of logs, event alerts, structured databases storing the characteristics and positions of the devices, and so on. Some of these data must be processed continuously in real time while others must be considered only for particular knowledge extraction activities. For this reason, the Safety Coordination Platform is equipped with a data lake, appropriate algorithms for the extraction of semantic relationships between data stored

in different sources (such as synonymies, homonymies, etc.), and data integration techniques, like the ones described in [272, 199, 124, 125].

Given its cloud-based nature, a possible implementation of the Safety Coordination Platform could involve Apache Kafka[1] installed on a cloud node (or on a cluster of nodes, depending on the size of the scenario under consideration). Kafka aims at managing the data flow produced by Personal Devices and Area Devices. The Kafka Publisher saves the flow data in the data lake; the latter could be managed through Kylo[2]. Data stored in the data lake can be processed through the ELK stack[3]. In particular, Logstash can be used to extract data from the data lake, perform some cleaning operations on it and pass cleaned data as input to Elasticsearch. The latter can be used to perform the appropriate analyses on the data received from Logstash. Finally, Kibana can be adopted to create different dashboards to display the data processed by Elasticsearch, to monitor the overall workspace environment and to set alarms based on control thresholds.

Based on all available data, the Safety Coordination Platform can, first of all, carry out descriptive and diagnostic analyses [173]. Thanks to them, it can monitor and show in real time the values of a set of Key Performance Indicators (hereafter, KPIs) that describe the situation of the working environment areas. Examples of KPIs that could be adopted in this case are the number of reported accidents and incidents, the time injury frequency rate, the time injury incidence rate, the number of equipment breakdowns, and so on. Overall, KPIs regard the work performance of each area and, above all, the safety of the operators who are working within it. Monitored data are represented in a dashboard and, whenever one KPI exceeds a certain threshold, the Safety Coordination Platform activates the corresponding alerts through the appropriate mechanisms and actuators with which it is equipped.

The presence of a data lake, as well as of high and flexible computing power, allows the implementation, within the Safety Coordination Platform, of appropriate Machine Learning and Artificial Intelligence based approaches, capable of supporting predictive and prescriptive analyses. The results of these analyses give rise to

[1] Apache Kafka (`https://kafka.apache.org/`) is an open source distributed event streaming platform used for high-performance data pipelines, streaming analytics, data integration and mission-critical applications.

[2] Kylo (`https://www.kylo.io/`) is an open source data lake management platform for self-service data ingestion and data preparation with integrated metadata management.

[3] ELK stack (`https://www.elastic.co/what-is/elk-stack`) comprises three open source projects: *(i)* Elasticsearch, which is a search and analytics engine; *(ii)* Logstash, which is a server-side data processing pipeline that ingests data from multiple sources, transforms it and sends it to Elasticsearch; *(iii)* Kibana, which lets users visualize data with charts and graphs in Elasticsearch.

appropriate knowledge patterns that can support decision makers in taking a set of actions to further improve the safety of the working environment.

For example, a thorough analysis of the logs could help to better tune the values of the actuator activation parameters in order to minimize the presence of false positives while keeping false negatives as low as possible (or, better, equal to zero). As a second example, the analysis of the various sensors of brightness, temperature, humidity, etc., relative to a year just passed, can lead to suggest a series of actions to be taken in certain areas of the working environment in order to improve the thermo-hygrometric well-being of the operators working therein.

13.2 Specialization of the proposed framework to fall detection

In Section 13.1, we have provided a general description of our framework. In this section, we want to show its behavior in detail, defining and testing the suitable Machine Learning algorithms for the extraction of knowledge about safety at work from the available data. In order to provide a detailed description of both the algorithms and the experiments, we must focus on a particular case of safety at work; for this purpose, we choose fall detection. This choice is motivated by the fact that some of the main causes of accidents in workplaces all over the world are slips, trips and falls. As shown in Section **??**, there are three different kinds of technique developed for fall detection, namely ambient sensor based, vision based, and wearable device based [450]. In the following, we focus on wearable device based techniques.

In the description of how our framework handles fall detection, we put a particular emphasis on the Personal Devices layer, as we have designed, built and tested an ad hoc smart object that implements Machine Learning techniques for fall detection starting from the data it derives through its sensors. Instead, as far as the Area Devices layer is concerned, we have used some fixed devices for fall detection already existing. Finally, as for the Safety Coordination Platform layer, we have defined a chain of Machine Learning based modules that uses the data provided by both the Personal Devices and the Area Devices to decide whether or not to activate an alarm and, in the affirmative case, to coordinate rescue operations.

13.2.1 Personal Devices for fall detection

In this section, we describe a Personal Device for fall detection that we designed, built and tested. Since it is based on Machine Learning, we first had to build a support dataset; we illustrate it in Subsection 13.2.1. Then, we had to make some descriptive analyses on the available dataset in order to better understand the reference context and the problems to face; we describe them in Subsection 13.2.1.

Starting from the results of these analyses, we could define the Machine Learning algorithms that our device could have implemented; we report them in Subsection 13.2.1. Once determined the best algorithms, we had to identify the most suitable hardware to implement them; we discuss this issue in Subsection 13.2.1. After having chosen the appropriate hardware, we had to embed our application logic on it; we discuss this activity in Subsection 13.2.1. Finally, once realized the device, we had to test it; we discuss our testing activity in Subsection 13.2.1.

Support dataset outline. In recent years, scientific community has highly explored wearable device based approaches for fall detection, especially due to the pervasive diffusion of portable devices, like smartphones, smartwatches, etc [126]. Many public datasets can be found online to perform analyses on slips, trips and falls, but also to define new approaches for their detection and management. We chose four datasets for our training and testing phases among all those analyzed. In particular, we selected those datasets that would help us define a generalized model, able to comply with the different activities performed by workers and operators of various sectors, who normally make very different moves during their tasks.

"SisFall: a Fall and Movement Dataset" (hereafter, SisFall) is the first dataset used. It was created by SISTEMIC, the Integrated Systems and Computational Intelligence research group of the University of Antioquia [594]. This dataset consists of 4505 files, each referring to a single activity. Activities are grouped in 49 categories; 19 of them are ADLs (Activities of Daily Living) performed by 23 adults, 15 are falls (Fall Activities) that the same adults had, and 15 are ADLs carried out by 14 participants over 62 years old. All the data were collected using a device placed on the volunteers' hips. This device includes different kinds of accelerometer (ADXL345 and MMA8451Q) and a gyroscope (ITG3200).

"Simulated Falls and Daily Living Activities" (hereafter, SFDLAs) is the second dataset used. It was created by Ahmet Turan Özdemir of the Erciyes University and by Billur Barshan of the Bilkent University [483]. It is made up of 3,060 files regarding 36 different activities carried out by 17 volunteers. Each task was repeated about 5 times by each volunteer. Specifically, the 36 activities are 20 Fall Activities and 16 ADLs. All data was recorded using 6 positional devices placed on the head, chest, waist, right wrist, right thigh, and right ankle of each volunteer. The wearable device was made up of an accelerometer, a gyroscope and a magnetometer.

"CGU-BES Dataset for Fall and Activity of Daily Life" (hereafter, CGU-BES) is the third dataset used. It was created by the Laboratory of Biomedical Electronics and Systems of the Chang Gung University [135]. It consists of 195 files referring to

15 volunteers who performed 4 Fall Activities and 9 ADLs. All data was collected using both an accelerometer and a gyroscope.

"Daily and Sports Activities Dataset" (hereafter, DSADS) is the fourth, and last, dataset used. It was created by the Department of Electrical and Electronic Engineering of the Bilkent University [30]. It is a collection of 9,120 files regarding 152 activities carried out by 8 volunteers. Each activity, which lasted about 5 minutes, was split into 5 seconds long recordings. Contrary to the other three datasets, this regards sport activities. The reason why we chose this dataset is to make our model generalizable, more resilient to the various situations occurring in a working environment. All data were collected with the usage of 5 sensors with an accelerometer, a gyroscope and a magnetometer. Each of them was placed on different parts of the volunteer's body.

The information we used was the one extrapolated from the accelerometers and gyroscopes. The reasons of this choice were two. The first one is data availability; indeed, acceleration and rotation were measurements found in all datasets. The second one concerns the better performances obtained by Machine Learning models than thresholding based models when using accelerometric data [271]. Acceleration and rotation data from the four datasets were merged to obtain a new dataset. It consists of a table with six columns reporting the values of acceleration and rotation along the X, Y, and Z axes. The structure of this table, together with some example tuples, is shown in Table 13.1. It is made up of 8,579 activities, where 4,965 are not falls while 3,614 are falls. Each file, linked to an activity, stores all the values of the 6 parameters considered for some samples.

$Acceleration_X$	$Acceleration_Y$	$Acceleration_Z$	$Rotation_X$	$Rotation_Y$	$Rotation_Z$
14.529	67.413	-12.506	18,271	-955.762	-9.447
14.383	65.208	-12.375	14.776	-951.406	-4.152
14.310	65.671	-15.453	13.564	-950.841	-7.296
15.674	68.120	-13.910	19.656	-948.253	-4.601
14.814	68.475	-15.168	19.234	-949.437	-6.797

Table 13.1: Structure and some example tuples of the merged dataset

Obviously, as the data comes from different datasets, the number of samples for each activity is not homogeneous; indeed, it is determined by the length of the activity and the sampling frequency used in the original dataset it comes from. However, different activity lengths and sampling frequencies do not significantly affect the final result, as long as sampling frequency is much higher than the activity length. This is true in all our datasets, because our features are barely influenced by the number of samples available. This happens not only for the maximum and the mini-

mum values, which is intuitive, but also for the mean value and the variance, because the more the number of samples increases the more both the numerator and the denominator of the corresponding formulas grow.

In order to reduce as much as possible the noise from our dataset, we applied a Butterworth Infinite Impulse Response (IIR) second order low-pass filter, with a cut-off frequency of 4 Hz to it. In this task, we kept the frequency response module as flat as possible in the passband. The Butterworth filter (also known as maximally flat magnitude filter) was chosen by us for its simplicity and low computational cost [316]. This filter was first described by Stephen Butterworth [113] in 1930. Its frequency response is maximally flat in the passband and rolls off toward zero in the stopband. These features make it perfect for our case of interest and for a future hardware implementation. In addition to using the Butterworth filter, we deleted all the excess data and made the appropriate adjustments to it by performing some Extraction, Transformation and Loading (ETL) activities. More specifically: *(i)* we removed all the rows containing null values; *(ii)* we removed all the columns not containing accelerometric and gyroscopic data; *(iii)* we replaced all the commas with decimal points; *(iv)* we trimmed all the blank values.

We then proceeded to the feature engineering phase. Specifically, we considered 4 features of a parameter ζ, which sampled data was in our dataset. Those 4 features are the maximum value, the minimum value, the mean value and the variance of ζ. Given n the number of samples of ζ in our dataset and $\zeta[k]$ the value of the k^{th} sample of ζ, $1 \leq k \leq n$, the definition of the 4 features is shown in Table 13.2.

Feature	*Definition*
Maximum Value	$\max_{k=1..n}(\zeta[k])$
Minimum Value	$\min_{k=1..n}(\zeta[k])$
Mean Value	$\mu = \frac{1}{n}\sum_{k=1}^{n}\zeta[k]$
Variance	$\sigma^2 = \frac{1}{n}\sum_{k=1}^{n}(\zeta[k]-\mu)^2$

Table 13.2: Feature definition

As shown in Table 13.1, our dataset contains 6 parameters, corresponding to the values of the X, Y and Z axes returned by the accelerometer and the gyroscope. So that, each activity is described by 24 features in total, having 4 features for 6 parameters at disposal.

Finally, in a very straightforward way, we can label each activity with one of two classes, which possible values are *Fall Activity* and *Not Fall Activity*.

We obtained an $8,579 \times 25$ matrix, representing the training set used to perform the next classification activity.

Preliminary analyses on the support dataset. In this section, we present some of the analyses done on the support dataset. They allowed us to better understand the context we were working in and to better face the next challenges. First, we computed the correlation matrix between the features. It is reported in Figure 13.3.

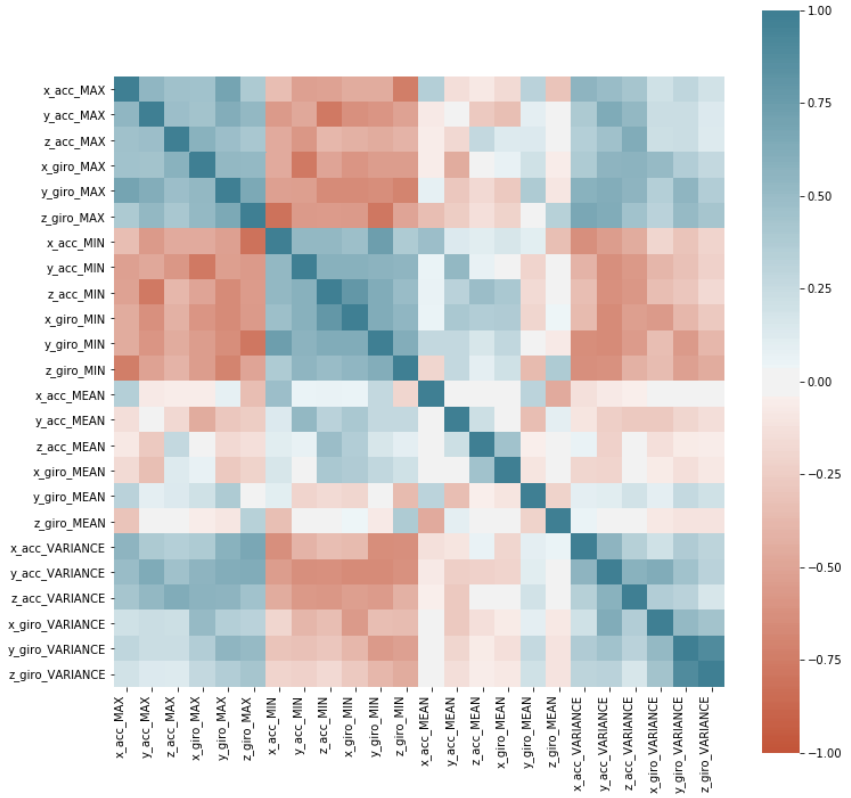


Fig. 13.3: Correlation matrix between the features

Looking at this matrix, it is evident that some negative correlations exist between the maximum and minimum values of some parameters. In addition, we can notice a positive correlation between the maximum values (resp., minimum values, variances) computed on the various axes and on the two sensors. On the other hand, some parameters have neither positive nor negative significant correlations. A result particular evident is where the feature "mean value" is involved, because in all these

cases the correlation is almost always null. This suggests that these last parameters would have been crucial in the next classification activity.

In order to verify this last intuition, we generated a list of features using a Random Forests algorithm [99] with a 10-Fold Cross Validation [294]. The features of the list were sorted according to their relevance in identifying the correct class of activities.

The algorithm used, given a decision tree $\mathcal{D}$ having N nodes, computes the relevance ρ_i of a feature as the decrease of the impurity of the nodes splitting on f_i weighted by the probability of reaching them [268]. The probability of reaching a node n_j can be computed as the ratio of the number of samples reaching n_j to the total number of samples. The higher ρ_i, the more relevant f_i will be. Formally speaking, ρ_i can be computed as:

$$\rho_i = \frac{\sum_{n_j \in N_{f_i}} \vartheta_j}{\sum_{n_j \in N} \vartheta_j}$$

Here, N_{f_i} is the set of the nodes of N splitting on f_i. ϑ_j is the relevance of the node n_j. If we assume that n_j has only two child nodes n_l and n_r, then:

$$\vartheta_j = w_j C_j - w_l C_l - w_r C_r$$

Here:

- w_j (resp., w_l, w_r) is the fraction of samples reaching the node n_j (resp., n_l, n_r);
- C_j is the impurity value of n_j;
- n_l (resp., n_r) is the child node derived from the left (resp., right) split of n_j.

The value of ρ_i can be normalized to the real interval [0,1]. To do this, it must be divided by the sum of the relevances of all features.

$$\overline{\rho_i} = \frac{\rho_i}{\sum_{f_k \in F} \rho_k}$$

where F denotes the set of all the available features.

The final relevance of a feature f_i returned by Random Forests is obtained by averaging the values of the normalized relevances $\overline{\rho_i}$ computed on all the available trees:

$$\widehat{\rho_i} = \frac{\sum_{t_q \in T} \overline{\rho_i}}{|T|}$$

Here, T is the set of all the trees returned by Random Forests.

Table 13.3 shows the result obtained with the approach explained above applied to the features of our interest.

Feature	*Relevance*
y_acc_MEAN	0.2435
y_acc_MAX	0.1877
x_acc_MIN	0.1004
y_acc_MIN	0.0545
x_gyro_MEAN	0.0504
z_gyro_MEAN	0.0357
z_gyro_MIN	0.0336
y_gyro_VARIANCE	0.0326
y_acc_VARIANCE	0.0298
z_acc_VARIANCE	0.0293
x_acc_MAX	0.0283
x_gyro_VARIANCE	0.0269
z_acc_MIN	0.0255
z_gyro_VARIANCE	0.0221
y_gyro_MIN	0.0175
z_acc_MAX	0.0138
x_acc_MEAN	0.0127
z_gyro_MAX	0.0103
z_acc_MEAN	0.0095
x_acc_VARIANCE	0.0095
y_gyro_MAX	0.0090
x_gyro_MIN	0.0081
x_gyro_MAX	0.0052
y_gyro_MEAN	0.0041

Table 13.3: Feature relevance in identifying the correct class of activities

Then, we wanted to check if what Random Forests suggested made sense. So, we took the two features with highest relevance that this algorithm returned, i.e., the mean and the maximum accelerations computed on the Y axis. In Figure 13.4 we show the scatter diagram drawn from these two features. Each orange dot is an activity labeled as *Not Fall*, while each blue cross is an activity labeled as *Fall*. We can see that the *Not Fall* activities has a very negative mean acceleration and a much lower maximum acceleration than the *Fall* ones. As a consequence, we can conclude that Random Forests actually returned a correct result rating these two features as the most relevant ones. Indeed, it is easy to distinguish falls from not falls with their combination.

Detecting a classification approach to apply on the available dataset. After having built a dataset for the training task of our Machine Learning campaign, we proceeded in our research with the definition of the classification approach to be natively implemented in the Machine Learning Core of LSM6DSOX, i.e., the sensor at the base of our device. Firstly, we verified if one (or more) of the existing classification algorithms, already proposed, tested, verified and accepted by the scientific community, obtained satisfactory results in our specific scenario. If that was con-

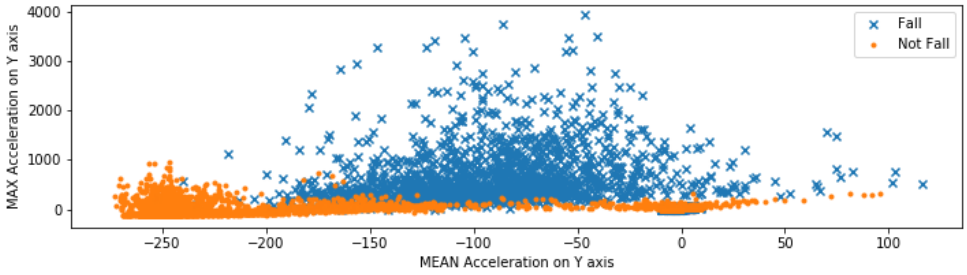


Fig. 13.4: Activities labeled as *Not Fall* and *Fall* against the mean and the maximum accelerations on the Y axis

firmed we could adopt an already accepted approach, instead of defining a new one. Indeed, this second case would have required an ad-hoc experimental campaign in our context, the publication in a journal and the consequent evaluation and possible adoption by research groups all over the world, in order to find possible weaknesses that could have been overlooked during our campaign.

In order to evaluate the existing classification algorithms, we decided to apply the classical measures adopted in the literature, i.e., Accuracy, Sensitivity and Specificity. If we indicate by: *(i)* TP the number of true positives, *(ii)* TN the number of true negatives, *(iii)* FP the number of false positives, and *(iv)* FN the number of false negatives, these three measures can be defined as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Sensitivity = \frac{TP}{TP + FN}$$

$$Specificity = \frac{TN}{TN + FP}$$

Accuracy is the number of correct forecasts on the total input size, and represents the overall performance of the algorithm. Sensitivity denotes the fraction of positive samples that are correctly identified. In our scenario, it is the fraction of *Fall Activities* that are properly identified by the algorithms. Finally, Specificity is the fraction of negative samples correctly identified, so it represents the fraction of *Not Fall Activities* properly identified by the algorithms. Accuracy, Sensitivity and Specificity are expressed by a number within the real interval $[0, 1]$; the higher the value, the better the algorithm. However, in principle, it may happen that a high accuracy is achieved with an unacceptable time performance. Therefore, in addition to accuracy, we have considered a measure of performance. For this purpose, we have chosen the worst case time complexity of Training and Prediction activities.

In Table 13.4, we report a summary of all the classification algorithms tested by us. In particular, we report Accuracy, Sensitivity and Specificity, obtained through a 10-Fold Cross Validation, on the left, and some measures of Performance (i.e., the worst case time complexity for Training and Prediction activities) on the right. Worst case time complexity is expressed in Big O notation, where n is the number of training samples, p is the number of features, n_{sv} is the number of support vectors, n_{l_i} is the number of neurons of layer i, m is the minimum between n and p.

	Accuracy	*Sensitivity*	*Specificity*	*Worst Case Time Complexity of Training*	*Worst Case Time Complexity of Prediction*
Decision Tree - C4.5	0.9487	0.9391	0.9566	$O(n^2p)$	$O(p)$
Decision Tree - CART	0.9128	0.8910	0.9223	$O(n^2p)$	$O(p)$
Multilayer Perceptron	0.9270	0.8829	0.9363	$O(n_{l_0}^2)$	$O(pn_{l_1} + n_{l_1}n_{l_2} + ...)$
k-Nearest Neighbors (k=3)	0.8790	0.8747	0.9263	$O(knp)$	$O(np)$
Logistic Regression	0.7707	0.8599	0.7057	$O(np)$	$O(p)$
Linear Discriminant Analysis	0.7557	0.4956	0.9663	$O(npm + m^3)$	$O(np + nm + pm)$
Gaussian Naive Bayes	0.7175	0.4947	0.8989	$O(np)$	$O(p)$
Support Vector Machine	0.7141	0.4103	0.9486	$O(n^2p + n^3)$	$O(n_{sv}p)$

Table 13.4: Accuracy, Sensitivity, Specificity values achieved by several classification algorithms when applied to our dataset (at left) and Worst Case Time Complexity of Training and Prediction (at right)

A metric can be more important than another one, depending on the application scenario. In ours, i.e., detecting falls in a work environment, Sensitivity is more relevant than Specificity. Indeed, a missed alarm (corresponding to a *Not Fall Activity* prediction of a *Fall Activity*) means no assistance to the worker. On the other hand, a false alarm can be confirmed as such by the worker interacting with the device.

Looking at Table 13.4, we can see that the Decision Tree - C4.5 is the Machine Learning model with the highest Sensitivity and the highest Accuracy. Also the Specificity of this model is excellent. Actually, the Linear Discriminant Analysis achieved a Specificity of 0.9663, higher than the one of the Decision Tree - C4.5. However, it obtained a very low value of Sensitivity and a low Accuracy.

As for the worst case time complexity, we can observe that, for the Training activity, there are important differences between the various approaches. For example, k-Nearest Neighbors and Logistic Regression have a better worst case time complexity than Decision Tree. However, the Training activity is performed only once, when the device is adopted; it might be repeated during the device life, but it is still very rare. Actually, the most important activity, in terms of worst case time complexity is Prediction, which occurs continuously. As for this activity, the various approaches have very similar performances and, in any case, Decision Trees shows the best one.

Given all the classification algorithms of Table 13.4 and the previous reasoning, we found that the best algorithm for our scenario was Decision Tree - C4.5. Furthermore, the performances were so good that we could adopt it for our case, without thinking a new ad-hoc classification algorithm, which performances would hardly be the same and would be affected by all the problems mentioned at the beginning of this section.

In addition to Decision Tree C4.5, we decided to implement two auxiliary algorithms in our Personal Device. These, using accelerometric and gyroscopic data, evaluate the intensity of the movement and the position of the device to confirm or not a fall detected through C4.5. In fact, when a potential fall is reported as a result of C4.5 processing, the Personal Device verifies its own intensity of movement and, through it, can determine whether the operator wearing it is moving. If this intensity is zero or very low, the operator is most likely on the ground, without the capability of moving. In this case, the Personal Device uses the 6DoF (i.e., Six Degree of Freedom), provided by the two sensors embedded in it, to evaluate the position of the operator. If this is compatible with a fall (i.e., it is a supine or prone position), it concludes that there has been a fall and triggers the alarm.

Hardware framework of our wearable device. Building the hardware framework of our device was difficult. Indeed, the device needed to implement our approach had to comply with some requirements. First, as said before, it had to be small and ergonomic, because it had to be worn by a person. Second, it should have an Inertial Measurement Unit (IMU), containing an accelerometer and a gyroscope. This was not sufficient because it also needed a Bluetooth module, able to manage the Bluetooth Low Energy (BLE) protocol. A device compliant with all these requirements is SensorTile.box provided by STMicroelectronics. It is shown in Figure 13.5.

Fig. 13.5: SensorTile.box (STEVAL-MKSBOX1V1)

SensorTile.box was designed to support the development of wearable IoT devices. It has a BLE v4.2 module and an ultra-low-power microcontroller STM32L4R9 that manages the following sensors:

- STTS751, a high precision temperature sensor;
- LSM6DSOX, a six-axis IMU and Machine Learning Core (MLC);
- LIS3DHH and LIS2DW12, three-axis accelerometers;
- LIS2MDL, a magnetometer;
- LPS22HH, a pressure sensor;
- MP23ABS1, an analogic microphone;
- HTS221, a humidity sensor.

In the current version of our device, the only sensor we used is LSM6DSOX. However, we do not exclude that we will employ one or more of the other sensors in the future.

LSM6DSOX has everything our approach needs; it is a system-in-package (SIP) that contains a three-axis high precision accelerometer and a gyroscope. The really important feature of this sensor is the Machine Learning Core (MLC) component, in addition to its low power consumption and its small size. Thanks to this, we are able to implement Artificial Intelligence algorithms directly in the sensor, without the need for a processor. MLC exploits the data provided by the accelerometer, the gyroscope and some possible external sensors to compute some statistical parameters (such as mean, variance, maximum and minimum values, etc.) in a specific sliding time window. All these parameters can be the input of a classification algorithm (the decision tree in our case) previously loaded by the user. Figure 13.6 reports the whole workflow of MLC.

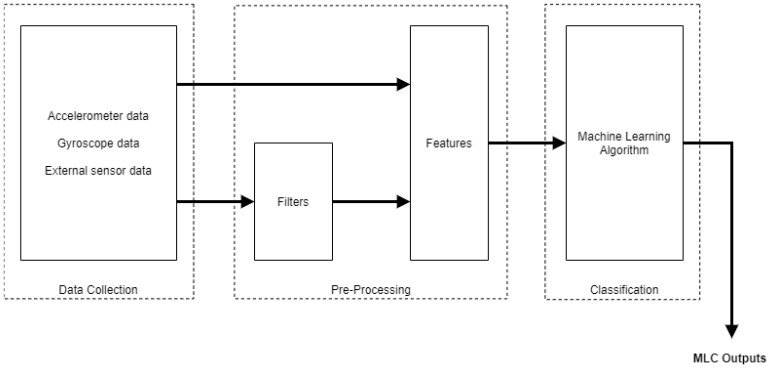


Fig. 13.6: Workflow of the Machine Learning Core of LSM6DSOX

In this figure, we specify some filters that can be applied to provided data. Examples of them are a low-pass filter, a bandwidth filter, a First-Order IIR and a Second-

Order IIR. This feature was very important for our approach because it lets us implement the Butterworth filter, applied on the data provided by the accelerometer and the gyroscope to reduce noise (see Section 13.2.1).

Embedding the defined logics in our device. In order to embed the defined logics in our device, we had to develop a firmware accepted by SensorTile.box. It had to be written in the C language, with all the instructions needed for the initialization of the micro-controller and the configuration of the Machine Learning Core. In order to do this, STMicroelectronics provides two software tools (i.e., STM32CubeMX[4] and STM32CubeIDE[5]) allowing users to develop C code for the microcontroller STM32L4R9. STM32CubeMX is a graphic tool to initialize the microcontroller peripherals, such as GPIO and USART, as well as its middlewares, like USB or TCP/IP protocols. The second software is an IDE allowing users to write, debug and compile the firmware of the microcontroller.

Our firmware has three essential functions, namely:

- `HAL_init()`, which starts the Hardware Abstraction Layer; this is a set of APIs above the hardware allowing the developer to interact with the hardware components safely.
- `Bluetooth_init()`, which initializes the Bluetooth stack. This operation includes the setting of the MAC address, the configuration of the HCI interface, the GAP and GATT protocols, and so forth.
- `MLC_init()`, which initializes the MLC component of LSM6DOX and enables the interruption of the output of decision trees. The MLC initialization is performed through the loading of a specific header file that configures all the registers of LSM6DOX. We describe this file below.

Configuring the MLC is not easy, because it also involves the configuration of the LSM6DSOX sensors and the settings of all its registers. This task is possible thanks to "Unico"[6], a cross-platform Graphical User Interface developed by STMicroelectronics. Unico allows a quick and easy setup of the sensors, as well as the complete configuration of all the registers. It also provides the user with advanced features (such as the Machine Learning Core, a Finite State Machine, a pedometer, etc.) embedded in the digital output devices.

Thanks to Unico it is possible to configure all the parameters of MLC and LSM6DSOX sensors, like the output frequency of MLC, the full scale of the ac-

[4] `https://www.st.com/en/development-tools/stm32cubemx.html`
[5] `https://www.st.com/en/development-tools/stm32cubeide.html`
[6] `https://www.st.com/en/embedded-software/unico-gui.html`

celerometer and gyroscope, the sample window of reference for the computation of features, and so on. Our complete configuration is shown in Table 13.5.

	Setting
Input data	Three axis accelerometer and gyroscope
MLC output frequency	12.5 Hz
Accelerometer sampling frequency	12.5 Hz
Gyroscope sampling frequency	12.5 Hz
Full scale accelerometer	±8 g
Full scale gyroscope	±2000 dps
Sample window	37 samples
Filtering	Second-Order IIR filter with cutting frequency at 4 Hz

Table 13.5: Adopted configuration of the MLC component

With these settings, the output of the classification algorithm is written into a dedicated memory register at each clock of MLC, so it is possible to read the result. If this last is set to *Fall* (i.e., the worker wearing it has presumably fallen) the alarm is activated.

Figure 13.7 shows a possible workplace scenario, on the top, and how the verification of a fall and the transmission of the corresponding alarm occur, on the bottom. More specifically, each device continuously checks its status in order to trigger the alarm when needed. Whenever the MLC component of the device worn by a user detects a fall, it sends a broadcast alarm message. All the other Personal and Area Devices in the signal range receive the alarm. If there are Personal Devices in the same area, the corresponding workers are alerted to go to see what happened. In any case, the alarm reaches the Area Device that alerts the Safety Coordination Platform. This last examines the received data and, if the fall is confirmed, triggers the alarm, activates a rescue plan and sends the suitable advices to all the people involved in this plan.

As said before, all communications of our wearable device are carried out through the Bluetooth Low Energy (BLE) protocol. Our device has two roles, Central and Peripheral. The BLE protocol allows our device to switch from a role to another at runtime. When the device is running normal, it listens to any other device; this role is the Central one. When the worker falls and the MLC component detects this fall, the device switches to the Peripheral role, triggering the alarm activation and sending all the corresponding data.

Device testing. Once the logic of our approach was deployed in the SensorTile.box, we started the testing phase of our device. In particular, we asked 30 volunteers (15 males and 15 females of different age and weight) to perform different kinds

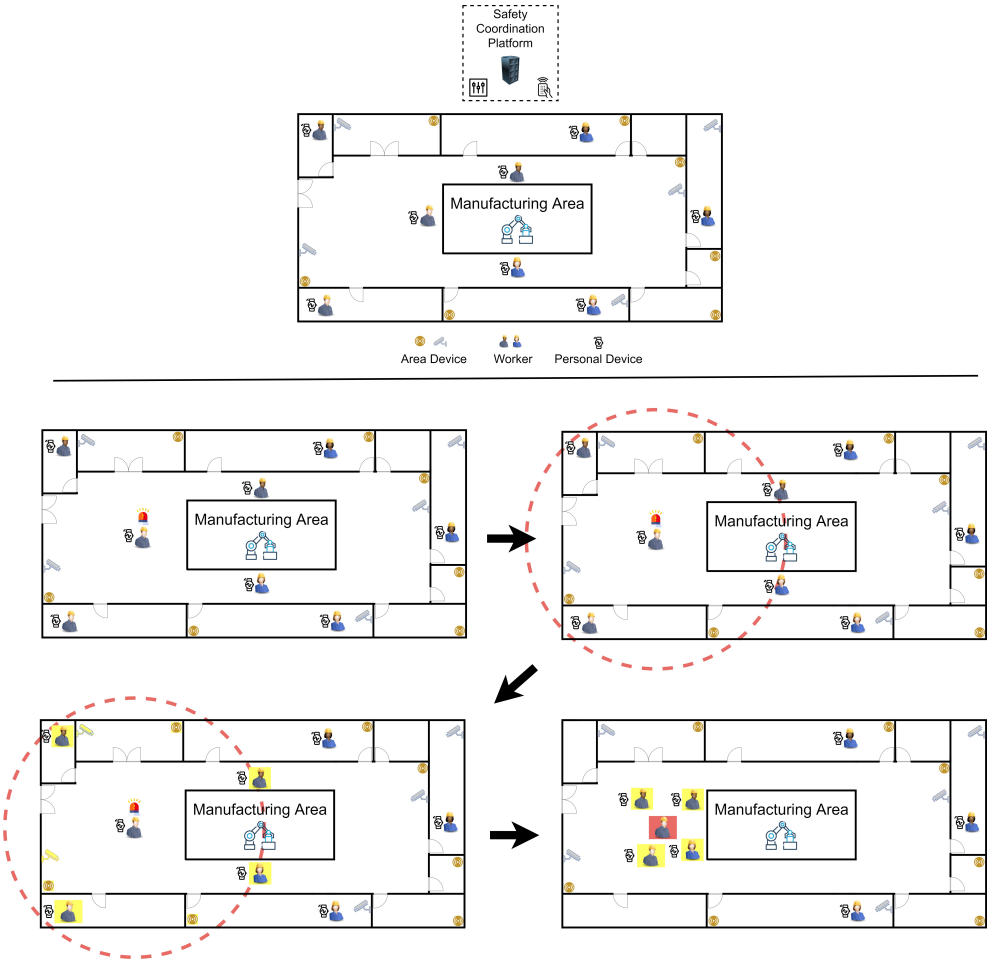


Fig. 13.7: Example of a workplace scenario (on the top) and description of how the verification of a fall and the transmission of the alarms occur (on the bottom)

of activities. The activities considered, reported in Table 13.6, include all the ones mentioned in the past literature. These could be grouped in *Fall Activities* and *Not Fall Activities*. Each time an activity was performed, the volunteer worn the device on the waist.

Table 13.7 shows the confusion matrix obtained for the output provided by our device. Looking at this table, we can see that the real number of *Fall Activities* was 1,205. Our device correctly identified 1,170 of them, while 35 were false negatives. *Not Fall Activities* classified as such were 595. Our device recognized 540 of them, but triggered 55 false alarms. As we said before, Sensitivity is much more important than Specificity in this application context. Indeed, we obtained a higher number of real *Fall Activities* than the one of real *Not Fall Activities*. At the end, we have a Sensitivity value equal to 0.97; Specificity is equal to 0.91. Finally, Accuracy is equal to 0.95.

Not Fall Activity	*Fall Activity*
Walk slow ($< 6km/h$)	Walk and fall forward after tripping
Walk fast ($\geq 6km/h$)	Walk and fall sideways (right) after tripping
Run slow ($< 8km/h$)	Walk and fall to the side (left) after tripping
Run fast ($\geq 8km/h$)	Fake fainting and fall on the right while standing
Sit slowly in a chair	Fake fainting and fall forward while standing
Sit slowly on the ground	Fake fainting and fall on the left while standing
Sit abruptly in a chair	Run and fall forward after stumbling
Jump to reach an object located at the top	
Go up and down the stairs slowly ($< 6km/h$)	
Go up and down the stairs quickly ($\geq 6km/h$)	
Walk and stumble without falling down	
Jump forward from an elevated position	
Jump forward from the floor	

Table 13.6: A taxonomy for *Not Fall Activities* (on the left) and *Fall Activities* (on the right)

	(Real) Fall	*(Real) Not Fall*
(Predicted) Fall	1170 (TP)	55 (FP)
(Predicted) Not Fall	35 (FN)	540 (TN)

Table 13.7: Confusion matrix for the output provided by our device

The training and testing phases of our device show very satisfying performances. In our opinion, the training dataset, which we built starting from some existing datasets, was fundamental to obtain such successful results, because we were able to construct a general model from heterogeneous activities. Indeed, our model can distinguish between sport activities and falls, a difficult goal to achieve. Sensitivity is very high and that is the most important parameter to evaluate in our context. Specificity is not particularly high, which can lead to some false alarms. In most cases, these can be directly stopped by the other two auxiliary algorithms embedded in our device (see Section 13.2.1) or, ultimately, by the worker wearing the device. On the other hand, considering that our reference scenario is a working environment, activities like running or jumping are common. These could lead to many false alarms if the model would not be sufficiently generalized to handle them, fully or partially.

13.2.2 Area Devices for fall detection

Area Devices represent the second level of our fall detection framework. They aim at monitoring a certain area of the working environment to check if one or more operators have fallen into it. In Section **??**, we have seen that there are three types of fall detectors, i.e., ambient, video and wearable detectors. While wearable detectors

belong to the Personal Devices category seen in the previous section, ambient and video detectors correspond to the Area Devices category.

An example of a fall detection approach using ambient detectors is proposed in [690]. It is based on the exploitation of a far-field microphone to record the audio of a given zone and classify possible falls. Instead, an example of a fall detection approach using video detectors is shown in [469]. It employs a single-camera system to detect a very large motion with a direction less than 180°. However, these are only two of the many approaches that can be adopted for fall detection at the Area Devices level.

The big advantage of this kind of fall detector is the ability to evaluate several people in the same area at the same time, unlike Personal Devices that are able to evaluate only the person wearing them. Actually, Personal Devices and Area Devices can be leveraged as mutual validators. In fact, as we will see in the next section, Area Devices receive data about falls from Personal Devices and, by cross-referencing this data with the ones detected by them, they are able to support the Safety Coordination Platform of our framework in the detection of possible falls, in the activation of alarms, and in the management of rescue activities. In case a Personal Device and the corresponding Area Devices are in conflict (for example because the Personal Device reports a fall while an Area Device does not recognize it), our framework always chooses the most pessimistic case (i.e., it reports a fall). This is justified by the fact that, in our reference scenario, a false alarm is much more tolerable than a missed one. An alarm notification activates the procedure described in Figure 13.7. If the alarm is triggered the operator is rescued; in the opposite case, the presence of a false alarm is reported.

13.2.3 Safety Coordination Platform for fall detection

The Safety Coordination Platform monitors the working environment and, in case of operator falls, raises an alarm and coordinates rescue activities. A fundamental tool within the Safety Coordination Platform is a map that represents the working environment divided into its areas. The map shows all the Area Devices that, communicating directly with the platform, send useful data for monitoring falls. These data are retrieved from the sensors inside the Area Devices and from the Personal Devices that communicate them to the platform through the Area Devices. Once data arrive to the platform, the latter proceeds with several elaborations on it to extract knowledge about the presence or absence of falls. In case of a possible fall, the platform triggers the alarms and the corresponding rescue operations, taking into account the cause(s) that provoked the fall and the gravity of the latter.

The Safety Coordination Platform consists of a chain of four modules (Figure 13.8):

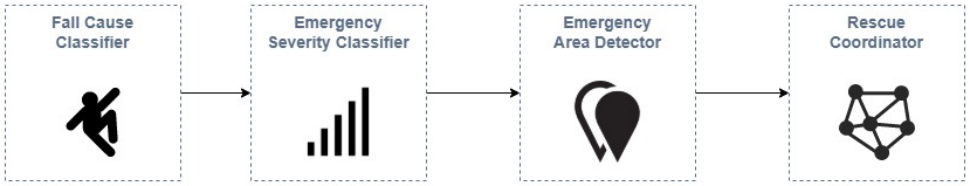


Fig. 13.8: Modules of the Safety Coordination Platform

- The *Fall Cause Classifier* aims at identifying the causes leading to a fall (e.g., slipping, fainting due to gas leaks, rushed escape due to fire or earthquake, etc.). For this purpose, it receives data from Area Devices and Personal Devices (through Area Devices) and passes it as input to one or more classification algorithms already proposed in past literature (e.g., Decision Tree, Support Vector Machine, Neural Networks, etc. [294]).
- The *Emergency Severity Classifier* examines the available data to identify the severity level of the emergency. This level is an integer between 1 and 5; the lower the value, the lower the severity is. The severity of an emergency depends on the type of a fall (for example, a slip is potentially less severe than a fall from the third floor), the cause of the fall (for example, a slip is potentially less severe than a fire) and the number of people involved. Also this classifier applies the data received from Area Devices to one or more classification algorithms already proposed in past literature.
- The *Emergency Area Detector* examines the available data to identify the area(s) involved in the emergency. For this purpose, it activates a clustering algorithm that groups the Area Devices and Personal Devices into those directly involved in the emergency, those indirectly involved in it (because, for example, they are involved in rescue activities), and those not involved in any way. The clustering algorithm determines the clusters taking into account the information related to the location of the various devices, as well as the type and level of the emergency (previously determined by the Fall Cause Classifier and the Emergency Severity Classifier).
- The *Rescue Coordinator* receives information on the cause of the fall, the severity of the emergency and the areas involved and, based on this information, it triggers the appropriate alarms. Next, it defines a rescue management plan (which may involve a rapid evacuation, a controlled evacuation, a simple first aid linked to a broken leg, etc.), providing each rescuer with the appropriate instructions.

These are sent to the Area Devices, which, in turn, send, in broadcast mode, the advices to all the operators, who are working in the area (for example, requiring the immediate evacuation of the area, in case of gas leakage). Furthermore, Area Devices send, in broadcast mode, the advices for each Personal Device to be transmitted to the corresponding operator (for example, specifying the fastest way for her/him to reach the injured colleague to give first aid). Each Personal Device, thanks to the use of the appropriate actuators, provides the worker who wears it with the appropriate instructions on what to do and how doing it.

14

Anomaly detection and classification in Multiple IoT scenarios

In this chapter, we report an attempt to investigate anomalies in a MIoT scenario. First, we propose a new methodological framework and three orthogonal taxonomies, in which each combination of the latter defines a specific type of anomaly to study. Then, in the context of anomaly detection in a MIoT, we define the so-called "forward problem" and "inverse problem". The definition of these problems allows the investigation of how anomalies depend on inter-node distances, the size of IoT networks, and the degree centrality and closeness centrality of anomalous nodes. The proposed approach is applied to a smart city scenario, which is a typical MIoT. Here, data coming from sensors and social networks can boost smart lighting in order to provide citizens with a smart and safe environment.

The material presented in this chapter was derived from [232].

14.1 Methods

14.1.1 Extending the MIoT paradigm

In this section, we extend the MIoT paradigm introduced in Chapter **??** in order to make it capable of representing and handling anomalies.

Given a MIoT $\mathcal{M} = \{\mathcal{I}_1, \mathcal{I}_2, \cdots, \mathcal{I}_m\}$, and pair of instances ι_{j_k} of o_j and ι_{q_k} of o_q in $\mathcal{I}_k$, the MIoT saves the set TrS_{jq_k} of the transactions from ι_{j_k} to ι_{q_k}. It is defined as:

$$TrS_{jq_k} = \{Tr_{jq_{k_1}}, Tr_{jq_{k_2}}, \cdots, Tr_{jq_{k_v}}\} \tag{14.1}$$

A transaction $Tr_{jq_{k_z}} \in TrS_{jq_k}$ is represented as follows:

$$Tr_{jq_{k_z}} = \langle st_{jq_{k_z}}, fh_{jq_{k_z}}, ok_{jq_{k_z}}, ct_{jq_{k_z}} \rangle \tag{14.2}$$

Here:

- $st_{jq_{k_z}}$ denotes the starting timestamp of $Tr_{jq_{k_z}}$.
- $fh_{jq_{k_z}}$ indicates the ending timestamp of $Tr_{jq_{k_z}}$.

- $ok_{jq_{k_z}}$ denotes whether $Tr_{jq_{k_z}}$ was successful or not; it is set to `true` in the affirmative case, to `false` in the negative one, and to `NULL` if it is still in progress.
- $ct_{jq_{k_z}}$ indicates the set of the content topics considered by $Tr_{jq_{k_z}}$. Specifically, it consists of a set of w keywords:

$$ct_{jq_{k_z}} = \{kw^1_{jq_{k_z}}, kw^2_{jq_{k_z}}, \ldots, kw^w_{jq_{k_z}}\} \tag{14.3}$$

An important subset of TrS_{jq_k} is $TrOkS_{jq_k}$, which stores the successful transactions of TrS_{jq_k}. It is defined as:

$$TrOkS_{jq_k} = \{Tr_{jq_{k_z}} | Tr_{jq_{k_z}} \in TrS_{jq_k}, ok_{jq_{k_z}} = \texttt{true}\} \tag{14.4}$$

In other words, this set comprises all the transactions through which ι_{q_k} gave a positive answer to a request of ι_{j_k}, thus providing this last one with services, information or data it required.

Now, we can define the set TrS_{j_k} of the transactions activated by ι_{j_k} in $\mathcal{I}_k$. Specifically, let $\iota_{1_k}, \iota_{2_k}, \cdots, \iota_{w_k}$ be all the instances belonging to $\mathcal{I}_k$. Then:

$$TrS_{j_k} = \bigcup_{q=1..w, q \neq j} TrS_{jq_k} \tag{14.5}$$

This means that the set TrS_{j_k} of the transactions of an instance ι_{j_k} is given by the union of the sets of the transactions from ι_{j_k} to all the other instances of $\mathcal{I}_k$.

We should note that, herein, we have reported only those aspects of the MIoT paradigm that are strictly necessary for our aim. The interested reader can find further details in [53].

We can now introduce the concept of neighborhood of an instance ι_{j_k} in $\mathcal{I}_k$. Specifically, the neighborhood Nbh_{j_k} of ι_{j_k} is defined as:

$$Nbh_{j_k} = ONbh_{j_k} \cup INbh_{j_k} \tag{14.6}$$

where:

$$\begin{aligned} ONbh_{j_k} &= \{n_{q_k} | (n_{j_k}, n_{q_k}) \in A_I, |TrS_{jq_k}| > 0\} \\ INbh_{j_k} &= \{n_{q_k} | (n_{q_k}, n_{j_k}) \in A_I, |TrS_{qj_k}| > 0\} \end{aligned} \tag{14.7}$$

In other words, Nbh_{j_k} comprises those instances directly connected to ι_{j_k} through an incoming or an outgoing arc, which shared at least one transaction with it.

Finally, we can define the concept of neighborhood of an i-arc $a_{jq_k} = (n_{j_k}, n_{q_k}) \in A_I$. Specifically, the neighborhood Nbh_{jq_k} of the i-arc a_{jq_k} is defined as:

$$Nbh_{jq_k} = ONbh_{jq_k} \cup INbh_{jq_k} \tag{14.8}$$

where:

$$\begin{aligned} ONbh_{jq_k} &= \{(n_{q_k}, n_{r_k}) | (n_{q_k}, n_{r_k}) \in A_I\} \\ INbh_{jq_k} &= \{(n_{l_k}, n_{j_k}) | (n_{l_k}, n_{j_k}) \in A_I\} \end{aligned} \tag{14.9}$$

Hence, $ONbh_{jq_k}$ contains all the arcs of A_I having n_{q_k} as source node, whereas $INbh_{jq_k}$ comprises all the arcs of A_I having n_{j_k} as target node.

14.1.2 Modeling anomalies in a MIoT

In this section, we propose a model allowing for the representation and management of anomalies in MIoTs. The core of our model consists of some possible taxonomies characterizing anomalies in this scenario. Each one will correspond to different analysis viewpoints. Borrowing a terminology typical in data analysis, these taxonomies can be seen as different dimensions of a multi-dimensional model, through which the fact "anomalies in a MIoT" can be investigated. Here, we consider three of these taxonomies, namely: *(i)* presence anomalies vs success anomalies; *(ii)* hard anomalies vs soft anomalies; *(iii)* contact anomalies vs content anomalies. However, we do not exclude that other taxonomies may also be possible in future works.

Continuing with the analogy between our taxonomies and the dimensions of a multi-dimensional model, we have that each combination of the possible values of these dimensions gives rise to a specific type of anomaly to study. Therefore, we have the *Presence-Hard-Contact Anomalies*, the *Success-Hard-Content Anomalies*, and so on. In the following subsections, we briefly illustrate each taxonomy and, then, provide a formalization for some types of combined anomalies. We point out again that the description of our taxonomies is orthogonal to specific anomaly detection techniques. In order to keep the formalization as clear as possible, we will focus on a simple anomaly detection scheme based on frequencies. However, more complex detection schemes may certainly be applied to our taxonomies.

Definition of anomaly taxonomies.

Presence Anomalies vs Success Anomalies. A *presence anomaly* denotes that there is a strong variation (i.e., *increase* or *decrease*) in the number of transactions carried out from an instance ι_{j_k} to an instance ι_{q_k} in a unit of time. A *success anomaly* shows that, although there is no presence anomaly from ι_{j_k} to ι_{q_k}, there is a strong *decrease* in the number of *successful* transactions from ι_{j_k} to ι_{q_k} in a unit of time.

Hard Anomalies vs Soft Anomalies. A *hard anomaly* indicates that the frequency of successful transactions carried out from an instance ι_{j_k} to an instance ι_{q_k} is higher than (or lower than) a certain threshold. A *soft anomaly* happens when the frequency of the (successful) transactions ranges between the maximum and the minimum

thresholds but, for several consecutive instances of time, it is higher (resp., lower) than the mean of these two thresholds and it shows a monotone increasing (resp., decreasing) trend. The rationale underlying this taxonomy is that hard anomalies are indicators of faults, whereas soft anomalies are indicators of a slow, but constant, degradation. Soft anomalies are extremely precious in applications such as predictive maintenance.

Contact Anomalies and Content Anomalies. A *contact anomaly* from an instance ι_{j_k} to an instance ι_{q_k} considers only the presence or the absence of transactions. By contrast, a *content anomaly* takes the content exchanged in the corresponding transactions into account[1]. Here, we assume that we are capable of identifying possible synonymies or homonymies relating terms. This is a well-known problem in the cooperative information system research field and several thesauruses have been proposed for this purpose. In this chapter, unless otherwise specified, we will refer to Babelnet [458], which is among the most advanced thesauruses. As far as content anomalies are concerned, a reference content set, consisting of some keywords, is necessary for verifying variations with respect to the content of the involved transactions. Two variants of content anomalies can be considered, namely: *(i)* the *strict* content anomalies, where the whole set of the reference keywords must be present in the involved transactions, and *(ii)* the *loose* content anomalies, where at least one of the reference keywords must be present therein.

Formalization of anomalies. The combination of the three taxonomies introduced above gives rise to eight possible kinds of anomaly. In the following, we provide the formal definition for representative cases. We recall that, for the sake of clarity, in these definitions we consider frequencies as the basic factor for anomaly detection. However, we point out that, even if frequencies are a well-accepted and widely adopted factor, even more complex factors could easily be incorporated into our taxonomies.

In the next subsections, we present a formalization of a representative selection of the eight anomaly types, providing the method for computing their anomaly degrees. We have not included the formalization for all cases, due to brevity. Yet, their definition would be analogous and straightforward.

The kinds of anomaly that we formalize below include: *(i)* Presence-Hard-Contact anomalies, *(ii)* Success-Hard-Contact anomalies, *(iii)* Presence-Soft-Contact anoma-

[1] Recall that, given a transaction $Tr_{jq_{k_z}}$, the corresponding content $ct_{jq_{k_z}}$ consists of a set of w keywords.

lies, and *(iv)* Presence-Hard-Content anomalies. In many of these definitions, the variable "time" plays a key role.

Presence-Hard-Contact Anomalies. Let t be a time instant and let Δt be a time interval (consisting of one or more time units). The frequency $TrFr_{jq_k}(t, \Delta t)$ of the transactions from ι_{j_k} to ι_{q_k} can be defined as follows:

$$TrF_{jq_k}(t, \Delta t) = \frac{|\{Tr_{jq_{k_z}} \mid Tr_{jq_{k_z}} \in TrS_{jq_k}, st_{jq_{k_z}} \geq t, fh_{jq_{k_z}} \leq (t + \Delta t)\}|}{\Delta t} \tag{14.10}$$

In other words, TrF_{jq_k} is given by the ratio between the number of transactions from ι_{j_k} to ι_{q_k} exchanged in the time interval $[t, t + \Delta t]$ to the length of this time interval (i.e., Δt).

We say that there is a Presence-Hard-Contact anomaly from ι_{j_k} to ι_{q_k} in the time interval $[t, t + \Delta t]$ if:

- TrF_{jq_k} is higher than a certain threshold th_{max}, in which case the anomaly degree is defined as $\alpha_{jq_k}(t, \Delta t) = \frac{TrF_{jq_k}(t,\Delta t) - th_{max}}{th_{max}}$, or
- TrF_{jq_k} is lower than a certain threshold th_{min} and this inequality does not hold in the time instants preceding t. This last condition is necessary to avoid that the lack of transactions from ι_{j_k} to ι_{q_k} is erroneously interpreted as a presence anomaly, as it would the case for instance when two instances have never performed transactions between them in the past. In this case, the anomaly degree is defined as $\alpha_{jq_k}(t, \Delta t) = \frac{th_{min} - TrF_{jq_k}(t,\Delta t)}{th_{min}}$.

If no Presence-Hard-Contact anomaly is detected, $\alpha_{jq_k}(t, \Delta t)$ is set to 0.

Here and in the following, the thresholds th_{max} and th_{min} can either be static or are dynamically computed over the previous observations. For instance, they could be computed considering both the mean and the standard deviation observed for TrF_{jq_k} in a predefined period of time. However, their actual definition depends on the application domain.

Presence-Hard-Contact anomalies focus on anomalies detected in the number of transactions (*presence*) occurring between two *instances* in a MIoT without considering the content they share (*contact*) and focusing on sharp variations of observed values (*hard*).

Their detection could be particularly relevant, for example, to identify faults concerning the ability of a MIoT object to send data. This may happen, for instance, because an object is no longer working.

Here and in the following, thanks to the concept of MIoT, anomalies between pairs of instances can be used to compute anomalies between the corresponding pairs of objects. In particular, given two objects o_j and o_q, let $\mathcal{IS}_{jq}$ be the set of IoTs

containing instances of both o_j and o_q connected by an *i-arc*. The anomaly degree $\alpha_{jq}(t, \Delta t)$ between the pair of objects o_j and o_q in a MIoT can be defined as:

$$\alpha_{jq}(t, \Delta t) = \frac{\sum_{\mathcal{I}_k \in \mathcal{IS}_{jq}} \alpha_{jq_k}(t, \Delta t)}{|\mathcal{IS}_{jq}|} \tag{14.11}$$

This way of computing anomalies between pairs of objects in a MIoT, starting from the anomalies of the corresponding pairs of instances, is valid for all kinds of anomalies.

Success-Hard-Contact Anomalies. Similarly to what we have done for Presence-Hard-Contact anomalies, we first define the frequency $TrOkF_{jq_k}(t, t + \Delta t)$ of the transactions from ι_{j_k} to ι_{q_k} that occurred successfully in the time interval $[t, t + \Delta t]$ as:

$$TrOkF_{jq_k}(t, \Delta t) = \frac{|\{Tr_{jq_{k_z}} \mid Tr_{jq_{k_z}} \in TrOkS_{jq_k}, st_{jq_{k_z}} \geq t, fh_{jq_{k_z}} \leq (t + \Delta t)\}|}{\Delta t} \tag{14.12}$$

Now, we can say that, in the time interval $[t, t + \Delta t]$, there is a Success-Hard-Contact anomaly if:

- there is no Presence-Hard-Contact anomaly in the same time interval;
- $TrOkF_{jq_k}$ is lower than a certain threshold th'_{min}.

In this case, the anomaly degree is defined as $\alpha_{jq_k}(t, \Delta t) = \frac{th'_{min} - TrOkF_{jq_k}(t, \Delta t)}{th'_{min}}$. Otherwise, $\alpha_{jq_k}(t, \Delta t) = 0$.

Success-Hard-Contact anomalies are very similar to Presence-Hard-Contact anomalies. However, they focus on the fraction of successful transactions occurring between two instances in a MIoT (*success*); they disregard the content exchanged by transactions (*contact*) and focus on sharp variations of observed values (*hard*).

The detection of this kind of anomaly might be particularly relevant, for example, in recognizing possible difficulties of a MIoT object to deliver requested data. Differently from the previous case, this may happen because there is an issue in the network rather than in the object itself.

Presence-Soft-Contact Anomalies. Let t be a time instant, let Δt be a time interval and let τ be a positive integer representing the number of time units after t into consideration (generally, $\tau \gg \Delta t$), and let $th_{avg} = \frac{th_{min} + th_{max}}{2}$. We can say that, in the time interval $[t, t + \tau]$, there is a Presence-Soft-Contact anomaly if, for each time instant θ such that $t \leq \theta \leq t + \tau$, the following conditions hold:

- $th_{min} \leq TrF_{jq_k}(\theta, \Delta t) \leq th_{max}$, which implies that no Presence-Hard-Contact anomaly exists in the time interval into consideration;

- $TrF_{jq_k}(\theta, \Delta t) > th_{avg}$ (resp., $TrF_{jq_k}(\theta, \Delta t) < th_{avg}$), which denotes that the frequency of the transactions from ι_{j_k} to ι_{q_k} is always higher (resp., smaller) than the average between th_{min} and th_{max};
- $TrF_{jq_k}(\theta + 1, \Delta t) \geq TrF_{jq_k}(\theta, \Delta t)$ (resp., $TrF_{jq_k}(\theta + 1, \Delta t) \leq TrF_{jq_k}(\theta, \Delta t)$), which implies that the frequency of the transactions from ι_{j_k} to ι_{q_k} is monotonically increasing (resp., decreasing) in the time interval Δt of interest.

If an anomaly is detected, the corresponding anomaly degree $\alpha_{jq_k}(t, \Delta t)$ is set to $\alpha_{jq_k}(t, \Delta t) = \frac{|TrF_{jq_k}(t+\tau, \Delta t) - th_{avg}|}{th_{avg}}$. Otherwise, $\alpha_{jq_k}(t, \Delta t) = 0$.

Presence-Soft-Contact anomalies focus on a smooth (*soft*) decrease in the number of all (*presence*) the transactions exchanged between two instances of a MIoT, without considering the exchanged content (*contact*).

The detection of this kind of anomaly may be useful in identifying a slowly but constantly changing behavior of an object. For instance, it could regard an object that is wearing out, an equipment whose battery has a very low charge level, and so forth.

Presence-Hard-Content Anomalies. Let $\overline{ct}$ be a content consisting of (presumably very few) keywords. We define the set $sTrCtS_{jq_k}(\overline{ct})$ of the transactions from ι_{j_k} to ι_{q_k} *strictly adherent* to $\overline{ct}$, i.e., the set of the transactions from ι_{j_k} to ι_{q_k} that contain *all the keywords* of $\overline{ct}$ as follows:

$$sTrCtS_{jq_k}(\overline{ct}) = \{Tr_{jq_{k_z}} \mid Tr_{jq_{k_z}} \in TrS_{jq_k}, \overline{ct} \subseteq ct_{jq_{k_z}}\} \tag{14.13}$$

As previously pointed out, here we assume that we are capable of identifying possible synonymies or homonymies relating a term of $\overline{ct}$ with a term of $ct_{jq_{k_z}}$. For this purpose, we use Babelnet [458].

Consider, now, a content $\overline{ct}$ consisting of some keywords. We define the set $lTrCtS_{jq_k}(\overline{ct})$ of the transactions from ι_{j_k} to ι_{q_k} that are *loosely adherent* to $\overline{ct}$, i.e., the set of the transactions from ι_{j_k} to ι_{q_k} that contain *at least one keyword* of $\overline{ct}$ as follows:

$$lTrCtS_{jq_k}(\overline{ct}) = \{Tr_{jq_{k_z}} \mid Tr_{jq_{k_z}} \in TrS_{jq_k}, (\overline{ct} \cap ct_{jq_{k_z}}) \neq \emptyset\} \tag{14.14}$$

Let t be a time instant and let Δt be a time interval. By applying the same approach described for Presence-Hard-Contact anomalies, it is possible to define the frequency $sTrCtF_{jq_k}(\overline{ct})$ (resp., $lTrCtF_{jq_k}(\overline{ct})$) of the transactions from ι_{j_k} to ι_{q_k} strictly (resp., loosely) adherent to $\overline{ct}$. Then, it is possible to state that, in the time interval $[t, t + \Delta t]$, there is a strict (resp., loose) Presence-Hard-Content anomaly from ι_{j_k} to ι_{q_k} against $\overline{ct}$ if:

- $sTrCtF_{jq_k}(\overline{ct})$ (resp., $lTrCtF_{jq_k}(\overline{ct})$) is higher than a certain threshold th_{max}, or
- $sTrCtF_{jq_k}(\overline{ct})$ (resp., $lTrCtF_{jq_k}(\overline{ct})$) is lower than a certain threshold th_{min} and this inequality does not hold in the time instants preceding t.

Analogously to what we have done for Presence-Hard-Contact anomalies, if the first condition is verified, the anomaly degree $\alpha_{jq_k}(t, \Delta t)$ can be defined as $\alpha_{jq_k}(t, \Delta t) = \frac{sTrCtF_{jq_k}(\overline{ct}) - th_{max}}{th_{max}}$, for strictly adherent anomalies, and $\alpha_{jq_k}(t, \Delta t) = \frac{lTrCtF_{jq_k}(\overline{ct}) - th_{max}}{th_{max}}$, for loosely adherent ones. Instead, if the second condition is verified, then $\alpha_{jq_k}(t, \Delta t) = \frac{th_{min} - sTrCtF_{jq_k}(\overline{ct})}{th_{min}}$, for strictly adherent anomalies, and $\alpha_{jq_k}(t, \Delta t) = \frac{th_{min} - lTrCtF_{jq_k}(\overline{ct})}{th_{min}}$ for loosely adherent ones. $\alpha_{jq_k}(t, \Delta t) = 0$ in all the other cases.

Presence-Hard-Content anomalies focus on sharp variations (*hard*) in the number of transactions (*presence*) exchanged between two instances in a MIoT, with regard to a certain set of contents (*content*).

The study of content variations paves the way to a wide variety of analyses, ranging from variations in the interests of a user who is adopting the MIoT objects, to variations in the sentiment of a user on a specific topic/service provided through the MIoT objects.

The other kinds of anomaly, whose formalization we have not reported in this chapter because they are very similar to the ones considered above, would provide four further viewpoints of the possible anomalies existing in a MIoT. It would be straightforward to see how these extra anomalies would allow us to model other possible real-world cases, which shows the generic applicability of our approach (three taxonomies and a multi-dimensional perspective).

14.1.3 Investigating the origins and effects of anomalies in a MIoT

After providing a multi-dimensional taxonomy of the possible anomalies present in a MIoT, in this section we aim at investigating their origins and effects. For this purpose, we address two problems that, according to what happens in several other research fields, we dubbed "forward problem" and "inverse problem", respectively. In the forward problem, given one or more anomalies, we aim at analyzing their effects on a MIoT. In the inverse problem, which is traditionally more complex than the forward one, given the effects of one or more anomalies on the nodes and the arcs of a MIoT, we aim at detecting the origin(s) of them, i.e., the node(s) or the arc(s) from which anomalies have started.

Forward Problem. As previously pointed out, this problem aims at understanding the effects that one or more anomalies have on the nodes of a MIoT. In the following, we will investigate the forward problem for one kind of anomaly, namely the

Presence-Hard-Contact anomaly. However, all our results can be extended to all the other cases introduced in Section 14.1.2.

First, given a node n_{j_k} of an IoT $\mathcal{I}_k$, along with the anomaly degrees of its outgoing arcs, in the forward problem we want to compute the overall effects of these anomalies over the corresponding IoT, $\mathcal{I}_k$. Specifically, the degree $\delta_{j_k}(t,\Delta t)$ of the anomalies of n_{j_k} in the time instant t and in the time interval Δt depends on the number of nodes belonging to $ONbh_{j_k}$ and, for each of these nodes n_{q_k}, on the degree $\delta_{q_k}(t,\Delta t)$ of the anomalies involving it and on the anomaly degrees measured for the corresponding arcs.

We wish to observe that, by saying that the degree of the anomalies of a node n_{j_k} recursively depends on the degree of the anomalies of the nodes belonging to $ONbh_{j_k}$, we introduce a way of proceeding that is similar to the one underlying the definition of the PageRank [484]. Thus, to compute δ_{j_k}, it is possible to adapt the formula for the computation of the PageRank to our scenario. Specifically:

$$\delta_{j_k}(t,\Delta t) = \gamma + (1-\gamma)\cdot\frac{\sum_{n_{q_k}\in ONbh_{j_k}}\delta_{q_k}(t,\Delta t)\cdot\alpha_{jq_k}(t,\Delta t)}{\sum_{n_{q_k}\in ONbh_{j_k}}\alpha_{jq_k}(t,\Delta t)} \tag{14.15}$$

This formula says that the degree $\delta_{j_k}(t,\Delta t)$ of the anomalies of n_{j_k} in the time instant t and in the time interval Δt is obtained by summing two components:

- The former component, γ, is the damping factor generally existing in each approach based on PageRank. It ranges in the real interval [0,1] and denotes the minimum absolute anomaly degree that can be assigned to a node of the MIoT.
- The second component, is a weighted sum of the anomaly degree $\delta_{q_k}(t,\Delta t)$ of the nodes n_{q_k} directly connected to n_{j_k} and, therefore, belonging to $ONbh_{j_k}$. The weight of each anomaly degree $\delta_{q_k}(t,\Delta t)$ is given by the value of the parameter α_{jq_k}, which considers the fraction of anomalous transactions performed from n_{j_k} to n_{q_k}.

In this formula, $\delta_{j_k}(t,\Delta t)$ ranges in the real interval [0,1].

The above formula allows us to determine the effects of a faulty node over the corresponding IoT, and consequently on the whole MIoT (as will become clearer next). However, we observe that the current formalization is valid only in the presence of a single faulty node. When multiple nodes simultaneously exhibit some anomalous behavior in one IoT (of the MIoT), our approach fails to distinguish among the contributions of each anomaly, particularly when the effects are measured in a single node. We wish to point out that this is our very first attempt to investigate MIoT anomalies, proposing a method to evaluate their effects. Our next priority as a follow-up of the present study, will be extending our method accordingly.

Having investigated the effects of an anomaly of an *instance* in an IoT, we can now exploit the features of the MIoT paradigm to analyze the effects of an anomaly of an *object* in a MIoT. In particular, the anomaly degree $\delta_j(t,\Delta t)$ of an object o_j can be computed starting from the anomaly degrees of its instances. Specifically, given the set $\mathcal{IS}_j$ of the IoT containing instances of o_j, $\delta_j(t,\Delta t)$ can be computed as:

$$\delta_j(t,\Delta t) = \frac{\sum_{\mathcal{I}_{j_k} \in \mathcal{IS}_j} \delta_{j_k}(t,\Delta t)}{|\mathcal{IS}_j|} \tag{14.16}$$

We observe that the value of $\delta_j(t,\Delta t)$, if compared with the one of $\delta_{j_k}(t,\Delta t)$, can provide very useful information. In particular, if $\delta_j(t,\Delta t)$ is very similar to $\delta_{j_k}(t,\Delta t)$ for each IoT $\mathcal{I}_{j_k} \in \mathcal{IS}_j$, we can conclude that o_j is really a source of anomaly. Instead, if the standard deviation of $\delta_j(t,\Delta t)$ is high, then we can conclude that o_j is involved in, or affected by, some anomalies in one or more IoTs, but not in some other ones.

Inverse Problem. As previously pointed out, the inverse problem is traditionally more complex than the forward one. For this reason, we will focus only on the simplest scenario, i.e., the case in which there is only one anomaly in the MIoT. In the future, we plan to extend our investigation to more complex scenarios. Let $a_{jq_k} = (n_{j_k}, n_{q_k})$ be an i-arc of a MIoT presenting an anomaly whose origin is not known. In the inverse problem we want to detect this origin.

First of all, we must verify if the origin of the anomaly is just a_{jq_k}. For this purpose, we consider the "siblings" of a_{jq_k}, i.e., the other arcs having n_{j_k} as the source node and the other arcs having n_{q_k} as the target node. If none of these present anomalies, then it is possible to conclude that a_{jq_k} is the origin of the observed anomaly and that this last one did not affect other nodes or arcs of the MIoT. In this case, the inverse problem has been solved and the investigation terminates.

However, the situation described above is very particular and, also, quite rare. More typically, anomalies tend to affect multiple nodes and arcs. In that case, given an anomaly found in an arc a_{jq_k}, in order to detect its origin, the first step consists in computing the anomaly degrees of n_{j_k} and n_{q_k} and to choose the maximum between the two. This becomes the current node under investigation.

At this point, an iterative process, aiming at finding the origin of the observed anomaly, is activated. During each step of this process, we apply the PageRank-based formula for the computation of the anomaly degree of a node, as discussed in Section 14.1.3, to all the nodes of the $ONbh$ and the $INbh$ of the current node. After this, we select the node having the maximum anomaly degree. If the degree of this node is higher than the one of the current node, it becomes the new current node and a new iteration starts. Otherwise, our approach concludes that the current node is the origin of the anomaly under consideration.

Clearly, the approach described above is greedy and, therefore, must be intended as a heuristic that could return a local maximum, instead of a global one. However, it is possible to apply to this approach all the techniques for improving the accuracy of a greedy approach already proposed in past literature, spanning from meta heuristics, such as hill climbing [531], to evolutionary optimization algorithms [572].

For instance, if the MIoT is not excessively large, it could be possible to compute the anomaly degree of all its nodes by applying the PageRank-based approach described in Section 14.1.3. In this case, the node having the maximum value of anomaly degree would be selected as the anomaly origin. This would correspond to applying an approach returning the optimum solution to the inverse problem, instead of one returning an approximate solution.

On the opposite extreme, if the network is very large, and the anomaly is affecting a vast portion of it, the greedy approach may be prohibitive. In this case, we will need to find an additional way to stop the iterative process, particularly when resources are limited and the process does not stop because, at each iteration, it continues to return a new current node with an anomaly degree higher than the one of the previous iteration. For instance, we could define a maximum number of iterations or a minimum increase of the anomaly degree necessary to activate a further iteration. Furthermore, this required minimum increase could be dynamic and could vary based on the number of steps already performed.

We conclude this section with an important consideration. Since this is our first effort to investigate the inverse problem, we had the necessity to limit our analysis to only one case, i.e., the one in which, in a certain time instant, there is only one anomaly in the MIoT. If at a given time instant, there are more anomalies in the MIoT, the search of the corresponding origins becomes much more complex, because the anomalies could interfere with each other. These interferences could make the search of the anomaly sources extremely complex.

For instance, we argue that, in presence of two anomalies whose source nodes are not known, in case these two nodes were relatively close to each other, the examination of the anomaly degree of their neighbors could be extremely beneficial. In fact, in this scenario, some of these neighbors are influenced only by one anomaly; other ones are influenced only by the other anomaly; a third group of neighbors is influenced by both anomalies; finally, a fourth group is not influenced by any anomalies. By deeply analyzing what happens in these four groups of nodes, it could be possible to derive precious information leading us to identify the sources of the two anomalies. In the future, we plan to conduct specific and accurate investigations about this case, and several other ones possibly characterizing the inverse problem.

14.2 Results

14.2.1 Testbed

To perform this analysis, we considered a reference scenario related to a smart city context. To model it, and to test our approach, we constructed a prototype. Furthermore, we realized a MIoT simulator.

In order to make "concrete" and "plausible" the simulated MIoT, our simulator needs to generate MIoTs having the characteristics specified by the user, whilst being as close as possible to real-world scenarios. In the simulator design, and in the construction of the MIoT used in the experiments, we followed the guidelines outlined in [283, 47, 48], where the authors highlight that one of the main factors used to build links in an IoT is node proximity.

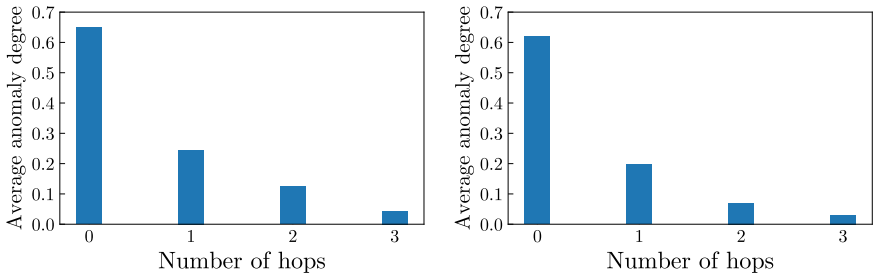
In order to reproduce the creation of transactions among objects, we decided to leverage information about a simulated smart city context. As for a dataset containing real-life paths in a smart city, we selected the one reported in `http://www.geolink.pt/ecmlpkdd2015-challenge/dataset.html`. This regards movements of objects, in terms of routes, in the city of Porto from July 1^{st} 2013 to June 30^{th} 2014. Each route contains several Points of Interest, corresponding to the GPS coordinates of each object as it moves in Porto. With this information at hand, our simulator associates an object (thus, creating a node) with one of the routes recorded in the dataset. Furthermore, it creates an arc between two nodes when the distance between the corresponding routes is less than a certain threshold th_d, for a predefined time interval th_t. The value of th_d and th_t can be specified through the constructor interface. Clearly, the higher is this value the more connected the constructed MIoT will be. When we defined the distribution of the transactions among the nodes, we leveraged scientific literature and used the corresponding results to properly tune our simulator. In particular, we adopted the values reported in [278].

The interested reader can find the MIoT created by our simulator for the experiments at the Web address `http://daisy.dii.univpm.it/miot/datasets/anomaly-detection`. It consists of 1,256 nodes and six IoTs having 128, 362, 224, 280, 98 and 164 nodes, respectively. The constructed MIoT is returned in a format that can be directly processed by the cypher-shell of Neo4J. Some statistics about our dataset are reported in Table 14.1.

We carried out all the tests presented in this section on a server equipped with an Intel I7 Quad Core 7700 HQ processor and 16 GB of RAM, with the Ubuntu 16.04 operating system. To implement our approach, we adopted Python, as programming language, and Neo4J (Version 3.4.5), as underlying DBMS.

Parameter	*Value*
Number of nodes	1,256
Number of relationships	6,860
Mean outdegree	5.44
Mean indegree	5.58

Table 14.1: Parameter values for our simulator



(a) (b)

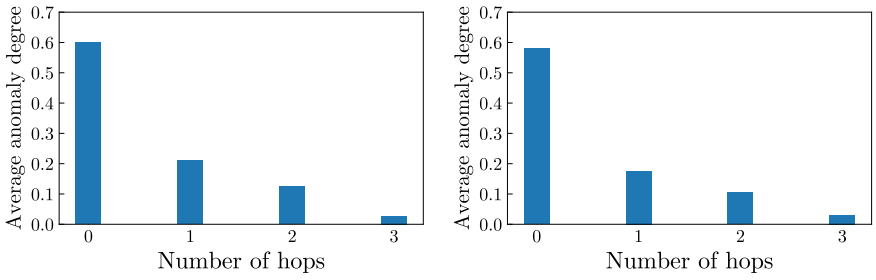
Fig. 14.1: Values of δ_{j_k} (corresponding to 0 hops) and average values of the anomaly degrees of all the nodes of $\mathcal{I}_k$ (on the left) and of the MIoT (on the right) being 1, 2 and 3 hops far from n_{j_k} in case of Presence-Hard-Contact anomalies

14.2.2 Analysis of the forward problem

Let us preliminarily define the concept of "number of hops" h_{jq_k} between the node n_{j_k} and another node n_{q_k} as the minimum number of arcs of the MIoT that must be traversed in order to reach n_{q_k} from n_{j_k}.

In a first step we analyzed the effects that the anomalous behavior of an object o_j had on the nodes of a MIoT. As pointed out in Sect. 14.1.3, given a node n_{j_k} of the IoT $\mathcal{I}_k$, its anomaly degree is represented by the parameter δ_{j_k}. This anomaly may propagate through the MIoT, thus affecting other nodes. To investigate this propagation, given an anomalous instance of an object o_j and the IoT $\mathcal{I}_k$, we measured the anomaly degree δ_{j_k} of n_{j_k} and the average of the anomaly degrees δ_{q_k} of all the nodes n_{q_k}, grouped by the number of hops from n_{j_k} to n_{q_k}. Moreover, we computed the same values but averaged through the IoT belonging to the MIoT. The same test has been run over 100 randomly chosen nodes, and results have been averaged over the runs.

Figure 14.1 shows the results obtained for Presence-Hard-Contact anomalies, while Figure 14.2 presents those regarding Presence-Soft-Contact anomalies. From the analysis of these figures it is possible to observe that the effects of an anomaly on a node spread over the surrounding nodes, even if they rapidly decrease against the



(a) (b)

Fig. 14.2: Values of δ_{j_k} (corresponding to 0 hops) and average values of the anomaly degrees of all the nodes of $\mathcal{I}_k$ (on the left) and of the MIoT (on the right) being 1, 2 and 3 hops far from n_{j_k} in case of Presence-Soft-Contact anomalies

number of hops. The corresponding trend follows a power law distribution. If we compare the left and the right distributions of Figures 14.1 and 14.2, we can observe that anomalies propagate more slowly on a MIoT than on a single IoT. However, this difference is negligible. Furthermore, there are no significant differences between Presence-Hard-Contact anomalies and Presence-Soft-Contact anomalies, except that the latter ones are slightly smaller than the former ones. This trend can be justified by considering that Presence-Soft-Contact anomalies are more difficult to be observed than Presence-Hard-Contact ones, since the former ones are not only required to show values higher (resp., lower) than a given threshold, but should also exhibit a trend that is monotonically increasing (resp., decreasing), within the time interval of interest. As the trends are very similar, in the following tests we focus only on Presence-Hard-Contact anomalies, without loss of generality.

Next, we investigated the effects that the anomaly of an object has on the other objects connected to it. In particular, given an object o_q, whose instances belong to the $ONbh$ of the instances of an anomalous object o_j in at least one IoT of the MIoT, we computed the value and the standard deviation[2] of δ_j and δ_q. We repeated this task 100 times with different pairs of objects o_j and o_q. Then, we averaged the values obtained over the runs. The corresponding results are shown in Figure 14.3, under the category ALL. As we can observe, the standard deviation of δ_j is very low. This result can be explained by the fact that all the instances of the anomalous object o_j present anomalies and, consequently, the corresponding anomaly degrees are almost uniform. By contrast, the value of δ_q is lower than the one of δ_j, exhibiting a very high standard deviation. This is explained by observing that the instances of o_q are not in the neighborhoods of the instances of o_j in all the IoTs of the MIoT. In fact, in

[2] Recall that δ_j and δ_q are computed by averaging the anomaly degrees of all the instances of o_j and o_q.

some of them, they can be 2, 3 or more hops away from the instances of o_j. In some cases, they may even be disconnected from the instances of o_j.

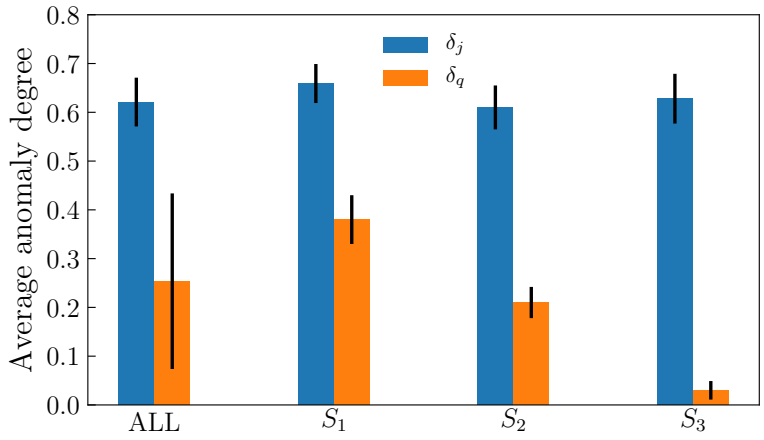


Fig. 14.3: Anomaly degrees and the corresponding standard deviations in different scenarios

As a next step, we repeated the previous experiment, enforcing some extra constraints, which defined three different scenarios. In the first (resp., second, third) one, all the instances of o_q were 1 (resp., 2, more than 2) hop(s) far from the instances of o_j; the third scenario includes also instances of o_q not connected to instances of o_j. The results obtained are shown in Figure 14.3 under the labels S_1, S_2 and S_3, respectively. Looking at the data labelled as ALL, these results are coherent with both the ones of Figure 14.1 and the ones of Figure 14.3. We can see that the effects of a single anomaly are rapidly reduced as soon as we move away from its origin. Furthermore, this experiment confirms what we pointed out in Section 14.1.3, i.e., that the anomaly degree δ is a parameter that really helps detecting the object that has caused the anomaly in the first place.

At this point, we investigated the number of nodes in a MIoT that turn out to be anomalous as a consequence of a single anomaly of an object o_j. Again, we repeated this experiment 100 times. Each time, we selected an anomalous object of the MIoT. The selected objects had different number of instances in the MIoT, ranging from 1 to 6. For each run, we computed the number of anomalous nodes detected in the MIoT. Then, we computed the averages, by grouping the cases based on the number of instances of the anomalous objects and, therefore, based on the number of IoTs of the MIoT involved in the anomaly.

The results obtained are shown in Figure 14.4, which shows how the number of anomalous nodes increases against the number of IoTs in a roughly linear way. This trend can be explained by considering that, even when the number of objects having

instances in many IoTs is usually limited with respect to the number of objects having instances in few IoTs, their anomalous behavior affects numerous nodes across several IoT and, consequently, their effect is amplified. On the contrary, anomalies observed on an object having instances in only one or two IoTs are more frequent. Yet, this is counterbalanced by the fact that each of these nodes only exerts a limited and localized impact, which affects only few nodes.

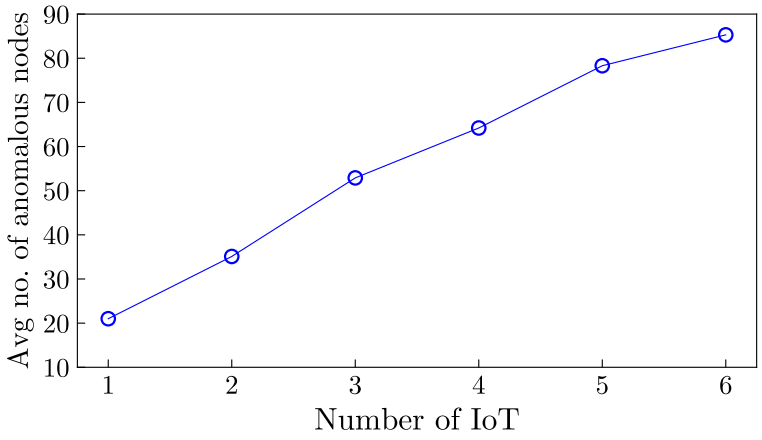


Fig. 14.4: Average number of nodes affected by anomalies against the number of IoT which an anomalous object participates to

Then, we aimed to characterize which of the node properties impacted the spread of anomalies the most. We repeated the previous experiment; but instead of choosing anomalous nodes randomly, we selected them based on their characteristics. A first characteristic that we considered was the outdegree of a node, i.e., the number of its outgoing arcs. In the various runs, we selected nodes with different outdegrees ranging from 10 to 60. For each of these values, we measured the average number of anomalous nodes throughout the MIoT detected by our approach. The results are illustrated in Figure 14.5, which clearly shows that the outdegree of anomalous nodes has a significant impact on the spread of the anomaly over the network. This result was not surprising, since it is consistent with the results about the information diffusion in social network analysis [613].

However, we argue that there is another form of centrality in social network analysis, which could be very promising as a node property to impact the spread of anomalies. This measure is closeness centrality. We recall that the closeness centrality of a node is defined as the reciprocal of the sum of the lengths of the shortest paths between the node itself and all the other nodes of the network.

Thus, we repeated the previous experiment; but this time we selected the anomalous nodes based on their closeness centrality. The values of this parameter for the

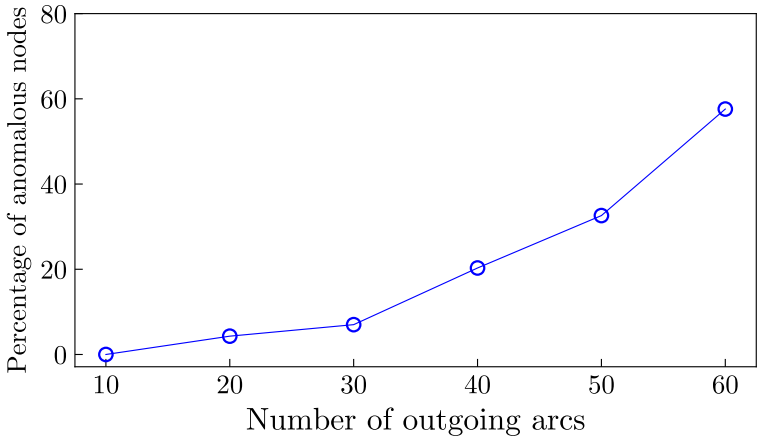


Fig. 14.5: Average percentage of anomalous nodes against their degree centrality

nodes selected ranged from 0.05 to 0.45. The results obtained are shown in Figure 14.6, where we can observe that our intuition was right. Closeness centrality is really a key parameter in the spread of anomalies in a MIoT. It is even more important than degree centrality in this task. In our opinion, this result is extremely interesting because the impact of closeness centrality on anomaly diffusion is substantial, whilst the role of this parameter was a-priori much less obvious than the one of degree centrality.

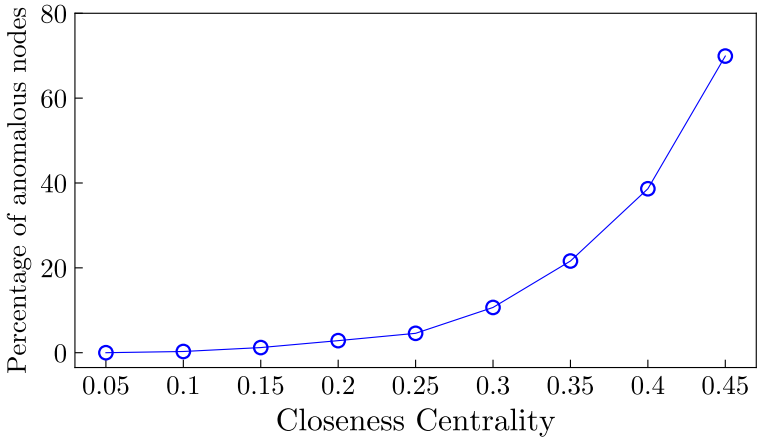


Fig. 14.6: Average percentage of anomalous nodes against their closeness centrality

As a final test on the forward problem, we evaluated the running time necessary to compute the anomaly degree δ_j of an object o_j in a MIoT against the number of its nodes. The results obtained are reported in Figure 14.7, where we can observe a polynomial (specifically, a quadratic) dependency of the running time against the number of nodes of the MIoT. This can be explained by the fact that, during the

computation of the recursive formula of δ_{j_k}, the values of α_{jq_k} tend to 0 rapidly while moving away from the node n_{j_k}.

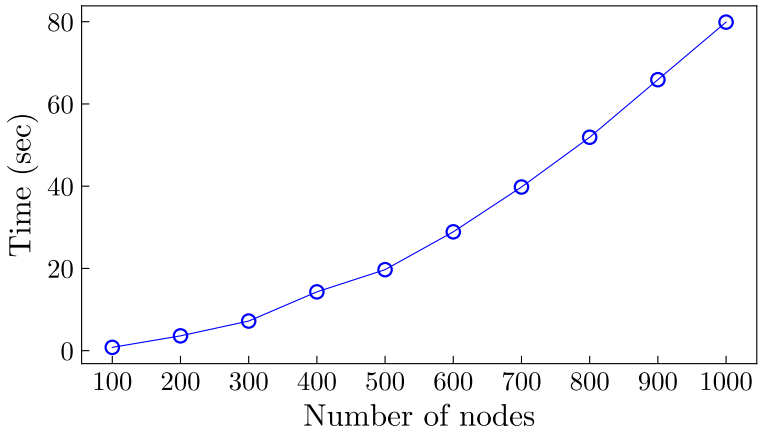


Fig. 14.7: Running time (in seconds) needed to compute δ_j in a MIoT against the number of its nodes

14.2.3 Analysis of the inverse problem

In this section, we present the results of the tests we carried out to validate our approach for solving the inverse problem. We recall that our solution to this problem starts from an i-arc of a MIoT that presents an anomaly whose origin is not known. It applies a greedy algorithm, which aims at detecting the node that originated the anomaly.

During this test, we repeated 100 times the following tasks. We simulated an anomaly on an object and, then, we randomly selected an anomalous i-arc from the whole MIoT. We applied our solution of the inverse problem on this arc and computed the following:

- the number of hits, i.e., the percentage of times our approach detected the anomaly source correctly (we call S_0 this scenario);
- the percentage of times our approach terminated in a node belonging to the $ONbh$ of the anomalous node and, therefore, being 1 hop away from it (we call S_1 this scenario);
- the percentage of times our approach terminated in a node being 2 hops far from the anomalous node (we call S_2 this scenario);
- the percentage of times our approach terminated in a node being more than 2 hops away from the anomalous node (we call S_3 this scenario).

The results obtained are reported in Figure 14.8. They show that our approach is capable of correctly identifying the anomaly source in most cases. In a fraction of cases it stops very near to the anomalous node, i.e., 1 or 2 hops away from it. The slightly higher frequency of the fourth case can be explained by the fact that the starting i-arc of the test is chosen randomly and, therefore, can be very far from the anomalous node. As a consequence, it comprises a relatively high number of cases (3, 4, 5 or more hops away from the anomalous object).

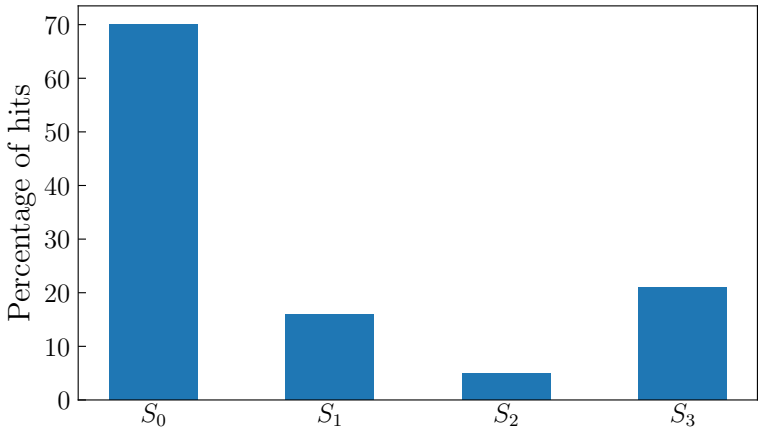


Fig. 14.8: Percentage of times when our approach correctly detects the anomaly source (indicated by the label 0) or terminates in a node being 1, 2 or more than 2 hops far from it

Next, we computed the average running time of our approach. Similarly to what we have done for the forward problem, we evaluated this time against the number of the MIoT nodes. The results obtained are shown in Figure 14.9, where we can observe that the running time increases polynomially against the number of MIoT nodes. This result can be explained by the fact that the greedy algorithm underlying our approach reaches the correct node, or a near one, in few iterations and by the fact that, on average, an anomaly on an i-arc can be observed only when this is not too far away from the node where the anomaly originated.

14.3 Use case

All of the devices installed in urban infrastructures, such as smart lighting systems and traffic management ones, contribute to the ecosystem of a so called *smart community*. This last one integrates a series of technological solutions for the definition and implementation of innovative models for the smart management of urban areas.

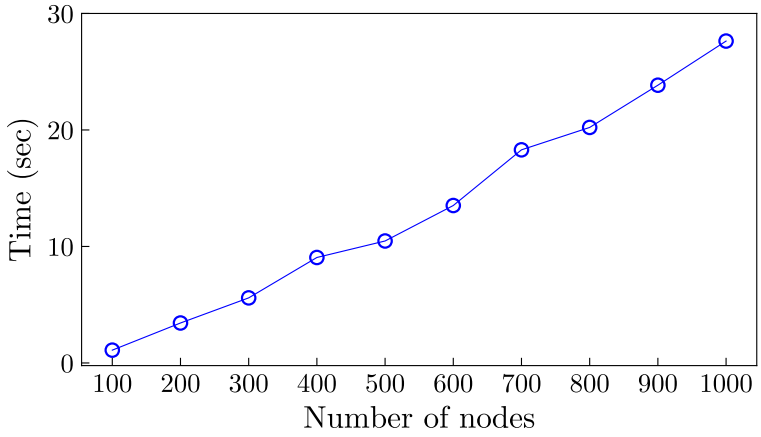


Fig. 14.9: Average running time (in seconds) of our approach for solving the inverse problem

One of the main challenges of the next generation of Information and Communication Technologies (ICT) applied to smart communities is the collection, integration and exploitation of information gathered from heterogeneous data sources, including autonomous smart resources, like SO, sensors, surveillance systems, etc., and human resources, such as posts in social networks. Another key challenge is the application of artificial intelligence tools, such as the ones based on automated reasoning, to advance state-of-the-art in smart community management [129].

The use case we focus on in this section refers to a smart lighting system in a smart city. In particular, we consider a data-centric platform integrated in a smart city environment, in which data coming from sensors and social networks can boost smart lighting, by operating and tuning different smart lighting objects located in the smart city area. The aim of the whole system is to provide citizens with a smart and safe environment.

Data are gathered from three different main sources, namely sensors, social networks and alerts exchanged among citizens on a dedicated social platform. Sensors data are gathered from a set of sensors installed on each smart lamp and handle different measures, such as temperature and humidity, but also several events, such as the presence of a person or the presence of rain. Sensors and smart lamps are organized in a Wireless Sensor Area Network (WSAN). Social networks data include geo-localized tweets from Twitter and posts from specific Facebook pages and are generated by smart personal devices.

All these data are stored in a data lake, which is directly accessed by a data mining module. This last module includes both sentiment analysis and anomaly detection tasks. The former focuses on the analysis of the data gathered from social posts. A polarity score, i.e., a positiveness/negativeness degree, is assigned to each keyword

that can be extracted from a post, and is used to intercept crucial information from the citizens moving around the city. In order to unambiguously single out significant information for the application context, keywords are mapped onto a specific urban taxonomy; this task is also carried out with the support of Babelnet [458]. Furthermore, thanks to the geo-localization of posts, information regarding a specific area of the smart city can be analyzed and assigned to the correct area.

Some data mining tasks are also carried out in order to identify, among other things, situations requiring a variation in the intensity of illumination for some area, for instance because of a variation in the security level perceived by citizens therein. Each smart lamp can communicate with neighboring ones in order to report variations in lighting parameters, as received by the mining module.

Anomaly detection works on both temporal data, gathered from sensors, and polarity scores, extracted by sentiment analysis, in order to detect potential anomalies. It exploits the taxonomies and the techniques presented here (Sections 4 and 5).

In our scenario, the urban area is modeled as a MIoT consisting of a set of IoTs $\{\mathcal{I}_1, \mathcal{I}_2, \cdots, \mathcal{I}_m\}$, each one associated with a portion of the area. The set of the objects of $\mathcal{M}$ comprises both the set of sensors, installed in the various smart lamps, and the set of personal devices of people who are moving around them. If an object o_j of the MIoT is active in the k^{th} portion of the urban area, it has an instance ι_{j_k} in the IoT $\mathcal{I}_k$. Clearly, when a person with a smart device o_j moves around different portions of the urban area, each one corresponding to a single IoT, o_j will have different instances, one for each IoT. An object o_j corresponding to a smart lamp sensor in the k^{th} urban area is fixed, and will contain only one instance ι_{j_k} in the corresponding IoT $\mathcal{I}_k$.

A transaction $Tr_{jq_{k_i}}$ between two object instances ι_{j_k} and ι_{q_k} can be generated in different ways. First of all, when citizens move around the various IoTs, they generate posts and alerts with their mobile devices. In this case, the transaction is associated with each post or alert. Sensors send transactions to the platform for sensed data, and smart lamps communicate with each other for parameter adjustments. Each of these events is translated into a transaction $Tr_{jq_{k_z}}$. Even the data mining module may send messages to the various smart lamps, thus generating transactions $Tr_{jq_{k_z}}$ in the MIoT.

15

Increasing protection and autonomy of smart objects in the IoT

In recent years, the Internet of Things paradigm has become pervasive in everyday life attracting the interest of the research community. Two of the most important challenges to be addressed concern the protection of smart objects and the need to guarantee them a great autonomy. For this purpose, the definition of trust and reputation mechanisms appears crucial. At the same time, several researchers have started to adopt a common distributed ledger, such as a Blockchain, for building advanced solutions in the IoT. However, due to the high dimensionality of this problem, enabling a trust and reputation mechanism by leveraging a Blockchain-based technology could give rise to several performance issues in the IoT. In this chapter, we propose a two-tier Blockchain framework to increase the security and autonomy of smart objects in the IoT by implementing a trust-based protection mechanism. In this framework, smart objects are suitably grouped into communities. To reduce the complexity of the solution, the first-tier Blockchain is local and is used only to record probing transactions performed to evaluate the trust of an object in another one of the same community or of a different community. Periodically, after a time window, these transactions are aggregated and the obtained values are stored in the second-tier Blockchain. Specifically, stored values are the reputation of each object inside its community and the trust of each community in the other ones of the framework. In this chapter, we describe in detail our framework, its behavior, the security model associated with it and the tests carried out to evaluate its correctness and performance.

The material presented in this chapter was derived from [219].

15.1 Methods

15.1.1 The reference IoT Model

In this section, we illustrate the model adopted to represent and handle the entities characterizing our framework. In our model, the main actor is the smart object. It has associated a profile with: *(i)* an identifier; *(ii)* a set of features characterizing it; *(iii)* a set of services it offers; *(iv)* the information needed for the communication with

other smart objects (such as the MAC address, the IP address, etc.). The smart objects of the IoT can be partitioned into communities according to some rules (see Section 15.1.2 for the rules we adopted in this paper). Each smart object belongs to exactly one community. Smart objects can communicate with each other. This communication relies on suitable transactions involving a source smart object and a target one. Transactions can be performed to require the features/services declared by the target smart object (we call them *ordinary* transactions) or to test what it declared in order to evaluate its reliability (we call them *probing* transactions). Furthermore, transactions can be classified into *intra-community*, if they involve smart objects of the same community, or *inter-community*, if they involve smart objects belonging to different communities.

Each community has associated a Local Blockchain; it registers information about the transactions having a smart object of that community as trustor. The overall IoT has associated a Global Blockchain; it registers aggregated information produced periodically starting from the probing transactions registered in the Local Blockchains. As we will see in the following, the information stored in the Global Blockchain regards: *(i)* the list of smart objects belonging to each community and their reputation scores inside their communities; *(ii)* the trust of each community in the other ones of the IoT. In order to improve the readability of this paper, in Table 15.1 we report the main symbols used in it.

15.1.2 Technical description of our approach

In this section, we present the core of our approach. In particular, we describe our strategy to build the local and global Blockchain tiers to support the definition of a trust and reputation solution for smart objects. This section is organized as follows: In Subsection 15.1.2, we provide the general overview of the proposed scheme. In Subsection 15.1.2, we discuss the computation of reliability measures for smart objects inside a community. In Subsection 15.1.2, we extend this activity to smart objects belonging to different communities.

General overview of the proposed scheme. As said in the Introduction, our goal is designing a framework to allow the protection of smart objects in an IoT scenario and, at the same time, the promotion of their autonomy. The autonomous interaction between smart objects occurs through mechanisms allowing each of them to understand what features/services can be provided by the smart objects it is in contact with [56]. The increasing of autonomy poses important challenges in terms of smart objects reliability. To address these challenges, we introduce in our framework suitable trust and reputation measurement techniques, which allow smart ob-

Parameter	*Meaning*
o_{i_k}	The smart object o_i of the community C_k.
C_k	A generic community of our framework.
tr_{i_k}	A trustor object belonging to C_k.
te_{j_q}	A trustee object beloning to C_q.
req_{i_j}	A probing transaction from tr_{i_k} to te_{j_q}.
P_{i_j}	A portion of smart objects able to answer req_{i_j}.
$\widehat{P_{i_j}}$	The "pruned" P_{i_j}.
out_j	The output to req_{i_j} provided by te_{j_q}.
$\overline{out_{i_j}}$	The average output to req_{i_j} provided by the smart objects of $\widehat{P_{i_j}}$.
T_{i_j}	The trust of tr_{i_k} in te_{j_q} after a probing transaction.
$\mathcal{F}$	A similarity function between out_j and $\overline{out_{i_j}}$.
τ	The tolerance admitted by the similarity function.
TrS_j	The set of trustors for te_{j_q}.
$\widehat{TrS_j}$	The "pruned" TrS_j.
R_j^ω	The reputation of te_j in C_q after the time window ω.
α	The weight of the importance of past data in R_j^ω.
$\overline{T_j^\omega}$	The average trust in te_j after the time window ω.
$\overline{t_q}$	The smart object in C_q supporting tr_{i_k} in its probing task.
$\overline{t_k}$	The smart object in C_k supporting te_{j_q} in its answer to tr_{i_k}.
$T_{k_q}^\omega$	The trust of C_k in C_q after the time window ω.
β	The weight of the importance of past transactions in the computation of $T_{k_q}^\omega$.
p	The probing probability.
$\Lambda_{k_q}^\omega$	A function evaluating the role of past transactions in the computation of $T_{k_q}^\omega$.
$\overline{T_{k_q}^\omega}$	The average trust values of the smart objects of C_k in the smart objects of C_q.
$\mathcal{I}_q^\omega$	A parameter denoting how much C_q has changed in the time window ω.
δ	A damping factor denoting the initial trust of a community.
$\mathcal{R}_{i_j}^\omega$	The reliability assigned by tr_{i_k} to te_{j_q} before starting a new transaction.

Table 15.1: The main abbreviations used throughout this paper

jects to assess the reliability of the smart objects they are in contact with, in order to "consciously" filter the information received. Following the standard approach accepted in the literature [4], our solution leverages two main tools to assess the reliability of smart objects. The first consists in the capability of verifying the ability of smart objects to provide the features/services they have declared. The second, instead, consists in the possibility of considering objects as belonging to a society in which information about measured objects' reliability can be propagated and is, hence, made available to all members. To enable the possibility for smart objects to

assess whether their peers are reliable in providing the features/services they advertise, as done in [102], we adopt an approach based on probing transactions. As thoroughly explained in the Introduction, to support the evaluation mechanisms mentioned above, in particular to certify probing transactions, our approach uses a Blockchain-based solution. Regarding this choice, we highlight that, due to the distributed and decentralized nature of the Blockchain paradigm, approaches combining Blockchains and IoT are increasingly attracting interest in both the research and industrial context [476, 208, 519, 217, 570]. Many promising IoT applications that use a Blockchain-based layer to improve the autonomy and security of the involved smart objects have already been proposed. To give some examples, we can mention approaches in the contexts of device configuration management, sensor data storing or micro-payments [207, 644, 380]. However, it is also well known that there are many problems regarding the use of the Blockchain technology in the IoT [170, 84, 206, 502, 400, 314, 539, 540, 565]. They are mainly related to the high number of nodes involved and the large amount of data generated, as well as the low computational power of many smart objects.

Our approach addresses these issues by leveraging a two-tier Blockchain. In particular, we assume that smart objects are grouped into suitable communities according to different criteria. Within these communities, smart objects can adopt control mechanisms aimed at identifying anomalous behaviors and making interactions as secure as possible. We point out that our approach is orthogonal to how communities are formed. To this end, we could use any approach, such as the one proposed in [470]. This requires that smart objects in a community should present a certain level of redundancy of the features/services offered. This property is also fundamental in our approach. Indeed, in order to enable mechanisms to evaluate the ability of a smart object to provide a given declared feature/service, it is necessary to have an alternative source as reference (see below for details). Therefore, the first Blockchain tier is internal to a single community and is intended as a local public ledger in which the probing transactions inside a community are stored. The second tier is global and concerns the whole IoT scenario; this level reports only *aggregated* information about the different communities.

As explained in the Introduction, the local tier could be implemented using a fully IoT-based solution that uses lightweight strategies to provide a public shared ledger. IoT devices alone cannot keep up with the computational power and energy demands of traditional Blockchains. In fact, most of them are based on the Proof-of-Work paradigm, which is not suitable for the IoT context. Nevertheless, several approaches to build lightweight Blockchains for IoT have been proposed in the scientific literature [476, 208, 519, 217, 570]. Among others, a very promising and up-

to-date project is IOTA[1]. This is one of the most popular Blockchain-based ecosystems (at the time of writing this paper, the cryptocurrency underlying this system is ranked 24^{th} in the market capitalization). Furthermore, it has an important developer base and also supports smart contracts, thanks to the QUBIC protocol [447].

IOTA is based on a micro-transaction infrastructure for the IoT context. It represents a more energy-efficient technology than classic Blockchains, because it increases the transaction speed and makes it possible to perform transactions without paying any fees. The foundation of IOTA is the adoption of an acyclic directed graph called Tangle [570, 503]. In it, there are no blocks and each new transaction references the previous two ones in order to gain network consensus. Even with these tricks, the data to store can grow rapidly because there are many interconnected devices. To address this issue, IOTA proposes two solutions depending on the overall environment. The first consists in the creation of special entities, called *Permanodes*, which keep all Tangle data. The second involves the *snapshotting* of data, i.e., storing only the balances of the local addresses and deleting everything else. In this way, it is possible to group together several transactions of the same address in a log, which requires less storage. Of course, the first solution is the most expensive one because it implies the creation of a new entity with more resources in terms of computational power and energy than common IoT devices. For this reason, in our case, the second solution seems more suitable, since it requires less storage and has many possible configurations (such as global and local *snapshotting* [570]).

Although we have described IOTA in more detail, we repeat that our approach is orthogonal to the specific solution adopted to have a lightweight Blockchain in the IoT.

Instead, the global tier can be implemented on any Blockchain network, e.g., Ethereum[2] or HyperLedger[3]. This tier is only used to store aggregated data involving multiple communities. The frequency of use of the global tier is very low, compared to the one of local tiers. Therefore, the cost to access it is negligible for the smart objects in our framework. Since there are no stringent requirements for the global tier, as there are for the local ones, in the following, due to space limitations, we will not discuss this topic in detail.

Figure **??** shows the general architecture of our approach. As can be seen from this figure, smart objects are organized in heterogeneous communities. There is no restriction on the interaction between smart objects of different communities. In fact,

[1] www.iota.org

[2] www.ethereum.org

[3] www.hyperledger.org

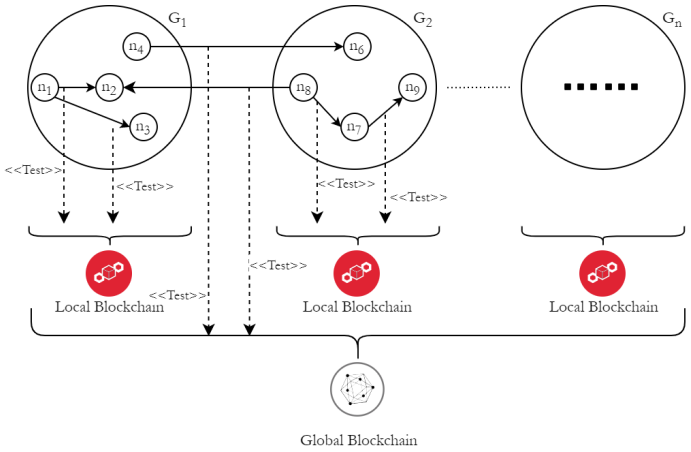


Fig. 15.1: General architecture of our approach

in our approach, smart objects can continue to interact as freely as in any other IoT, thus preserving one of the main features of this kind of network [283, 47, 48].

To control and limit the transaction volume to be analyzed, the normal communication bursts among smart objects are conceptually divided into time windows. Within each window, in addition to normal interactions, probing transactions are performed. These tests are randomly generated; in particular, each smart object can decide to test another one belonging to its community with a certain probability. The test performed must be compliant with the features/services offered by the tested smart object.

In order to verify the reliability of the tested smart object, the tester requires the support of other smart objects belonging to its community and providing the same feature/service. With regard to this, based on the feature/service redundancy hypothesis characterizing the smart objects of each community, we assume that it is always possible to identify a subset of smart objects providing the same features/services of the tested one. They can be involved in the verification task.

Tests are used to compute the reputation of smart objects within their communities. All transactions associated with tests are recorded in the Local Blockchain of the community in order to make them provable and, therefore, reliable. After a defined time window, the reputation of each smart object in its community is computed by aggregating the results of the tests it has undergone. This also allows a limitation of the growth of the number of transactions in the Local Blockchain. The computation of the reputation values can be performed directly on the Blockchain using a dedicated smart contract. After this task, the list of community members, together with the corresponding reputation scores, is published in the Global Blockchain. Smart objects having a reputation score below a certain threshold are automatically removed from the community.

Our approach also guarantees protection in case of interaction between members of different communities. In order to deal with possible inter-community attacks, it provides mechanisms to test the reliability of smart objects even in case of communications between members of different communities. In particular, when two smart objects belonging to different communities contact each other, one of them can undergo the other one to a test with a certain probability. In order to compute the test result, the tester object requires to the tested one the features/services it declares. Furthermore, it performs the same request to other objects belonging to the same community as the tested object. The test result is saved in the Local Blockchain of the community which the tester object belongs to. Analogously to what happens for local tests, after a defined time window, the transactions associated with the tests of smart objects belonging to external communities are aggregated. In this way, we get a trust value of a community in each external community with which at least one of its objects interacted. Also the trust values between communities are saved in the Global Blockchain.

Therefore, the Global Blockchain stores the reputation of each smart object in its community, as well as the trust of each community in the other ones it interacted with in the past. If there has been no interaction between two communities, our approach assumes that each of them has a default trust value in the other one, equal to the minimum trust value allowed.

Thanks to all information stored in the Global Blockchain, when a smart object o_{i_k} of a community C_k wants to interact with a smart object o_{j_q} of a community C_q, $C_q \neq C_k$, o_{i_k} can compute the reliability of o_{j_q} taking into account the reputation of o_{j_q} within C_q and the trust of C_k in C_q.

The Local Blockchain tier: assessing trust and reputation inside communities. In this section, we illustrate the tasks carried out by our approach to evaluate trust and reputation inside communities. In particular, in Subsection 15.1.2, we present the computation of the trust between two smart objects belonging to the same community; instead, in Subsection 15.1.2, we describe the computation of the reputation of a smart object inside its community.

Measuring trust in point-to-point interactions

In an IoT scenario, some malicious owners may exist. They could use their misbehaving devices for self-interests, for instance to perform some attacks to ruin the reputation of other IoT devices. For this reason, trust and reputation management is a key issue in IoT, and many researches on this topic can be found in the past liter-

ature [651, 59, 57, 58, 153, 148, 150, 569]. Typically, in the evaluation of trust and reputation, the following factors are considered [4]:

- *The quality of service provided by the device*. Also known as QoST (Quality of Service Trust), it is the ability of an IoT device to provide a service with a certain level of quality. QoST generally refers to performance and may depend on several parameters, such as competence, cooperativeness, reliability, task completion capability, and so forth.
- *The trust derived from the relationship between two objects or between an object and its owner*, also known as *Social Trust*. It may depend on parameters like intimacy, honesty, privacy, centrality, connectivity, etc. It is prevalent in Social IoT systems, where IoT devices must be evaluated based not only on QoST but also on the behavior of their owners [46].

A challenge-response approach is generally used for QoST computation. The idea is to estimate the reliability of the response obtained in a challenge between a trustor and a trustee. Generally, the trust interaction between two smart objects can be represented by means of a triplet `<trustor, trustee, feature/service>`. The field `feature/service` denotes the subject of the evaluation and is closely related to the application context. As an example, this can be a service offered (like the news of the day) or a simple measurement of a quantity that the trustee can return to the trustor.

The social trust, instead, refers to the social behavior of an object, and possibly its owner, in its interaction with each other object in its community or, more generally, in the IoT.

In our system, we adopt a mixed solution that considers both the ability of an object to answer a probing query and the information on the same feature/service that can be obtained from other IoT objects answering the same query.

As said before, in our IoT framework, smart objects are grouped into communities. These are built taking care to guarantee the heterogeneity of features/services provided by its objects, and the redundancy in the provisioning (i.e., more objects can offer the same feature/service in one community).

Each node in a community can activate a probing activity towards another node in the same community by requesting the provision of a feature/service that the latter has declared to provide.

So, given a feature/service, say req_{i_j}, requested by a trustor tr_i to a trustee te_j, it is possible to identify a partition P_{i_j} of smart objects in the community able to provide an answer to req_{i_j}. Then, a "pruning" is performed on P_{i_j} to select the smart objects that are most likely to return an output close to the one returned by te_j. We call $\widehat{P_{i_j}}$ the partition P_{i_j} after this pruning. For example, if the required feature/service regards

temperature measurement, P_{i_j} contains all the smart objects in the community able to measure temperature. Since this parameter is related to the context where a smart object operates, $\widehat{P_{i_j}}$ contains only those objects of P_{i_j} having a context compatible with te_j [200].

To measure the trust T_{i_j} of tr_i in te_j, we consider the deviation between the output out_j returned by te_j and the average output $\overline{out_{i_j}}$ provided by the smart objects of $\widehat{P_{i_j}}$. In particular, T_{i_j} can be computed as:

$$T_{i_j} = 1 - \mathcal{F}(out_j, \overline{out_{i_j}}, \tau) \tag{15.1}$$

Here, $\mathcal{F}$ is a dissimilarity function that returns real values in the range [0,1]. The greater the dissimilarity between out_j and $\overline{out_{i_j}}$ and the higher the value returned by $\mathcal{F}$. Clearly, $\mathcal{F}$ depends on the parameter we are measuring and the range of values it can assume. It also takes into account the tolerance τ allowed by the parameter. Also τ can assume values included in the real range [0,1].

Observe that the definition of $\mathcal{F}$ is orthogonal to our approach and may depend on several factors related to the parameter to measure or the service required. In case of services, $\mathcal{F}$ can take into account parameters such as Quality of Service (QoS) or Quality of Experience (QoE) [214, 242]. In case of a numerical output, linked for instance to a measurement, a possible definition of $\mathcal{F}$ could be the following:

$$\mathcal{F}(out_j, \overline{out_{i_j}}, \tau) = \frac{|out_j - \overline{out_{i_j}}|}{max(out_j, \overline{out_{i_j}})} \cdot (1 - \tau)$$

From the implementation point of view, our approach is based on a permissioned Blockchain in which there are some smart contracts dedicated to the computation and propagation of trust and reputation values. In particular, our approach saves probing transactions in a Local Blockchain. For this purpose, for each transaction, it activates a dedicated smart contract that implements the steps reported in Algorithm 7. The same steps are shown graphically in Figure 15.2. The choice of using a permissioned Blockchain allows the definition of different policies for smart objects. In particular, these policies could be closely related to the criticality level of the communities. For example, in communities with a high level of criticality, joining can be made during the installation and maintenance tasks performed by system administrators. Instead, some communities, such as those related to smart home scenarios, might define less restrictive joining policies. In this case, smart objects could autonomously join a community.

Require: The probability p_{act} that tr_i activates a probing transaction with te_j
  generate a random value v_{act}
  if $(v_{act} < p_{act})$ **then**
    tr_i activates a probing transaction asking features/services to te_j
    te_j provides the required output out_j
    the smart objects of $\widehat{P_{i_j}}$ are required to provide the same output as te_i
    the average value $\overline{out_{i_j}}$ is computed
    the value of the trust T_{i_j} of tr_i in te_j is determined by applying Equation 15.1
    the transaction and the corresponding trust is stored in the Local Blockchain
  end if

Algorithm 7: Smart contract for the computation of the trust of a trustor tr_i in a trustee te_j

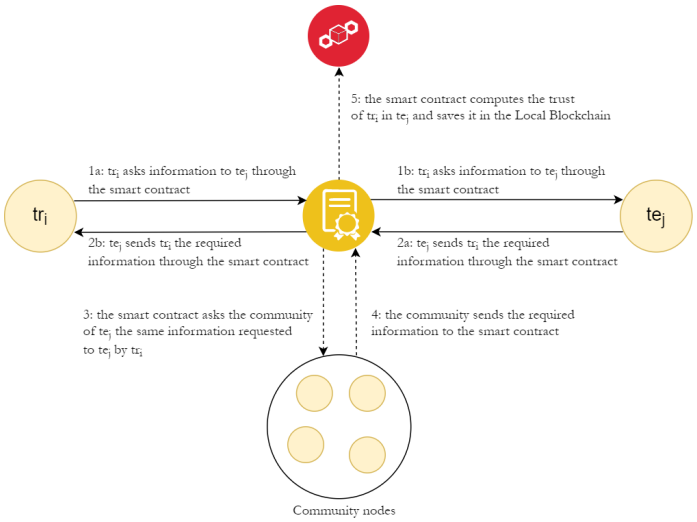


Fig. 15.2: Computation of the trust of a trustor tr_i in a trustee te_j

Using a lightweight Blockchain for the computation of reputation values

Once many probing transactions are available in a community, it is possible to compute an aggregate measure, called reputation, for each smart object. It summarizes the opinion of the whole community towards the object [290]. In our case, the computation of the reputation also allows the implementation of a technique to keep the Blockchain size low.

Following an approach similar to the one presented in [555], we define a time window ω and consider all probing transactions made in this time window. At the end of ω, our framework aggregates the information about trusts and computes the reputation of the smart objects in their community as specified below.

Let TrS_j be the set of nodes that required at least one probing transaction to te_j. The reputation R_j^{ω} of te_j at the end of the time window ω is computed as:

$$R_j^{\omega} = \begin{cases} \alpha \cdot R_j^{\omega-1} + (1-\alpha)\overline{T}_j^{\omega} & if\ \ TrS_j \neq \emptyset \\ \\ R_j^{\omega-1} & otherwise \end{cases} \tag{15.2}$$

Here:

- $R_j^{\omega-1}$ is the reputation of te_j at the end of the previous time window;
- α is a parameter used to weigh the importance of past data with respect to the present ones. It plays an important role in the ability of our approach to react to anomalous situations. In fact, a high value of α would give a great importance to the historical behavior of a node, smoothing the effect of recent temporary variations in its interactions with other nodes. On the contrary, a low value of α would make our approach extremely reactive to any variation in the behavior of a node. A perfect balance between the history of a node and its recent interactions (which is achieved by setting $\alpha = 0.5$) might be a good choice for most application contexts. However, low values of α could be adopted in critical scenarios, where a high security level must be guaranteed. In this case, having a fast reaction of the reputation system is essential to exclude nodes that start to show a suspicious behavior.
- $\overline{T_j^{\omega}}$ is the average trust obtained by aggregating the values of trusts in te_j computed during ω. It can be obtained as:

$$\overline{T}_j^{\omega} = \frac{\sum_{tr_i \in TrS_j} T_{i_j}}{|TrS_j|} \tag{15.3}$$

The reputation values thus computed have a great influence on the evaluation of communities. In fact, the smart objects that do not meet the minimum reputation requirements are removed from the community. After these computations, the Local Blockchain is reset, following the approach described in [555], and all the transactions occurred during ω are no longer considered.

From a technical point of view, the computation of the reputation is carried out by a smart contract activated at the end of each time window. Given a community, the smart contract computes the reputation of each of its smart objects using Equation 15.2. The smart contract time scheduling can be done following existing technical approaches, for example the Ethereum Alarm Clock[4]. Note that the length of the time window can be related to the number of transactions generated in the Blockchain, instead of a clock. In this case, when transactions exceed a certain threshold, the smart contract is activated.

Once all the reputation values have been computed, the smart contract updates the list of smart objects that can be still part of the community (i.e., the smart objects

[4] https://www.ethereum-alarm-clock.com/

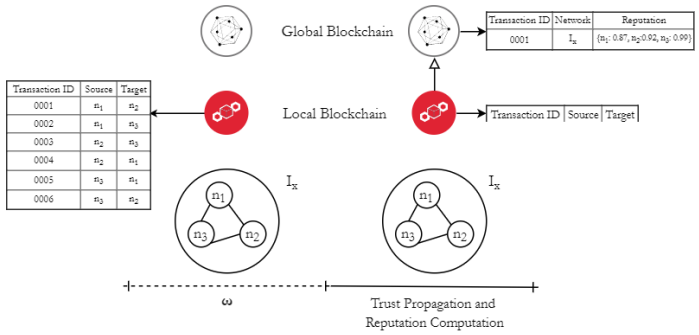


Fig. 15.3: Transaction aggregation and computation of the reputation of the smart objects of a community

whose reputation score is higher than a minimum threshold). This list, together with the reputation score of the smart objects (also of the ones that will be removed), is registered in the Global Blockchain through a new transaction. This behavior is represented in a more coded way in Algorithm 8. Instead, a graphical representation is provided in Figure 15.3.

Require: the time window ω, the reputation threshold th_R and the weight α of the past reputations
wait for the end of the time window ω
for te_j in the community C **do**
  compute $\overline{T}_j^{\omega}$ applying Equation 15.3
  compute R_j^{ω} applying Equation 15.2
  if $R_j^{\omega} < th_R$ **then**
    remove te_j from C
  end if
end for
register all the reputation values in the Global Blockchain

Algorithm 8: Transaction aggregation and computation of the reputation of the smart objects of a community

The Global Blockchain tier: towards reliable community level interactions. In this section, we discuss how to evaluate and activate reliable transactions between smart objects belonging to different communities. In particular, in Subsection 15.1.2, we illustrate how probing transactions between smart objects belonging to different communities are managed. In Subsection 15.1.2, we describe the computation of the trust of a community in another one. Finally, in Subsection 15.1.2, we show how a

smart object can evaluate the reliability of another smart object of a different community before starting a communication with it.

Enabling community level reliable interactions

So far we have seen the interactions between smart objects in the same community. However, as we said in Section 15.1.2, our framework also allows smart objects from different communities to interact with each other. This is done implementing a hierarchical approach, that is possible thanks to the presence of a two-tier Blockchain framework. Our approach is inspired by some of the ideas proposed in [190].

In our framework, the smart objects of each community are free to interact with any smart object of the framework, even those belonging to different communities. In the latter case, a smart object can rely on the information concerning the community of the smart object it wants to communicate with and the reputation of this last object in its community. This information is registered in the Global Blockchain.

However, different attack scenarios may lead to the fact that the only information about the reputation of a smart object, resulting from its interactions within its community, is not sufficient to guarantee its reliability. In fact, an attacked smart object could assume a polymorphic behavior, interacting positively with all the smart objects of its community but acting negatively with the ones belonging to other communities.

For this reason, our approach provides a mechanism to compute the trust also between objects belonging to different communities. To this end, we extend the probing mechanism described in Section 15.1.2. In particular, a smart object tr_{i_k}, belonging to a community C_k, can start the test of a smart object te_{j_q}, belonging to a community C_q, $k \neq q$, with a certain probability. For this purpose, tr_{i_k} makes a request req_{i_j} for a feature/service to te_{j_q}.

After that, tr_{i_k} randomly selects a node $\overline{t_q}$ and sends it the same request req_{i_j} previously sent to te_{j_q}. Clearly, req_{i_j} is built taking into account the features/services offered by te_{j_q} to make sure that this last object can provide an answer. The node te_{j_q} does not know that tr_{i_k} is making a probing transaction, while $\overline{t_q}$ is informed about it. The task of $\overline{t_q}$ is to select a partition TrS_q of the nodes of the community C_q that can answer req_{i_j}. Similar to what has been done in Section 15.1.2, among the nodes of TrS_q, our approach selects those ones being most likely to provide a correct answer. This leads to a "pruning" of TrS_q; we call $\widehat{TrS_q}$ the resulting set.

It is worth noting that, also in this case, the partition TrS_q of support nodes is selected from the trustee community itself. This choice strictly depends on the use cases considered in our approach and described in the Introduction. Indeed, al-

though a larger set of nodes from different communities might be involved in the selection of the support partition, our approach makes assumptions only on the availability of redundant services within communities and not across them. This choice also reduces the complexity of service discovery strategies [636, 512], which can be applied to just a controlled-size community, instead of a whole world-scale IoT.

The smart object $\overline{t_q}$ sends req_{i_j} to the objects of $\widehat{TrS_q}$ and, when it receives the corresponding answers, computes the average output $\overline{out_{i_j}}$. At this point, $\overline{t_q}$ must send this value to tr_{i_k}. However, to avoid attacks from this last object, $\overline{t_q}$ does not send $\overline{out_{i_j}}$ directly to tr_{i_k}. Instead, it randomly selects an object $\overline{t_k}$ of C_k and sends $\overline{out_{i_j}}$ to it by specifying the final receiver. Indeed, if $\overline{t_q}$ would send the answer to tr_{i_k} directly, this last could alter this answer and, therefore, force the assignment of a disadvantageous trust value to te_{j_q}. Instead, if $\overline{t_q}$ sends the answer to a random node $\overline{t_k}$ of the same community as tr_{i_k}, because all trust interactions are stored in the Local Blockchain, tr_{i_k} cannot change the answer provided by $\overline{t_q}$ through $\overline{t_k}$.

Finally, te_{j_q} returns its output out_j to tr_{i_k}. Again, in order to be protected from the attacks of tr_{i_k}, te_{j_q} can decide, with a certain probability, to forward out_j to tr_{i_k} through a randomly selected node $\overline{t'_k}$ of C_k, instead of sending out_j directly to tr_{i_k}. Again, in case of an indirect answer, te_{j_q} specifies the final receiver in the message sent to $\overline{t'_k}$. When tr_{i_k} receives out_j and $\overline{out_{i_j}}$, it can compute the value of the trust T_{i_j} by applying Equation 15.1. All communications made within the communities C_k and C_q continue to be saved in the corresponding Local Blockchains so that the test results can be verified by the other smart objects of the communities. Also these results are saved in the Local Blockchain of C_k and, therefore, can be partially reproduced starting from the transactions involving $\overline{t_k}$ and $\overline{t'_k}$ (if this last one has been involved by te_{j_q}).

Computing community trust

As seen in Section 15.1.2, when a time window ω has passed, probing transactions are used to compute the trust of the community C_k in the community C_q. For this purpose, we proceed as specified below. Let TrS_{k_q} be the set of the smart objects of C_q with which any smart object of C_k had a probing transaction during the time windows just passed. The trust $T^{\omega}_{k_q}$ of C_k in C_q at the end of ω can be computed as:

$$T^{\omega}_{k_q} = \begin{cases} \beta \cdot \Lambda^{\omega-1}_{k_q} + (1-\beta)\overline{T}^{\omega}_{k_q} & \text{if } TrS_{k_q} \neq \emptyset \\ \\ \Lambda^{\omega-1}_{k_q} & \text{otherwise} \end{cases} \tag{15.4}$$

Here:

- The parameter $\Lambda_{k_q}^{\omega-1}$ is defined as:

$$\Lambda_{k_q}^{\omega-1} = \mathcal{J}_q^{\omega} \cdot T_{k_q}^{\omega-1} + (1 - \mathcal{J}_q^{\omega}) \cdot \delta \tag{15.5}$$

  where:

  - δ is a damping factor that denotes the initial trust in a community C_q when no probing transaction has been requested to any of its smart objects.
  - $\mathcal{J}_q^{\omega}$ is an index that expresses how much C_q has changed (in the composition of its smart objects) since the previous time window (ω–1). It can be obtained by computing the Jaccard Coefficient between the sets of the smart objects present in C_q in the time windows $\omega - 1$ and ω.
  - $T_{k_q}^{\omega-1}$ is the trust of C_k in C_q in the time window $\omega - 1$.

  The rationale behind this equation is related to the fact that the trust of C_k in C_q depends on the interactions that the corresponding nodes had during the time window ω and on the ones they had during the other past windows. If C_q has changed heavily (because its smart objects present during ω are very different from the ones present in the past) then the historical trust must be reset to the damping value δ.
- β is a parameter weighting the importance of historical data compared to those obtained in the last time window. The role of β is identical to the one assumed by α in Section 15.1.2. Specifically, it can be used to adjust the adaptation level of our security mechanism to new probing results. Again, the lower β, the higher the importance of recent trust interactions.
- $\overline{T_{k_q}^{\omega}}$ is the average of the trust values that the smart objects of C_k had in the smart objects of C_q with which they interacted during ω. It can be computed by means of the following formula:

$$\overline{T}_{k_q}^{\omega} = \frac{\sum_{te_{j_q} \in TrS_{k_q}} \overline{T_j}}{|TrS_{k_q}|} \tag{15.6}$$

  where $\overline{T_j}$ is the average trust assigned by the smart objects of C_k to the smart object te_{j_q}.

The values of $T_{k_q}^{\omega}$ obtained through Equation 15.4 are published in the Global Blockchain. Also in this case, at a technical level, the activities carried out to obtain the trust values described above are managed through a dedicated smart contract.

Assessing smart object reliability for community-level interactions

In the previous sections we have seen that it is possible to compute the trust of a smart object in another one of the same community. We have also seen that each

smart object has a reputation within its community. Finally, we have seen that it is possible to compute the trust of a community in another one. These last two pieces of information are registered in the Global Blockchain and, as we will see, allow us to compute the reliability that a smart object of a community assigns to a smart object of a different community.

In particular, the reliability assigned by a smart object tr_{i_k} of a community C_k to a smart object te_{j_q} of a community C_q after the time window ω is computed as:

$$\mathcal{R}_{i_j}^{\omega} = \begin{cases} T_{k_q}^{\omega} \cdot R_j^{\omega} & \text{if } T_{k_q}^{\omega} \text{ is not null} \\ \\ \delta \cdot R_j^{\omega} & \text{otherwise} \end{cases} \tag{15.7}$$

The rationale behind this equation is as follows: In case C_q has interacted with C_k in the past (which implies that $T_{k_q}^{\omega}$ is not null), the reliability $\mathcal{R}_{i_j}^{\omega}$ assigned by tr_{i_k} to te_{j_q} is obtained by multiplying the reputation that te_{j_q} has within C_q with the trust of C_k in C_q. Instead, if there has been no interaction between C_k and C_q in the past, then, in order to obtain $\mathcal{R}_{i_j}^{\omega}$, the reputation of te_{j_q} in C_q is "corrected" with a damping factor equal to the one used in Equation 15.5 and indicating the minimum trust that a community has in another one of the framework. Clearly, tr_{i_k} only interacts with te_{j_q} if $\mathcal{R}_{i_j}^{\omega}$ is high enough, i.e., greater than a minimum acceptable value.

15.1.3 Security Model

In this section, we illustrate the security model conceived for our framework. In particular, we present both the attack model and a security analysis showing that our framework addresses its objectives also in presence of attacks. In the security analysis, we refer to classical attacks to reputation systems adapted to our approach [300, 307] .

Attack Model. As a preliminary assumption, we consider a realistic scenario in which a sufficient number of nodes is available so that our approach can be implemented successfully. Therefore, we do not consider anomalous situations or the startup time, in which the number of the nodes available in the framework is less than the minimum necessary.

In the analysis of security properties, we will consider that our threat model includes the following assumptions:

A.1 At most t smart objects can collude to break the security properties of the protocol.

A.2 The size of all the pruned support partitions, $|\widehat{P}|$ and $|\widehat{TrS}|$, is greater than t (see Sections 15.1.2 and 15.1.2).

A.3 An attacker cannot control a whole group of smart objects; moreover, she/he cannot own all the smart objects providing a certain service.

A.4 An attacker has no additional knowledge derived from any direct physical access to smart objects.

A.5 The Blockchain technologies exploited to implement both the Local and the Global tier are compliant with the standard security requirements already adopted for common Blockchain applications.

As for the first assumption, we recall that probing transactions are produced collaboratively by several smart objects in our protocol. Some of them might be corrupted but we assume the honesty of the majority of them, as done in [245, 183].

The list of the security properties (hereafter, SP) that our framework must assure is the following:

SP.1 Resistance to the Local and Global Blockchain tier Attacks.

SP.2 Resistance to Self-promoting Attacks.

SP.3 Resistance to Whitewashing or Self-serving Attacks.

SP.4 Resistance to Slandering or Bad-mouthing Attacks.

SP.5 Resistance to Opportunistic Service Attacks.

SP.6 Resistance to Ballot Stuffing Attacks.

SP.7 Resistance to Denial of Service (DoS) Attacks.

SP.8 Resistance to Orchestrated Attacks.

SP.9 Resistance to malicious probing exploitation.

Security Analysis. In this section, we focus on each of the security properties introduced above and analyze if and how our approach can guarantee it.

SP.1 *- Resistance to the Local and Global Blockchain tier Attacks*

This category of attacks aims at finding vulnerabilities in the Blockchain layers adopted in our framework. Of course, if one of the ledgers is compromised, our approach cannot work properly because probing transactions could be tampered or removed to modify the recorded behavior of each smart object involved. Even if, in the recent years, Blockchain has received a lot of attention from both the scientific community and the industry, the security of Blockchain is still subject of debate. A lot of approaches to face security flaws of the Blockchain in application scenarios related to ours have been proposed [400, 346]. Our approach is orthogonal with respect to these approaches. Indeed, we do not focus on improving Blockchain security, but we use it to implement a public ledger to store probing transactions among

smart objects, as well as reputation values. Therefore, as stated by Assumption **A.5**, we consider the Blockchain as a secure layer in our approach.

SP.2 - *Resistance to Self-promoting Attacks*

This attack occurs when a smart object manipulates its own reputation to increase it falsely and promote itself. It can be carried out by an attacker operating alone or organized in groups of collaborating identities.

In our approach, a smart object tests another one through probing transactions. The trust score, obtained thanks to them, is stored in the Local Blockchain of the corresponding community. After this, a reputation score is computed starting from these trust scores. Hence, a smart object cannot alter its own score by itself. This ensures data authenticity and integrity and, therefore, our framework's capability of resisting to such an attack.

As for inter-community transactions, a smart object cannot assign a false score to itself because, also in this case, our framework allows the computation of smart object reliability only after a set of probing transactions, devoted to evaluate the reputation of the smart object in its community and the trust of the other communities in this last one. All the probing transactions are stored in Local Blockchains and, after a time window, they are aggregated in the Global Blockchain.

However, even if source data is authentic, a self-promotion attack would be still possible if a single attacker (or more attackers) manipulates nodes through a Sybil attack [209]. In this case, the sybil nodes would collude to promote each other. However, this cannot be possible due to Assumptions **A.1**, **A.2** and **A.3**, which imply that only $t < |\widehat{P_{i_j}}|$ smart objects (where $|\widehat{P_{i_j}}|$ is the number of smart objects in the pruned partition involved in a probing transaction - see Section 15.1.2) can collude.

Furthermore, at a local level, since smart objects are not aware if they are answering a probing or a standard query, a malicious behavior of them would cause a reduction of their reputation score. Instead, at the global level, support smart objects for testing are chosen randomly (see Section 15.1.2). This inhibits an attacker to rely on the possibility of controlling both the tested smart object and the support one.

In the remote possibility that, by chances, the support smart object is controlled by the attacker, this last could force a low trust score for a target smart object. However, this malicious attempt will not strongly affect the overall trustworthiness of the target smart object. Indeed, our metrics considers the whole history of interactions and, therefore, the impact of outlier values is strongly reduced.

SP.3 *- Resistance to Whitewashing or Self-serving Attacks*

This attack occurs when a malicious smart object with a compromised reputation, also called traitor [429], behaves in such a way as to quickly degrade its reputation with the goal of being removed from the framework. After this, it asks to rejoin the framework with a fresh reputation score in order to continue behaving maliciously. This kind of attack cannot be carried out in our framework because reputation scores are stored permanently in both the Local Blockchains and in the Global one. Due to this fact, our approach keeps memory of a malicious behavior even after the corresponding smart object has been removed from our framework.

Actually, it is possible to define a time interval, say ϕ_{ban}, during which the object can no longer be part of the framework. After this interval, the object can be restored and can join its community again with the initial minimum reputation value. Of course, the tuning of ϕ_{ban} is strictly related to the safety level of the considered scenario. The higher the safety level, the higher the ban interval. In the extreme case, for a very critical scenario (e.g., smart grids, nuclear firms, and so forth), ϕ_{ban} can even tend to infinity (which is equivalent to a permanent removal of the banned node from the framework). It should be noted that, in case of object outage, the ban interval can also be estimated based on the time required by a system administrator of the local community to intervene and restore it.

The previous solution implies that our approach must be able to maintain a clear association between objects and their corresponding reputations. This assumes that each object has an appropriate identifier. However, in a real-world scenario, in which objects join the network autonomously, an attacker could forge a new identifier for an object each time it is banned from the system. In this way, she/he could try to whitewash the reputation of the object, which would be identified as a new actor. Consequently, she/he could make multiple attacks avoiding the banning interval.

The past literature on this topic reports several studies aimed to define mechanisms allowing the management of strong identifiers for smart objects even in untrusted scenarios. For example, the authors of [297] propose a fully decentralized, self-maintaining and lightweight approach to handle consistent ID-to-dynamic IP mappings and use them in the routing process. Other approaches are based on object fingerprinting and focus on the problem of identifying general characteristics that may be present in any IoT device, whose values allow the extraction of patterns to unambiguously identify a single specific object. For example, to compute object fingerprints, the authors of [188] extract 19 features from 802.11 probe fields, while the authors of [481] focus on a set of features related to TCP timestamp and clock characteristics. Still in this context, the authors of [423] consider the relationships

between objects and human actors in the IoT to model a new identifier format called GARI. Each of these approaches could be adopted by our model to ensure a robust mapping between trust and reputation values and the corresponding smart objects.

Actually, as for this issue, there is another research strand that proposes mitigation strategies for whitewashing attacks by leveraging a pessimistic attitude to the initial reputation values associated with newly added actors [664, 261, 376]. In this case, a new object, or an object with a new forged identifier, is admitted to the network with a low default reputation value, less than the chosen threshold. Therefore, it is automatically put in a suspended state for a time equal to ϕ_{ban}. By adopting this strategy, an attacker is discouraged from performing whitewashing attacks by changing the object identifier. In fact, joining the system with a new identifier would coincide with the case where a node is temporarily banned. This solution seems the most appropriate for our scenario because it allows our approach to be resistant to this type of attacks without the need to integrate strategies for managing object identifiers.

SP.4 *- Resistance to Slandering or Bad-mouthing Attacks*

In this case, an attacker tries to manipulate the reputation of other smart objects by reporting false data. The attack can be carried out by a single smart object or a coalition of smart objects. Our model is resistant to this kind of attack because of its strict feedback mechanisms and the fact that the input validation is based on the Blockchain technology.

In particular, as for the intra-community case, smart objects are not aware if they are answering a test or a query. Hence, being malicious for an object could mean lowering its own reputation score and, after a while, being removed from the framework. Observe that, in this case, controlling a coalition of smart objects would not guarantee any benefit to the attacker.

As far as inter-community communications are concerned, several interesting situations should be analyzed. Specifically, assume that a smart object, say tr_{i_k}, belonging to a community C_k, decides to test another smart object te_{j_q}, belonging to a community C_q. As explained in Section 15.1.2, tr_{i_k} randomly chooses a support smart object, say $\overline{t_q}$, belonging to C_q. In turn, $\overline{t_q}$ has to select a pruned partition $\widehat{TrS_q}$ of smart objects of C_q that can answer the probing query. At this point, the following Slandering Attack attempts could be carried out:

1. The attacker tries to control $\widehat{TrS_q}$ in such a way that, after $\overline{t_q}$ sends req_{i_j} to the objects of $\widehat{TrS_q}$, it receives only false answers (or, at least, a great majority of false answers) from them. Of course, in this case, the computation of the trust score

of te_{j_q} would be compromised. However, this scenario cannot happen due to Assumptions **A.1**, **A.2** and **A.3**. Indeed, according to them, the attacker cannot control the overall community and only $t < |\widehat{TrS_q}|$ smart objects can collude. As for the case in which a partial attack occurs, smart objects for testing are randomly chosen. This lowers the probability of selecting two or more colluding smart objects among the ones controlled by the attacker, which are at maximum t. Finally, in the very remote case in which all the t smart objects controlled by the attacker have been included in the partition, this malicious attempt would impact the single trust value computed for te_{j_q}. However, it does not affect the overall trust score yet. In fact, as already said, our metric is designed in such a way as to average all trust values. Therefore, no overall advantage is achieved by the attacker in this case.

2. $\overline{t_q}$ could send req_{i_j} to the smart objects of $\widehat{TrS_q}$ and, after having received the corresponding answers, it returns a corrupted average output. In this case, since the choice of $\overline{t_q}$ is random, the attacker cannot control this situation and design a global attack strategy. As a consequence, even in this case, our trust and reputation model is not compromised.
3. tr_{i_k} lies on the answers of te_{j_q} and $\overline{t_q}$. As explained in Section 15.1.2, in order to contrast this case, te_{j_q} and $\overline{t_q}$ can send their responses to randomly selected objects of C_k. These last ones will use the Local Blockchain to securely store such values.

SP.5 *- Resistance to Opportunistic Service Attacks*

In this case, a malicious smart object can provide good or bad services opportunistically. In our scenario, this attack can be designed as a partial Slandering Attack, in which a smart object acts well inside its community, whereas it acts maliciously when interacting with smart objects of other communities in order to lower the trustworthiness of its community. This could happen during the inter-group probing transactions, when a smart object chosen for the test, say $\overline{t_q}$, returns a corrupted average output. However, thanks to Assumptions **A.1**, **A.2** and **A.3**, since the choice of $\overline{t_q}$ is random, an attacker cannot design a global attack strategy and, hence, it cannot compromise the overall trustworthiness of the community.

SP.6 *- Resistance to Ballot Stuffing Attacks*

In this case, an attacker could boost the reputation of bad objects providing good recommendations for them to increase the chance that they are trusted by the commu-

nity. The countermeasures for this kind of attack fall in the ones described for Slandering Attacks (see Section 15.1.3). Recall that, thanks to the use of the Blockchain technology, no smart object can corrupt or change responses by itself, either positively (in such a way as to increase its trust or reputation scores) or negatively (in such a way as to decrease the trust and reputation scores of other objects).

SP.7 *- Resistance to Denial of Service (DoS) Attacks*

In this case, attackers may cause Denial of Service preventing a reputation system from operating properly due to the flooding of an excessive number of transactions. A particular group of DoS attacks, very common in an IoT scenario, is represented by the Sleep Deprivation Attacks. In this case, the goal of an intruder is to maximize the power consumption of a victim in order to minimize its lifetime.

In general, our approach does not deal with DoS attacks. Hence, the strategies for preventing them are orthogonal to it, and any of these strategies, such as the ones presented in [94, 167, 655], could be adopted. For example, a naive strategy might operate as follows. Whenever a target smart object receives a suspect sequence of consecutive queries from a source one (it can use the communication history to classify anomalous probing activities), it starts to add a random delay in its answers to them. In case the anomalous probing continues over time, the target object stops answering any next query coming from the attacker for a certain time interval.

SP.8 *- Resistance to Orchestrated Attack*

In this case, malicious smart objects orchestrate their actions and leverage several of the previous strategies to perform a coordinated and multi-faced attack, which can change over time. All these types of attacks cannot happen thanks to Assumptions **A.1**, **A.2**, and **A.3**. Hence, an attacker cannot compromise an overall community or even a number of smart objects sufficient to conduct these attacks.

SP.9 *- Resistance to malicious probing exploitation*

In this case, a probing request is made against a node providing invasive services, like critical automation.

First of all, it is worth observing that this kind of device can introduce important critical issues in the considered scenario. In fact, if an adversary gains access to these objects, the consequences of the actions that she/he could make may strongly impact on the safety of the environment. Think, for example, of objects such as smart

kitchen appliances, like smart gas valves or electric cookers. For these devices, probing transactions can lead to dangerous actions if performed with respect to these invasive services. However, our solution does not introduce vulnerabilities that could provide advantages to an attacker who gained access or control of a smart object in the system. In fact, it leverages normal object-to-object communications to implement probing transactions. In this sense, a probing request does not differ from a real one. Consequently, an attacker who chooses to make a probing request through a smart object is not empowered with more functionalities than the ones she/he could obtain in a standard solution without using our approach. However, in this context, the probing strategy could represent a safety risk in itself.

Generally speaking, smart objects can be classified into sensors and actuators. Sensors provide sensing capabilities, measure well-defined physical indicators or collect information on their network and/or possible applications [226]. Actuators perform specific actions based on the inputs received. In our scenario, we are explicitly referring to modern smart objects for IoT. To achieve autonomy, these objects are equipped with both sets of monitoring sensors and a management module that controls object automation services. The probing tests we consider in our approach generally consist of measurements that can be reproduced and compared by means of the other related devices. Therefore, in scenarios characterized by modern smart objects, our solution can be configured in such a way that probing transactions leverage only the sensing capabilities of objects (and not on their capability of performing automation services). Consider, for example, a modern electric cooker. It generally has a management module to control cooking automation (e.g., to turn it on to starting cooking). However, it also has sensing modules, e.g., a module to measure the temperature in order to keep the food at an acceptable temperature with respect to the surrounding environment. In general, using only the sensing capabilities of objects for probing transactions can reduce the risks introduced by critical automation services.

However, in legacy IoT contexts, where several objects can be dummy actuators, our approach could be forced to rely on automation services. Once again we observe that, since probing transactions are based on normal object transactions, they do not introduce additional vulnerabilities. Nevertheless, the vulnerability related to the need to use dummy actuators remains. Consider, for example, the case in which the object to be tested is a legacy smart gas valve. Of course, opening a valve is a critical action and if an attacker were to gain access to this object, a big safety problem could arise. For this reason, it is worth carrying out the probing transactions only in conjunction with normal ones in such a way as not to increase the number of occurrences in which the dummy actuator carries out its actions.

Given this premise, our approach works using the normal interactions between other objects and the dummy actuator to assess the reliability of the latter. In fact, with a configurable probability degree, the querying object will perform the probing task along with the normal transaction. For dummy actuators, a transaction is in any case a request to perform the actions associated with them. In this case, the partition of support nodes, engaged to check the trustworthiness of the queried node, must verify that the action was performed correctly. Therefore, this partition should contain objects that provide sensing services compatible with the action performed by the tested node. For example, consider the case where the action performed by an actuator is switching on a light bulb. In this case, the support partition for probing could consist of smart cameras, smart light detectors, etc.

To cope with this setting, the only necessary change in our strategy concerns the fact that, in Equation 15.1, the value out_j is not the measure returned by the probed node te_j, but the expected variation of a suitable measurable quantity corresponding to the impact of the environment caused by this action. In the previous example, switching on a smart bulb would increase the brightness of the environment related to the total amount of visible light that the bulb is able to emit in the unit of time.

Finally, we observe that, since our probing mechanism is triggered only when a normal transaction is made between a generic object and the dummy actuator, there might be an impact in terms of the time required to collect enough probing results to measure a degradation in the reputation of the dummy actuator. Furthermore, the interaction with the suitable partition of smart objects involved to assess the quality of the action performed by the dummy actuator could involve a larger number of transactions than in the case where a simple measurement sensed by an object is tested. We performed some tests to evaluate these aspects. They are shown in Section 15.2.

15.1.4 Experiments

In this section, we report the experiments we have carried out to test the effectiveness and the performance of our proposal. Specifically, in Subsection 15.1.4, we describe the dataset adopted. In Subsection 15.2, we analyze the performance of our approach. Finally, in Subsection 15.2.1, we compare it with other related ones previously proposed in literature.

Dataset Description. In order to test the effectiveness of our approach we needed both a prototype (that we realized) and a dataset. As real datasets with information about IoT transactions on a two-tier Blockchain do not exist yet, we built a simulator.

To make "concrete" and "realistic" the simulated scenario, we leveraged real-life datasets.

In order to perform our task, we needed two main pieces of information, namely: *(i)* data exchanged among the smart objects of the IoT during a given time interval; *(ii)* data about real Blockchain transactions. We employed a complete online report about IoT data exchanges across several domains, available online at the Zscaler company website[5]. We joined this information with data available from a complete dataset of US Ethereum transactions, obtained at the address `https://console.cloud.google.com/marketplace/details/ethereum/crypto-ethereum-Blockchain?pli=1`.

By proceeding in this way, our final dataset contained information about both the number of transactions performed by IoT smart objects during a month and the actual time required for these transactions to be also stored in a real-life Blockchain. Table **??** shows an example of our dataset. Here, *Source Object* and *Destination Object* are the identifiers of a transaction end-points; *Timestamp* is the time instant in epoch when a transaction took place; finally, *Duration* represents the transaction execution time in seconds.

Source Object	*Destination Object*	*Timestamp*	*Duration*
1	3	1575158400	0.025
2	4	1575163800	0.028
4	6	1575167220	0.022
…	…	…	…

Table 15.2: An example of our dataset

Using the above dataset, we were able to simulate different configurations of our multi-IoT framework. Specifically, we simulated different combinations of smart object communities and object interactions. To measure the impact of probing transactions, as well as smart contract execution times, we built our prototype on top of a real-life public Blockchain. In this way, we had the possibility to experiment probing traffic impact according to our two-tier Blockchain model. In our experiments, we adopted Hyperledger as referring platform.

Figure 15.4 reports the number of ordinary transactions (i.e., those performed to obtain a feature/service and not for probing goals) performed in a month against the community size. The average time necessary to execute all the ordinary transactions of a month against community size is reported in Figure 15.5.

[5] `https://www.zscaler.com/threatlabz/iot-dashboard`

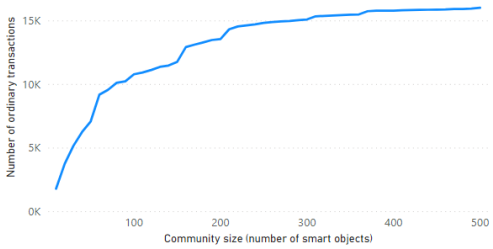


Fig. 15.4: Number of ordinary transactions performed in a month against community size

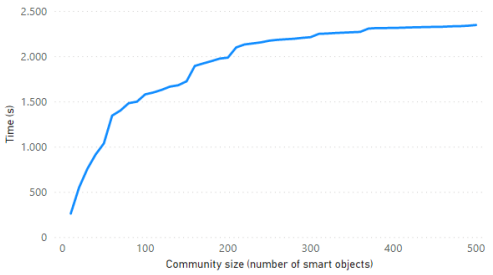


Fig. 15.5: Average time necessary to execute all the ordinary transactions of a month against community size

15.2 Results

Performance analysis of our approach. The first experiment that we carried out was devoted to test the efficiency of our approach. To do so, we focused on the most costly operation, which is the computation of trust values between pairs of smart objects inside communities. We recall that, to create a safe and controlled domain inside each community, smart objects are forced to perform tests on other members of their community randomly selected according to a given probability. We measured the overhead in terms of both the number of generated transactions and the time spent to perform tests. Let p be the probing probability, i.e. the probability for a smart object to generate a test towards another one. We considered a variable size of communities, ranging from 10 to 500 smart objects, and five different values of the probing probability, i.e. $p = 0.1$, $p = 0.2$, $p = 0.3$, $p = 0.4$, and $p = 0.5$. The results obtained are reported in Figures 15.6-15.7.

In these figures, blue lines represent the cost of the ordinary transactions, whereas red lines denote the costs of the probing ones. In more detail, each box corresponds to one of the possible values of p and reports two graphics. The top one compares the execution time of ordinary transactions (in blue) and probing transactions (in red). The bottom one, instead, compares the number of ordinary and probing transactions. This figure suggests that as p increases the overhead introduced by our ap-

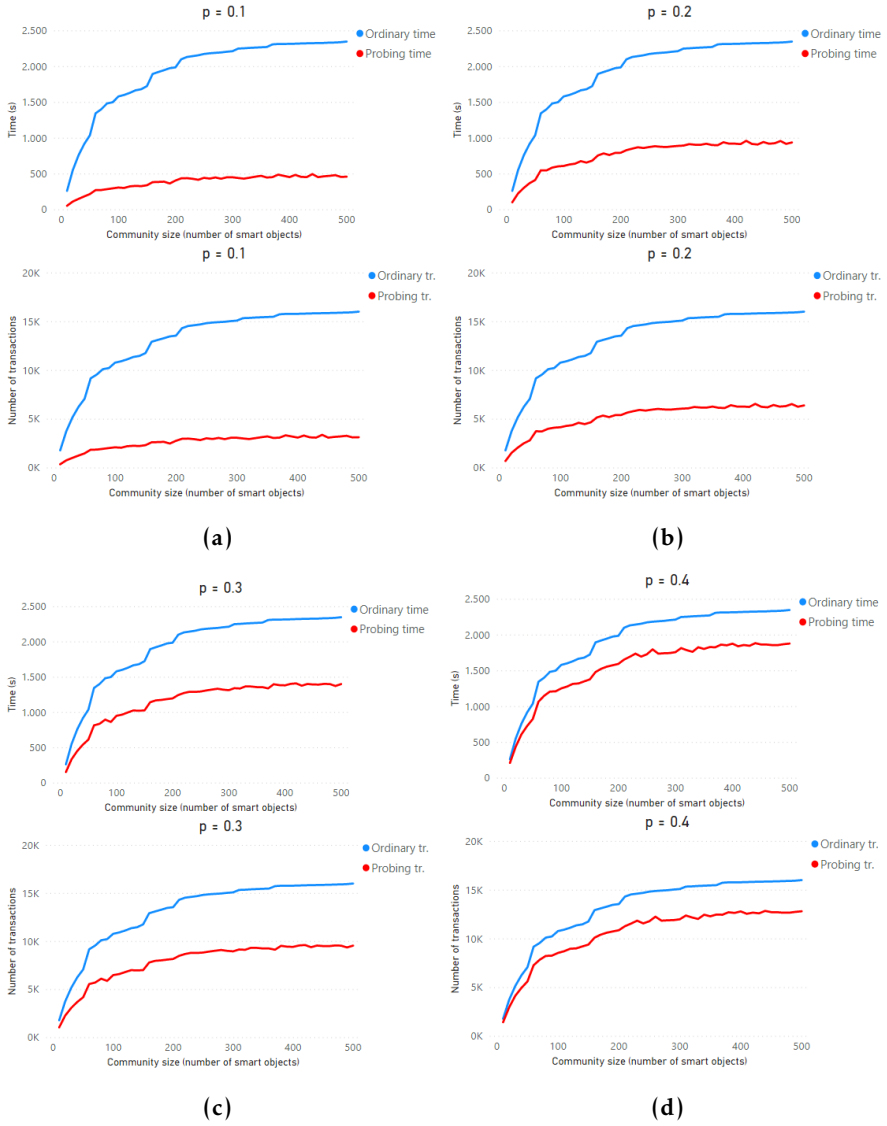
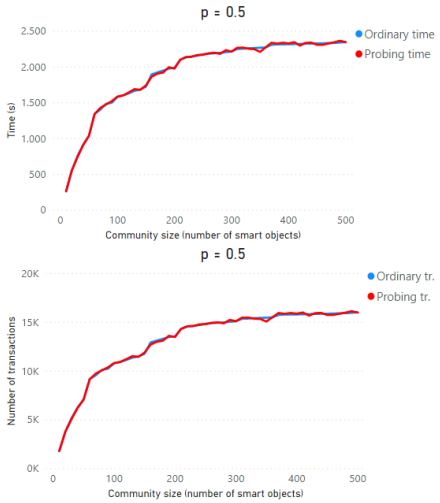


Fig. 15.6: Number of transactions in a month and time necessary to execute them against community size and probing probability - Part I

proach grows linearly reaching the same value as the one of the ordinary transactions when $p = 0.5$; in this last case, the effort to maintain object interaction is doubled.

At this point, to properly tune our framework, we performed a further experiment with the aim of computing the time required by communities to identify (and, hence, remove) an attacked smart object. We carried out this task considering the same probing probabilities analyzed in the previous experiment. Furthermore, we fixed the size of communities to 100 smart objects and we forced our framework to recompute all reputation values after every probing transaction inside a community. Figure 15.8 reports the trend of the reputation decay of an attacked smart object over time. In this figure, each plot corresponds to a value of the probing probability. This



(e)

Fig. 15.7: Number of transactions in a month and time necessary to execute them against community size and probing probability - Part II

figure shows that, as p increases, the reputation decay curve is increasingly steep, as more tests will be executed in a very small time interval. If we assume a value of 0.6 as the minimum reputation for a node to be a member of a group, we can see that the reputation of the attacked node goes under the minimum threshold after less than 4 seconds in all the five plots of Figure 15.8. The lowest time of about 2 seconds is reached for $p = 0.5$.

Now, in Sections 15.1.2 and 15.1.2, we have seen that, in a real world scenario, the propagation of local trust values and, hence, the computation of node reputations cannot be performed continuously. Indeed, this activity implies the activation of a dedicated smart contract requiring computational efforts to Blockchain peers. To avoid this situation, in our approach, we defined a time window tuning the activation frequency of the above smart contract. The objective of this way of proceeding is limiting the activation frequency of the above smart contract, on the one hand, and controlling the dimension of the Local Blockchain (before aggregating all probing transactions and resetting it), on the other hand.

As a consequence, in a real life scenario, the value of the size of the time window should not be too low. In a related study, in which Blockchain transactions are aggregated to control the size of the chain, the time interval for the aggregation is set to 3600 seconds [555]. Of course, we could set the same size, even if, in presence of specific security requirements (i.e., an attacked node must be isolated in less time than an hour), we could reduce it accordingly. Therefore, starting from the value reported in [555], we could use a heuristic based on the Elbow method [344] to reduce this

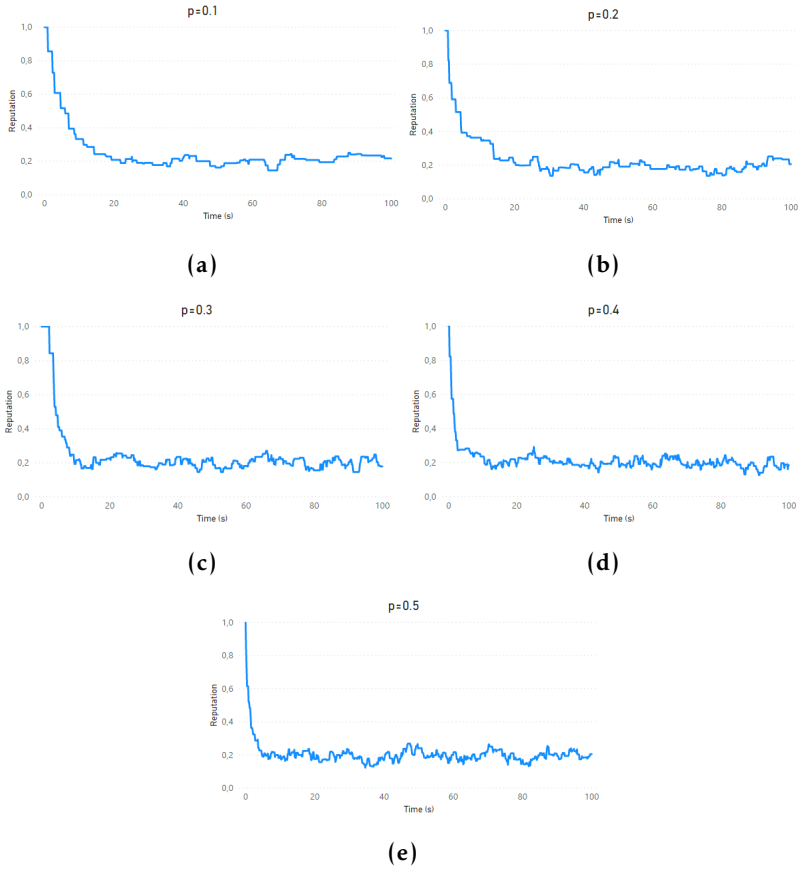


Fig. 15.8: Reputation decay for a malicious smart object inside a community of 100 components

value and the size of the time window in such a way as to satisfy both the requirement on the size of Local Blockchains and the security constraints. Anyway, thanks to the experiment described above, we proved that, in a common IoT scenario, with p set to its lowest value (i.e., $p = 0.1$), only 4 seconds are necessary to collect the probing transactions needed to reduce the reputation of an attacked node and detect it as malicious. Therefore, for any value of the time window size greater than 4 seconds, we can set $p = 0.1$ without losing detection precision. This choice preserves the framework usability, because a negligible overhead will be generated, and still guarantees a satisfactory performance from the security point of view.

As a final experiment, we considered the scenario, described in Section 15.1.3, in which the network also includes legacy devices that provide only automation services (we previously called this type of devices “dummy actuators”). As seen in Section 15.1.3, the presence of dummy actuators could cause an increase in the number of transactions required with objects in the support partition to properly assess the quality of the action performed. Furthermore, this presence could lead to an increase in the overall time required to collect a sufficient number of probing results.

To carry out this experiment, we considered a community consisting of 100 nodes and ran the simulation for the same number of ordinary transactions seen above. In our experiment, we chose different percentages of involved dummy actuators (ranging from 5% to 20%). The results obtained are shown in Figures 15.9 and 15.10.

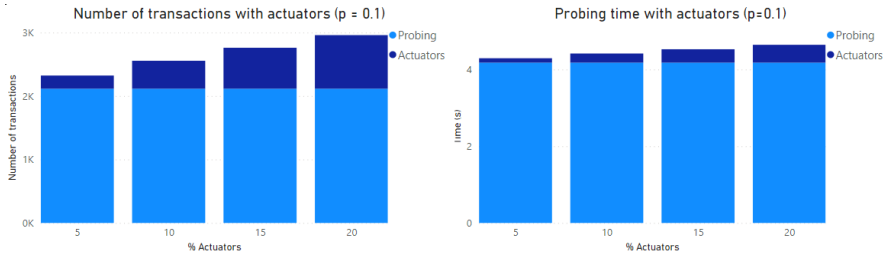


Fig. 15.9: Number of probing transactions and probing time with dummy actuators ($p = 0.1$)

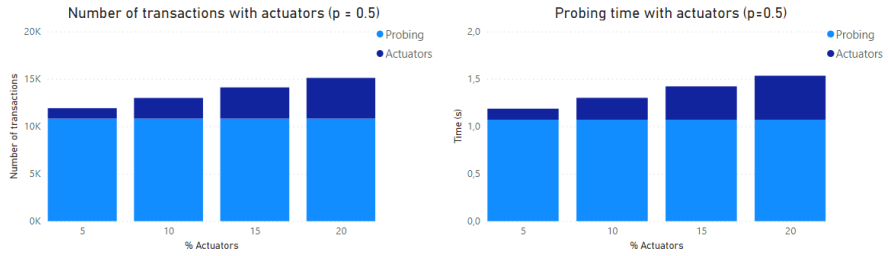


Fig. 15.10: Number of probing transactions and probing time with dummy actuators ($p = 0.5$)

From the analysis of this figure, we can observe that, as we expected, the presence of the dummy actuators leads to an increase in the number of transactions required to complete the probing activities. This increase ranges from 9.78% to 39.88% for $p = 0.1$ and from 10.07% to 41.05% for $p = 0.5$, depending on the percentage of the dummy actuators (the smaller the percentage, the smaller the increase). This increment, although not always negligible, is anyway acceptable, also because the highest values are obtained in correspondence of very high percentages of dummy actuators. We can also observe an increase of the average time necessary to collect the number of probing results needed to reduce the reputation of an attacked node and identify it as malicious. Specifically, the average additional time ranges from 2.63% to 11.24% for $p = 0.1$ and from 11.21% to 42.98% for $p = 0.5$ (again, the smaller the percentage, the smaller the increment). This increase is negligible for $p = 0.1$, which is the configuration suggested by us. It is not negligible for $p = 0.5$, especially in presence of a high percentage of dummy actuators. However, we observe that the

configuration $p = 0.5$ is extreme; it certainly has a theoretical interest but is very far from the one we suggest for real cases (i.e., $p = 0.1$).

15.2.1 Comparison with other approaches

The aim of this experiment is comparing our approach with other related ones proposed in past literature. In particular, we selected two related approaches having different goals but sharing several similarities with ours in both the reference scenario and the adopted methodology.

The first selected approach [29] regards an intrusion detection system useful to protect smart devices in vehicular networks. The main idea proposed by the authors is grouping nodes into "clusters" to identify protected zones where security is achieved with nodes collaboration. Even though the aim of this approach is quite different from ours, they are similar in two aspects, namely: *(i)* the definition of a security model operating on smart devices and IoT, and *(ii)* the usage of groups and clusters of things (corresponding to communities of smart objects in our model).

The second selected approach [470] deals with an orthogonal issue, that is the modeling of a privacy preserving object grouping scheme. This guarantees the protection of user's privacy in all those IoT scenarios where the knowledge of the object features may help an attacker to collect information about user habit and behavior.

In order to compare our approach with the ones of [29] and [470], we measured the communication delay introduced by the evaluated approach against the community size. The communication delay refers to the latency rate introduced by the activation of the evaluated approach in the considered application scenario. Basically, it consists of the increase of the delay in processing and delivering a specific service. In our approach, we defined this parameter as the average difference, in terms of delivery time, between a scenario in which our approach is used and another one where it is not adopted. The results obtained are shown in Figure 15.11.

This figure shows that the average delay introduced by our approach ranges from 20 *ms* to 100 *ms*, whereas the one of the approach of [29] ranges between 24 *ms* and 150 *ms* and the one of the approach of [470] ranges between 22 *ms* and 300 *ms*. This result highlights that the performance of our approach is comparable with, and even better than, the ones of the approaches described in [29] and [470]. Hence, we can state that our approach achieves good results still maintaining the overall IoT overhead to considerable low values.

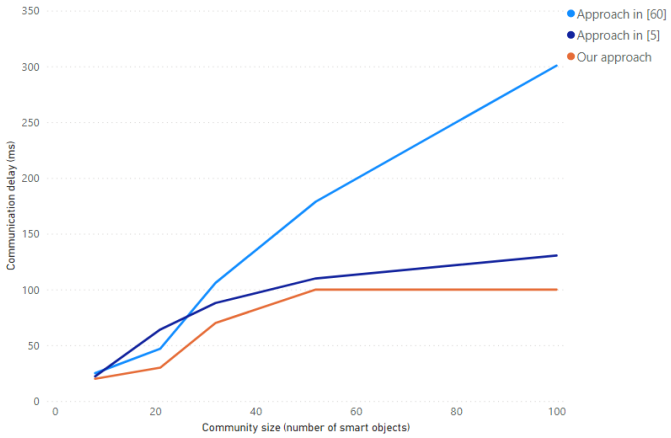


Fig. 15.11: Comparison of the average delay against the community size between our approach and the ones of [29] ad [470]

16

Extending saliency maps and gaze prediction in an Industry 4.0 scenario

In recent years, researches dealing with the study of visual attention have become very popular thanks to the enormous increase of Artificial Intelligence. Machine Learning and, in particular, Deep Learning allowed researchers to propose new predictive models operating on natural images. In the meantime, an increasing number of websites has been made available on the Internet. However, few approaches, aiming at extending the results obtained on natural images to web pages, have been proposed. In this chapter, we provide a contribution in this setting by applying fine-tuning and other refinements to two existing GAN-based approaches (i.e., SalGAN and PathGAN) originally proposed to predict the saliency maps and gaze paths on natural images. Our ultimate goal is defining some variants of them able to deal with websites. In particular, our SalGAN variant represents one of the first attempts to employ GANs for saliency map prediction on web pages, whereas our PathGAN variant is the first attempt to adopt GANs for gaze path prediction on websites. Here, we present our proposals, highlight their main novelties, describe the tests done and the results obtained. We also highlight two further contributions of this paper, namely: (i) a new dataset, more complete than the existing ones, supporting the analysis of visual attention on websites, and (ii) a tool supporting a web page designer in her attempt to increase the visitor interest and curiosity.

The material presented in this chapter was derived from [220].

16.1 Methods

16.1.1 Improving SalGAN to derive saliency maps for web pages

As pointed out in the Introduction, in order to derive saliency maps for web pages, we started from SalGAN [488], because this approach has proven to be the most accurate in the prediction of saliency maps for natural images. Then, we performed several adjustments to make it more suitable to operate on websites and to return accurate results.

Here, we feel important pointing out that, during our research, we also started from TSGAN [397] and tried various refinements on it, in order to improve its performance. As pointed out in Section **??**, to the best of our knowledge, TSGAN is the only already existing GAN-based approach to predict saliency maps of users on websites. Actually, all our attempts to obtain improved versions of TSGAN have been unsuccessful. Therefore, as we will see in Section 16.2.3, in conducting the test campaign for evaluating the performance of our SalGAN variants when they are applied on websites, we compared them to the original TSGAN and not to our proposed variants.

We started by investigating how SalGAN behaves when it is directly applied on websites. The architecture of SalGAN is reported in Figure 16.1. The generator is a simple single stage autoencoder that generates saliency maps from input images. The discriminator receives both generated and real images and must identify which of them are coming from the real data distribution. The SalGAN loss is built around the standard GAN loss function, customized to obtain better results on images. Authors have also published pre-trained weights of their network on a dataset made of natural images.

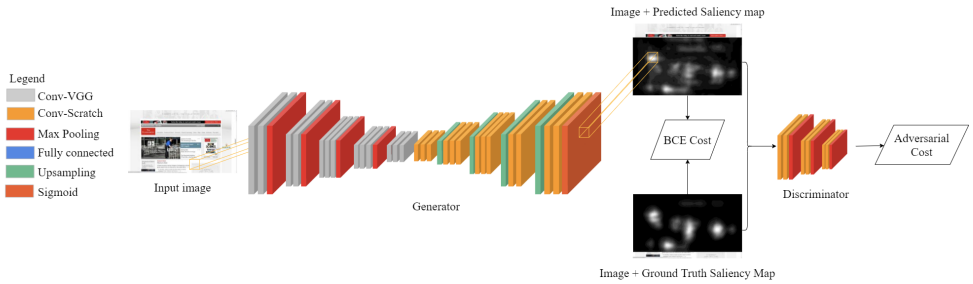


Fig. 16.1: The architecture of SALGAN

In order to apply SalGAN to web pages, we focused on fine-tuning this model in the best possible way, leaving most of the network structure unchanged.

We did not consider necessary the change of the architecture, as it already performs very well on real images. Furthermore, we left the structure of the loss unchanged, with one part measuring the content loss and the other one measuring the adversarial loss [488].

The main fixed point we had was to keep frozen the layers referring to the encoder inside the generator. In fact, in this part of the network, the authors of [488] use the pre-trained VGG network structure because it has been shown to accelerate the convergence of the model. TSGAN also uses this approach, which greatly reduces the amount of time required to obtain a good training. Instead, we have considered

necessary a complete re-training of the second part of the generator, i.e., the decoder, where the saliency map is generated and adapted as much as possible to the web page domain.

The reasoning underlying our choice of freezing only one part of the generator concerns the different goals of the two parts. The first part is responsible for recognizing objects in the input image. It already obtains very satisfactory results with the pre-trained configuration. Therefore, we decided to keep it unchanged. The second part has the goal to create the saliency map. Since the generator of [488] is trained only on natural images, it can create saliency maps suitable for them, while it makes several faults in performing predictions in an artificial domain, such as the one regarding websites. For this reason, the second part of the generator needs to be trained again so that it can learn how to create saliency maps for both natural images and web pages.

In addition to the first part of the generator, we also decided to freeze the first four convolutional layers of the discriminator. The reason for this choice is similar to the previous one. In particular, we need to freeze the first layers of both the generator and the discriminator in order to maintain the right level of competitiveness between these two neural networks, which is crucial to get fine results from a GAN architecture. For example, training the discriminator from scratch implies that all the weights obtained from the natural image dataset must be recomputed, which would lead the discriminator to overfit on the web page domain, where it would train very quickly, being this set small and very specific.

Instead, the choice to freeze the first four layers of the discriminator keeps its ability to distinguish between real and fake saliency maps almost completely intact. In fact, if we compare two saliency maps, one coming from the natural image domain and one coming from the web page domain, it should be difficult to determine which comes from one domain instead of from the other. In their own right, saliency maps from these two different domains can be considered similar because their structure does not present remarkable differences. The features that the discriminator has learned as determinant for asserting the quality of a saliency map are common in both domains. This is the reason for which it is important to preserve what the network has learned previously, avoiding the training of these first levels. With this choice, training improves the quality of the saliency maps produced thanks to a discriminator with a lot of "experience".

In Figure 16.2, we can see the layers of the network that have been frozen. We obtained the optimal number of layers to keep out of training after making preliminary observations on the quality of generated images. Deriving the optimal number of layers to freeze implies a trade-off on how many layers we should train on our

dataset and how many layers we should keep with the same weights provided by [488]. For the sake of space, we are not reporting all the experiments we made, but the reasoning underlying our choice. Indeed, as we pointed out before, the first layers of both generator and discriminator are devoted to extract features from the input image, while the next ones are used for creating a saliency map, and detecting if the input saliency map is real or fake, respectively. In this perspective, if we freeze more layers than the optimal solution we found, the resulting SalGAN would not be able to adapt to the web domain, since there are few layers to train on our dataset. On the other hand, if we train more layers than the optimal solution, we both lose the training weights of [488] and overfit the resulting SalGAN to the web pages layout, thereby taking away the capability to perform well with natural images.

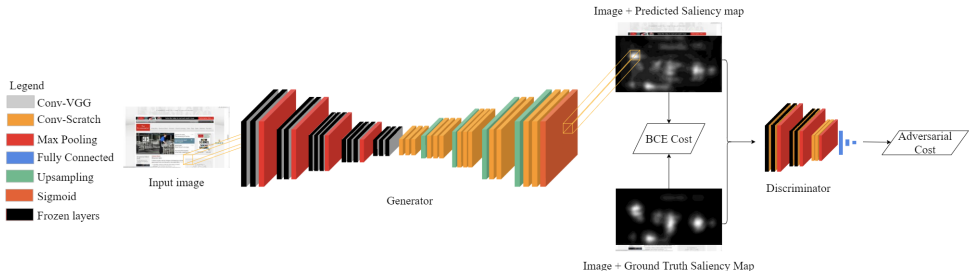


Fig. 16.2: SalGAN frozen layers during training

Furthermore, we decided to lower the learning rate of the neural network from $3 \cdot 10^{-4}$ to $1 \cdot 10^{-4}$. This allows a more gradual, but smoother and more stable, convergence to the optimal solution. A higher learning rate could allow a faster convergence to the optimal solution, but this would be done with the presence of "ups and downs" of the loss function before it reaches the possible convergence.

As a final remark, we point out that both the two refinements mentioned above are necessary to adapt SalGAN to the web layout domain. In fact, assume that we change only the learning rate parameter, without freezing any layers. The network weights provided in [488] must be recomputed. To perform this task, we should train the whole SalGAN from scratch, which is a huge time-consuming task. Actually, we need to preserve the SalGAN's capability of working with both natural images and web layouts, because web pages could contain several natural images. Therefore, we must freeze the first layers of both generator and discriminator. This implies that we should keep the weights of the pre-trained networks.

On the other hand, assume that we keep frozen the first layers of both generator and discriminator and do not modify the learning rate parameter. This leads to an unsuitable scenario. In fact, maintaining the previous learning rate means train-

ing SalGAN too quickly, which makes the weight tuning unstable for many epochs, eventually resulting in mode collapse or unstable training.

As a conclusion to our reasoning regarding fine-tuning operations, in Figure 16.3, we report a qualitative visualization of the predictions returned by the original and fine-tuned SalGAN. Specifically, we report in this figure some examples of the results returned by these two models. From the analysis of it we can see that the saliency areas returned by the fine-tuned SalGAN are more in line with the ground truths. In fact, the original SalGAN identifies saliency areas that are not present in the ground truths, and does not report some saliency areas present in the ground truths. This is much less the case for the fine-tuned SalGAN, whose predictions are much closer to the ground truths than the original SalGAN.

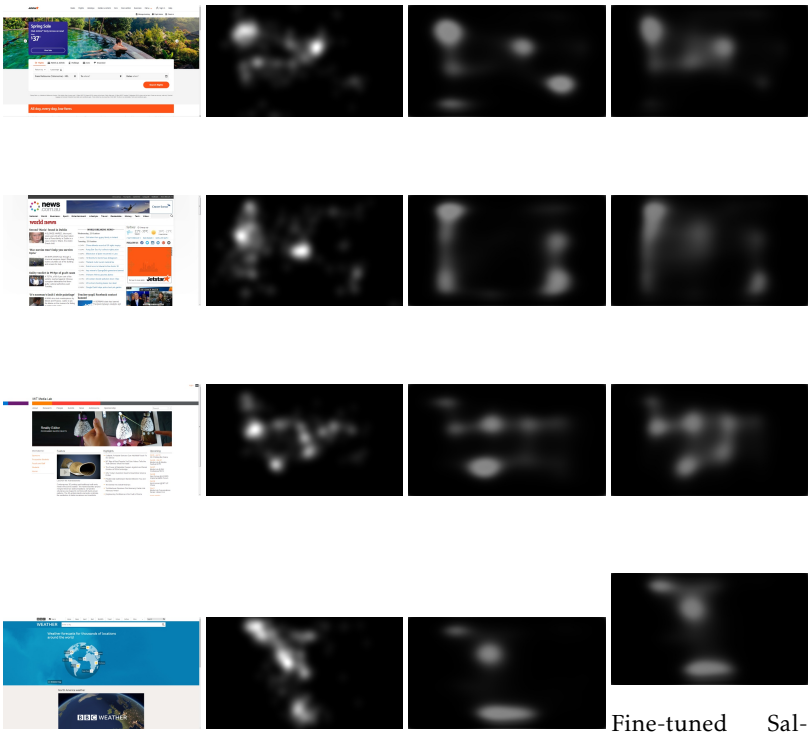


Fig. 16.3: Qualitative comparison between the predictions of the original and fine-tuned SalGAN

The images shown in Figure 16.3 represent only qualitative examples of the potential of fine-tuned SalGAN. Beside a qualitative evaluation, it is important to quantitatively verify the possible benefits brought by the fine-tuning procedure. To this end, in Section 16.2, we present the results of several tests showing that fine-tuned SalGAN achieves better results than other models.

16.1.2 Improving PathGAN to derive gaze path predictions for web pages

In the previous sections, we defined an approach to derive saliency maps for web pages. However, saliency maps are not sufficient to understand the order in which the elements are seen by a user. In fact, unlike natural images, the layout of elements in a web page affects its ability to capture the user's attention. This is the main reason why we have defined an approach for estimating the gaze path in a web page. Indeed, there are several practical applications, which are really difficult (or even impossible) to perform with saliency maps alone. Some examples of such applications are:

- understanding how user behavior is affected by different web page layouts;
- finding the best priority order for the elements of a web page;
- performing automatic A/B testing on different layouts.

In the field of eye movement prediction, scientific research did not achieve many important results yet. The lack of annotated datasets makes the creation of new models not easy: no data very often leads to the impossibility of performing a successful training. All the solutions proposed so far apply to the prediction of gaze on natural images. Among the limited studies in this field, a Generative Adversarial Network, called PathGAN [43], stands out for its results. PathGAN predicts the visual scanpath of people observing images, both in normal and in 360 degrees format. The quality of its results places it as one of the best performing models in this domain. In Figure 16.4, we report the architecture of PathGAN. The first part of the network is a generator; an input image is fed to obtain a gaze path, which represents the route of the eyes of a potential user observing that specific image. The generated path is a sequence of 63 fixations, each consisting of a tuple of four elements: a x-coordinate, a y-coordinate, a timestamp and an end of path probability. This last element allows the generation of paths of variable length; a threshold is set on it to determine which fixations should not be included in the final prediction. As expected, the first three values of each tuple include information on both position and duration of every fixations.

The first tests with the network did not give encouraging results in the GUI domain. We identified several problems that needed to be resolved to improve performances. First of all, the generator and discriminator weights are not updated with the same frequency. The choice to make more updates on the discriminator, rather than on the generator, did not lead to performance improvements. Moreover, assigning a very low weight to the content loss ($\alpha = 0.05$) makes the discriminator very strong, compared to the generator. Training the generator for the first 5 epochs alone was still not sufficient to prevent this phenomenon. In addition, the number of

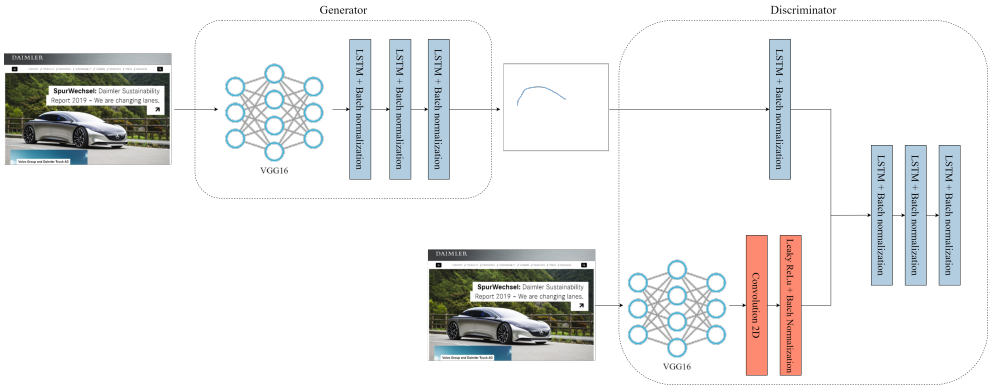


Fig. 16.4: The architecture of the original PathGAN

values to be predicted complicated the problem too much. In order to allow the prediction of paths of variable length, the last element of each fixation tuple is an end of path probability. We noticed that the training did not manage this extra variable adequately, resulting in either very short or very long paths. Overall we experienced completely wrong predictions that, in the long run, during the training, led to mode collapse. In Figure 16.5, we reported two examples of mode collapse. In this figure, yellow lines and squares represent the original PathGAN predictions, while blue lines and red squares denote the ground truth.

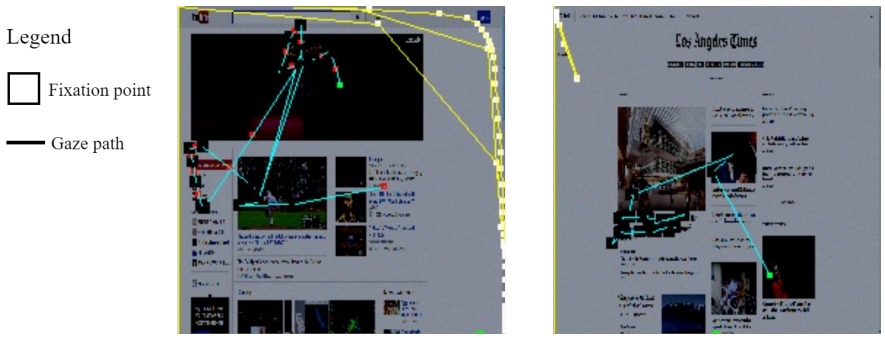


Fig. 16.5: Two examples of mode collapse

During mode collapse the generator tends to predict outputs that match the edge of the image, completely ignoring the original ground truth. The triggering cause of this phenomenon was identified in the combination of several elements. The discriminator becomes too strong, compared to the generator, which can no longer make realistic predictions. The lack of data does not help in this regard. The discriminator clearly overfits on the training data after several epochs. Changing the number of times the generator and discriminator weights are updated does not prevent this.

An explanatory graph of this phenomenon is visible in Figure 16.6. As shown in this figure, the generator (blue line) is very strong in the early training stages because it was trained alone for the first five epochs. In the first steps, the discriminator loss starts to decrease gradually. It suddenly experiences a huge drop that matches with a degradation of the generator's loss score. From that point onward, the predictions start to be totally wrong. It is clear that the network needs some adjustments before being applied to our domain.

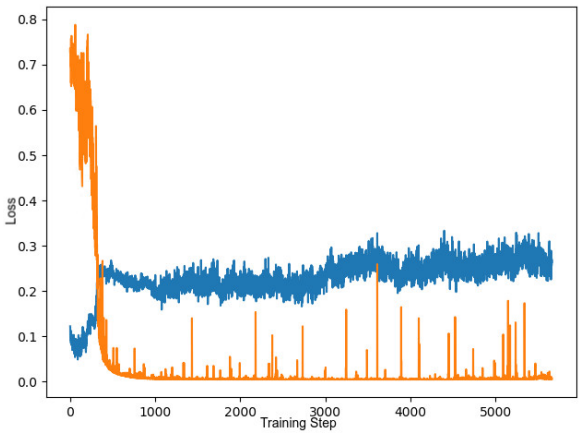


Fig. 16.6: Generator (blue) and discriminator (orange) loss after that the discriminator overfits the training set

All the improvements we have introduced to the network strive to mitigate the problems described above. Since we did not have a dataset as large as the one used by the authors of PathGAN, we chose to reduce the complexity of the problem. Therefore, instead of predicting a sequence of tuples of four elements (x-coordinate, y-coordinate, timestamp, end of path probability), we chose to remove a variable. Our goal was to predict paths of different lengths with only three variables, instead of four. We removed the end of path probability to force the network to predict paths of the same length. However, since not all the paths of our dataset are of the same length, we opted to introduce dummy nodes on the arcs connecting two fixations. This solution makes sure that also our dataset has 63 fixation long paths. The dummy nodes have been added on the arcs by applying linear interpolation. Since, in reality, they do not correspond to a fixation, the timestamp is increased by a negligible constant. This allows us to distinguish which are the real fixations and the fictitious ones. After post-processing predictions, we are able to generate a sequence of real fixations. Each timestamp is transformed into a duration, allowing us to better distinguish the real fixations from the dummy ones. The predicted fixations with a duration below a threshold are considered dummy and, therefore, removed from the

path. This procedure preserves the generation of paths of variable length without the need of the end of path probability.

As a result of this modification, we obtained a PathGAN model having a different output structure from the original one. In fact, this new version can generate a gaze path of variable length, in which the duration of human fixation points is in line with the literature [329], and there is no need for the end of path probability. This was already an important improvement because the output path is consistent with a real scenario. However, we believed that this is not enough because it produces changes only in the structure of the model outcome.

The next step was to introduce improvements in the way the network is trained. At the same time, we also changed the weight assigned to content and adversarial loss. The only way to make the discriminator weaker is to update the generator weights more often.

At first we tried to keep the weights assigned to the various parts of the loss unchanged. Initially, we tried to tune the number of weight updates for every training step of both the generator and the discriminator. After several attempts, we realized that we were just postponing the moment when the discriminator would overfit. Therefore, it proved essential to also modify the weights of the various parts of the loss. We saw positive effects when we decreased the weight given to the adversarial loss within the objective function. A higher content loss weight prevented the discriminator from taking over. The quality of the samples generated increased dramatically, allowing network training to be completed successfully. In conclusion, we decided to multiply the adversarial loss by a constant equal to 0.35, and the content loss by a constant equal to 1. Moreover, we found the right number of weight updates for every part of the network; specifically, we decided to update 16 times the generator weights every step, limiting the discriminator to only 4 times.

We also introduced other modifications aimed at avoiding overfitting. We took our cue from saliency prediction models and added noise to the images passed to the discriminator. Also in this case, the qualitative evaluations of the output, together with the loss trend, were fundamental; in fact, we could immediately detect any overfitting and mode collapse. We undertook further attempts to improve the results, but without success. For example, we tried to modify the network to receive a saliency map input. Both the generator and the discriminator should have benefited from this modification, because there is a match between saliency map and path. Instead, we noticed that there were no tangible benefits; on the other side, the complexity of the architecture increased. We also tried to modify the path preprocessing; in particular, instead of adding dummy nodes on the arcs, we tried to superimpose them on existing nodes. This should have brought more precision in

predicting fixations. Again, we noticed no improvement. We believe that further attempts can be made using analogous techniques in the future. After all these changes and improvements to the original PathGAN, we obtained a new version of it, which we called NormalGAN. This has the same architecture as PathGAN. However, thanks to the changes explained above, it is able to deal with the web page scenario.

Beside this first version of improved and fine-tuned PathGAN, we designed a second one. In this new version, we started from the considerations that had led us to define NormalGAN and flanked them with additional considerations that prompted us to make further changes. In particular, we modified the network making it to follow the structure of a conditional Wasserstein GAN [288] (we call it WGAN in the following). We also modified the training process to respect the characteristics of a WGAN. Moreover, we updated the weights of the generator and the discriminator with different frequencies. In particular, the discriminator was updated more often because weight clipping was introduced. The discriminator was updated 5 times, while the generator was updated only once. We also reset the weights of the various terms of the loss to their original values; in particular, we set content loss to 0.05 and adversarial loss to 1.

Setting the update rate of the weights and the constant that multiplies the loss function allowed us to achieve a balance between the strength of the generator and the discriminator. In fact, increasing the update rate of the weights leads to a scenario where the generator and/or the discriminator learn too much from our dataset, which causes overfitting. On the other hand, decreasing the update rate of the weights implies that the training process takes longer or that, for the same duration, the generator and/or the discriminator cannot learn enough from the dataset. The constants that multiply the loss functions are even more important because they tune the balance between the generator and the discriminator. Recall that, in a GAN scenario, both the generator and the discriminator learn from the other's errors. This process requires the right amount of time. For example, if we increase the constant that multiplies adversarial loss, the discriminator will have much more power than the generator, which means it would be able to discriminate real paths from fake ones without giving the generator the time necessary to acquire enough information from that and react appropriately. By contrast, decreasing the weight of the adversarial loss leads to a weak discriminator, which is fooled by the generator. A similar reasoning can be made for the constant that multiplies the content loss. In Table 16.1, we summarize these considerations, along with the corresponding ones related to the setting of all the other parameters involved in the two variants of PathGAN.

The improvements we made to the original PathGAN can also be observed by analyzing the loss values of the generator and discriminator of WGAN, shown in

	Parameter	*Our solution*	*Lower values*	*Higher values*
NormalGAN	Constant multiplying the adversarial loss	0.35	The Generator overcomes the Discriminator	The Discriminator overcomes the Generator
	Constant multiplying the content loss	1	The Discriminator overcomes the Generator	Not feasible
	Update frequency of the generator weights	16	Generator underfitted	Generator overfitted
	Update frequency of the discriminator weights	4	Discriminator underfitted	Discriminator overfitted
WGAN	Constant multiplying the adversarial loss	1	The Generator overcomes the Discriminator	Not feasible
	Constant multiplying the content loss	0.05	Not feasible	The Generator overcomes the Discriminator
	Update frequency of the generator weights	1	Generator underfitted	Generator overfitted
	Update frequency of the discriminator weights	5	Discriminator underfitted	Discriminator overfitted

Table 16.1: Overview of the parameters of our PathGAN versions

Figure 16.7. From the analysis of this figure, we can see that the discriminator does not overfit, as previously happened in Figure 16.6, because the loss values do not increase after some training steps. Furthermore, the loss values of the generator decrease rapidly and, therefore, reach an equilibrium point. This means that it has the time to learn how to create good saliency maps.

We report the WGAN architecture in Figure 16.8. As we can see, it is similar to the PathGAN architecture (and also to the NormalGAN one, because PathGAN and NormalGAN share the same architecture). However, unlike the latter, it does not include the batch normalization components.

As a first qualitative result of our work, we tested the obtained WGAN on the images that led the original PathGAN to collapse. In Figure 16.9, we report the results obtained. In it, colored lines represent the gaze paths. The image on the left shows

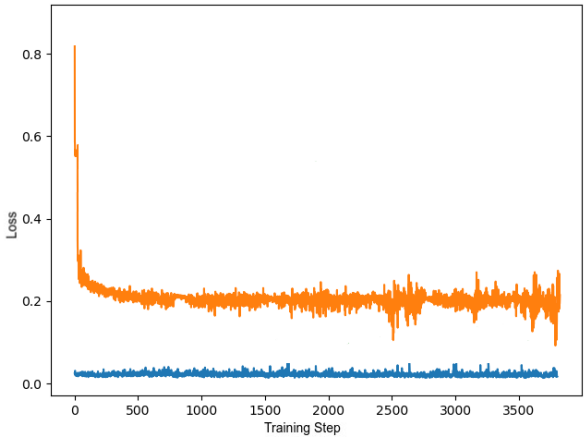


Fig. 16.7: Loss values of the generator (blue) and discriminator (orange) of WGAN

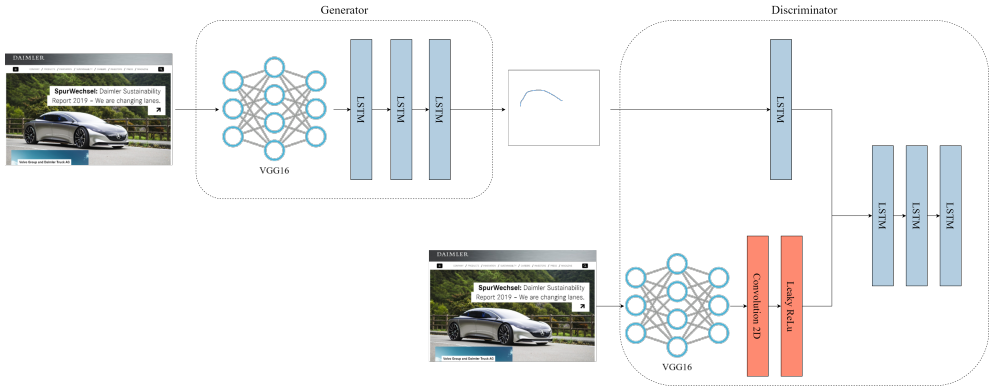


Fig. 16.8: The architecture of WGAN

the ground truth, while the image on the right shows the prediction of WGAN. As can be easily seen, our model does not suffer from mode collapse and predicts gaze paths comparable with those of the ground truths.

Finally, in Table 16.2, we provide a summarization of the differences between the original PathGAN and the two modified versions that we are proposing in this paper.

Original PathGAN	*NormalGAN*	*WGAN*
Best results with natural images	Fine-tuned for the GUI domain	Fine-tuned for the GUI domain
End of path probability	Fixed path length	Fixed path length
Content loss weight equal to 1	Content loss weight equal to 1	Content loss weight equal to 0.05
Adversarial loss equal to 0.2	Adversarial loss equal to 0.2	Adversarial loss equal to 1
Conditional GAN	Conditional GAN	Conditional Wasserstein GAN

Table 16.2: Differences between the original PathGAN and our proposed variants (i.e., NormalGAN and WGAN)

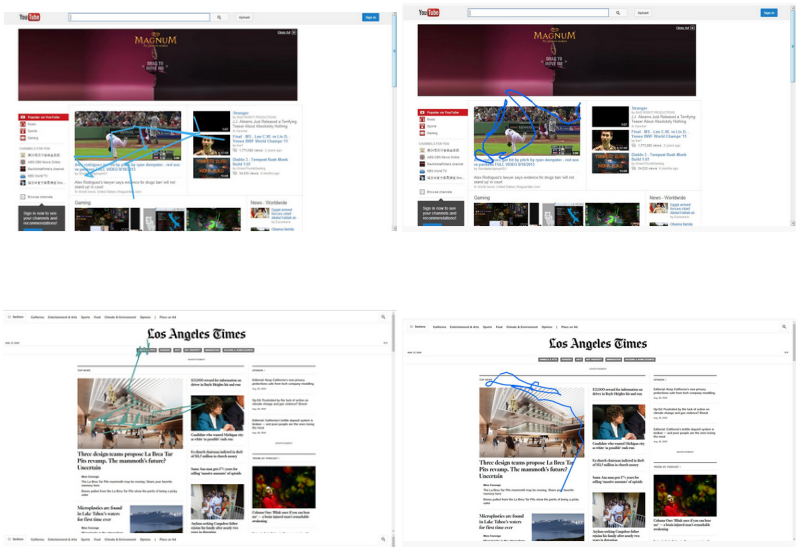


Fig. 16.9: Ground truth (on the left) and WGAN prediction (on the right) of images that had led the original PathGAN to mode collapse

16.2 Results

16.2.1 Saliency map and gaze path prediction tool

In this section, we illustrate the tool implementing our approaches for the generation of saliency maps and gaze paths of users accessing websites. We strived to build a usable and efficient tool, which can be easily employed by any user (for instance, a designer), who wants to evaluate the effectiveness of a web interface. Our tool consists of a web application. We developed it in Python; in particular, we implemented the neural network-based algorithms representing the core of our approach using the well-known Keras and Tensorflow Python libraries. Moreover, we used Django as the core of our web application.

Having in mind the need to guarantee the best possible User Experience, we created a home page where a user can upload an image and specify if she desires to evaluate the saliency map or the gaze path for that image.

In Figure 16.10 (resp., 16.11), we report an example of the output provided by our tool for saliency map (resp., gaze path) prediction.

16.2.2 Dataset description

To the best of our knowledge, only one dataset containing both web page layout images and gaze data is available in scientific literature. This dataset, called FiWI (Fixations in Webpage Images) [563] is used in all researches concerning saliency map and gaze path prediction in the GUI domain. It was built by collecting data from 11 volunteers, each observing 149 websites. The limited number of volunteers involved in

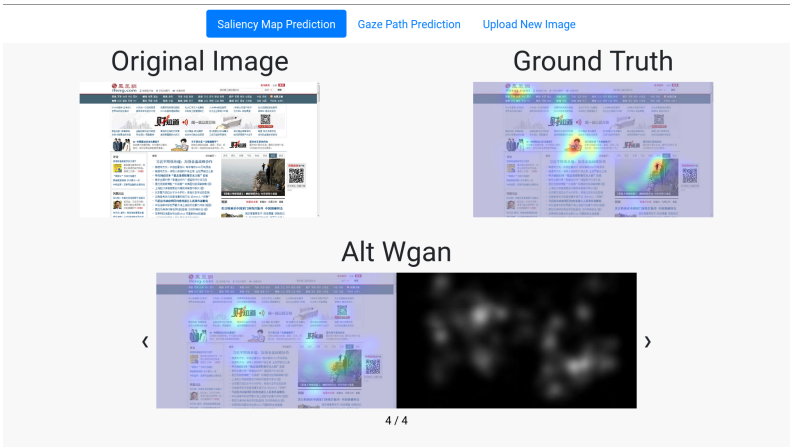


Fig. 16.10: An example of a saliency map prediction returned by our tool

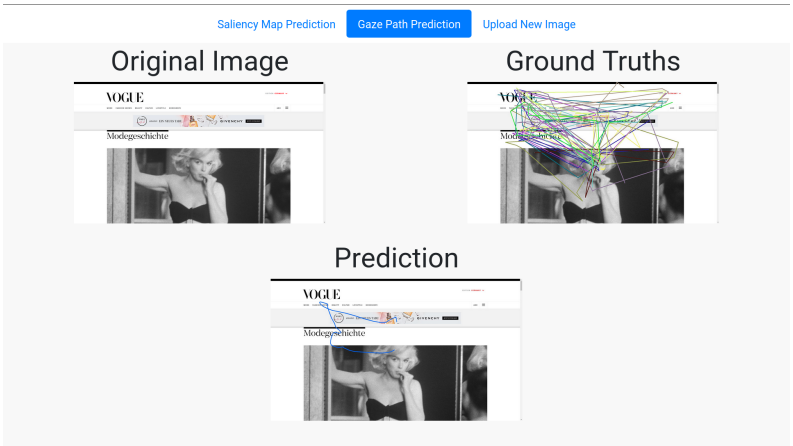


Fig. 16.11: An example of a gaze path prediction returned by our tool

data collection makes FiWI incapable of completely enclosing all the ways in which human beings observe images. Furthermore, the user gender is not balanced in it, because 7 volunteers were women and 4 volunteers were men; this fact could introduce a gender bias. The data gathering procedure used for this dataset was very intensive because each volunteer was required to observe numerous datasets, each for 5 seconds, generating a considerable amount of stress to her/him. Furthermore, data was collected in an unnatural way, which could prevent the creation of a realistic model. In fact, volunteers were placed with their chin on a head rest, in a dark room, 60 cm away from the screen. Finally, the set of images of the dataset comes from the same time period. With regard to this aspect, we observe that web pages are subject to changes of style guidelines over time; for this reason, today's web pages are very different from the layouts present in the original FiWI dataset. This fact introduces a bias related to the evolution of the page design techniques over time.

All these considerations led us to build a new dataset aiming at avoiding, or at least mitigating, these biases.

Since the beginning, we thought it was necessary to increase the total number of images present in the dataset. We added new layouts to the ones already considered in FiWI, as they are useful to make the results as general as possible.

The websites composing FiWI belong to three different classes, namely: *(i) Pictorial*, in which case the pages are occupied by a dominant picture, or several thumbnail pictures and little text (e.g., photo sharing websites); *(ii) Text*, whose pages contain high-density informational text (e.g., Wikipedia); and *(iii) Mixed*, whose pages present a mix of thumbnail pictures and text (e.g., social network sites). In our dataset, these classes have been extended while keeping balanced the fraction of websites belonging to each of them.

Furthermore, in our opinion, the three classes of websites were not fully representative of the whole variety of the World Wide Web. For this reason, we added a fourth class consisting of a set of *Business* websites, presenting analytical layouts (in particular, dashboards) and layouts of the Daimler intranet. Analytical dashboards and web pages of an intranet are very different from traditional websites, because they are not designed for ordinary web surfers.

Finally, we considered that the design principles used in the creation of websites change over time. For this reason, we decided to add in the dataset the updated layouts of the pages already present in FiWI. As a last task, we dropped from the final dataset the FiWI websites without an updated version available (e.g., web pages of companies that no longer exist) in order to obtain a balanced set of old and new layouts. Starting from this original core of 149 images, we arrived at a total of 262 web layouts.

Furthermore, it was necessary to consider more people than the FiWI dataset; for this reason, we collected data from 100 volunteers to include more nuances of how different people look at images. The need to have more testers made us focus also on who are the potential users of the websites under consideration. We felt that collected data should come from both an enterprise and a more general environment. For this reason, 25% of the data collected come from employees of the Daimler AG, who are used to work with websites. Also the gender of enrolled volunteers was perfectly balanced. Due to time limitations and the difficulty to recruit old people, we were not able to balance the dataset by age groups, making it slightly unbalanced towards younger people. However, compared to FiWI, we have a better representation of all age groups. In particular, the age of volunteers ranges from 15 to 70 years old.

Since most of the time people navigate the web in uncontrolled environments, we chose to respect this principle also during data collection. Our data gathering activity allowed volunteers to stay in a comfortable position and made them more willing to participate to the test. This also granted us to collect more natural data, compared to a "laboratory" situation, like the one used in FiWI. In fact, we argue that a controlled environment can cause a different user behavior. We employed a laptop connected to an eyetracker fixed to the base of the display and placed on a horizontal plane (e.g., a desk or a table) in front of the volunteer. The screen distance was variable according to the eyetracker's ability to correctly detect the eyes of the volunteer. After explaining the task to the volunteer, we carried out a quick calibration of the device, assuring a high quality of gathered data. Then, we started the data collection procedure, with an estimated duration of about 3 minutes. Each image appeared on the volunteer's screen for 4 seconds. Images were interspersed with a black screen with a central white dot to allow the volunteer to rest her/his eyes. When ready, she/he could press the space bar to continue with the next image. In total, every volunteer observed 30 web page layouts extracted from the image dataset. Our algorithm selected the images to display in such a way as to keep balanced the number of volunteers who observed each page.

In our dataset, each path consists of a sequence of fixation points, associated with a volunteer and an observed image. For each captured fixation point, the x and y coordinates, along with the timestamp it was observed, were recorded. Each of the 262 images was seen by 11 or 12 volunteers. Since each volunteer observed 30 images, our dataset stores a total of 3000 gaze paths. If compared with the FiWI dataset, the number of available paths is more than twice, providing us with a solid base for training the path prediction model.

In Table 16.3, we report several information allowing a comparison between our dataset and FiWI.

	FiWI [563]	*Our dataset*
Number of subjects	11 (4 males, 7 females)	100 (50 males, 50 females)
Age range of subjects	21 - 25	15 - 70
Number of web pages	149	262
Time necessary to display a web page	5 seconds	5 seconds
Screen resolution	1360×768	1920×1080
Number of gaze paths	1,639	3,000

Table 16.3: Comparison of several characteristics of FiWI and our dataset

16.2.3 Experiment Results

Saliency map prediction. As for the saliency map prediction, we adopted several metrics, which have been largely employed in the past literature. They are:

- *Normalized Scanpath Saliency* (hereafter, NSS) [500]; it ranges in the real interval $[0, +\infty)$.
- *AUC-Judd* [523]; it ranges in the real interval $[0, 1]$.
- *AUC-Borji* [92]; it ranges in the real interval $[0, 1]$.
- *Pearson Correlation Coefficient* (hereafter, CC) [439]; it ranges in the real interval $[-1, 1]$.
- *Kullback-Leibler divergence* (hereafter, KL) [230]; it ranges in the real interval $[0, +\infty)$.

For the first four metrics, the higher their value, the better the quality of the approach into evaluation. Instead, as for KL, the lower its value, the better the approximation of the ground truth by the saliency map.

We compared the different SalGAN variants we have proposed in Section 16.1.1 to verify if at least one of them provided better results than the original SalGAN. In particular, we evaluated four SalGAN models. The first (hereafter, Reference) is the original SalGAN using pre-trained weights on natural images. The second (hereafter, FineTuned) is the SalGAN that we fine-tuned by ourselves. The third (hereafter, KeepTrain) is the SalGAN that we kept trained, using our dataset, without any fine-tuning. The fourth (hereafter, FromScratch) is the SalGAN that we completely re-trained with our dataset. In Table 16.4, we show the metric values for the four models. We also report the values of the same metrics for the original TSGAN.

	NSS	*AUC-Judd*	*AUC-Borji*	*CC*	*KL*
TSGAN	1.43	0.82	0.76	0.66	0.63
Reference SalGAN	1.25	0.80	0.76	0.56	0.90
FineTuned SalGAN	1.61	0.85	0.82	0.74	0.52
KeepTrain SalGAN	1.58	0.84	0.83	0.73	0.52
FromScratch SalGAN	1.49	0.83	0.80	0.68	0.65

Table 16.4: Values of the adopted evaluation metrics obtained for the original SalGAN, the three variants of this network proposed in this paper and TSGAN

This table highlights that the worst performing model is the original SalGAN. This can be easily explained considering that, as SalGAN was previously trained only on natural images, the domain change causes a significant performance drop.

Instead, our three SalGAN variants prove to have a great ability to predict saliency maps. All of them have similar or higher metric values than TSGAN. Overall, we observe a considerable superiority of the model that has undergone fine-tuning. For some metrics, the performance is also superior to the one achieved by SalGAN for natural images in the MIT300 dataset [488]. We also observe that both the models previously trained on natural images benefit a lot in terms of performance. In fact, websites are often very rich of natural images; this fact give both FineTuned and KeepTrain a big advantage.

In Table 16.5, we report the results obtained by FineTuned SalGAN, TSGAN and some of the state-of-the-art saliency map prediction approaches, as reported by the authors of [390], when they are applied on the FiWI dataset. As we can see from this table, our fine-tuned variant of SalGAN returns better results than all the other approaches for all the metrics considered.

Model	*NSS*	*AUC-Judd*	*CC*
TSGAN Reference	1.43	0.82	0.66
FineTuned SalGAN	1.61	0.85	0.74
Li et. al [390]	0.91	0.73	0.44
Shen and Zhao [563]	0.88	0.72	0.43
Garcia-Diaz et al [265]	0.82	0.68	0.41

Table 16.5: Values of the adopted evaluation metrics obtained for our fine-tuned variant of SalGAN and some other saliency map prediction approaches proposed in the past literature

We end this section with a qualitative evaluation of the approaches into consideration. In particular, Figure 16.12 reports a representation of how the fine-tuned variant of SalGAN, on one hand, and TSGAN, on the other hand, behave on a layout rich of images, where text is not predominant. From this figure, we can see that our fine-tuned variant of SalGAN is actually performing slightly worse than TSGAN. Indeed, salient areas are less detailed, highlighting wider portions of the image. A less specific prediction introduces many false positives. Instead, TSGAN achieves a better performance than our SalGAN variant, as it minimizes the defects found in this last approach.

From the examination of this figure, we might think that TSGAN performs better than our SalGAN variant. Actually, this is not the case. In fact, the situation changes dramatically in presence of layouts with rich textual information and images. We identify this configuration as a weak point of TSGAN.

(a) Original image **(b) Ground truth**

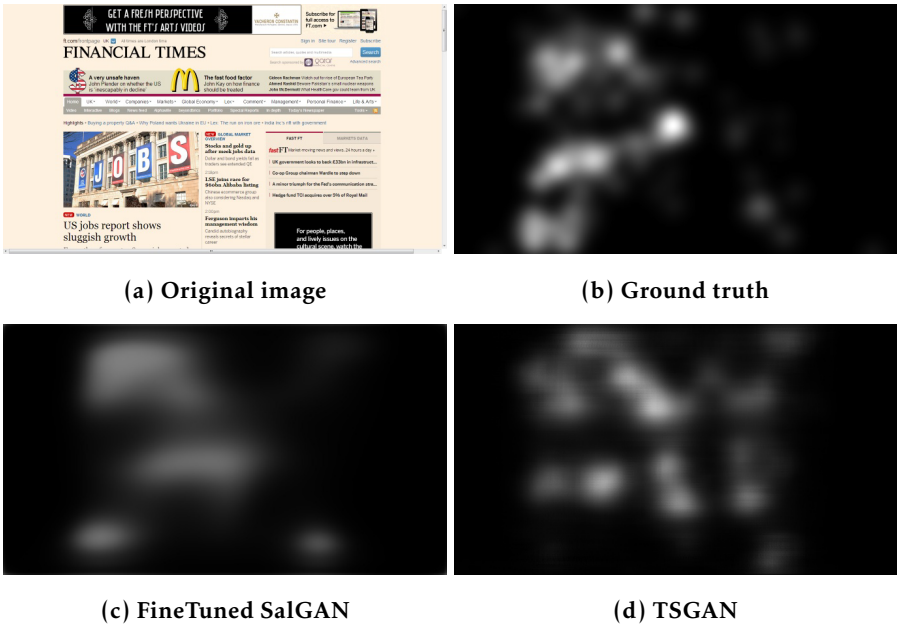
(c) FineTuned SalGAN **(d) TSGAN**

Fig. 16.12: Comparison of the predictions returned by our fine-tuned variant of SalGAN and TSGAN on a layout rich of images

Figure 16.13 shows how our fine-tuned variant of SalGAN is able to return a much better prediction in this case. In fact, TSGAN fails to identify the salient parts and produces an almost completely wrong map. If we compare the metric values computed on this image, we obtain that our SalGAN variant performs much better than TSGAN. The metrics that most highlight this difference are *NSS* and *AUC-Borji*; here, TSGAN scores 1.02 and 0.72, respectively, while SalGAN scores 1.25 and 0.81. The difference in the values of *AUC-Borji* confirms once again how TSGAN struggles to find the salient areas, introducing many false positives.

Actually, it is possible to show that our variant of SalGAN performs generally better than TSGAN on a wider variety of layouts. TSGAN suffers when working with pages rich of information, where every single element could be a highlight. On the other hand, SalGAN is generally not able to provide too detailed information about salient areas, merely identifying large areas that are equally likely. TSGAN generally proves to be better in all those layouts where there is an information scattering, ensuring better detail. As SalGAN is already trained on a large dataset, it is able to generalize better than TSGAN.

All the previous reasonings allow us to conclude that our fine-tuned variant of SalGAN is the preferred model in most situations requiring the saliency map prediction on web pages.



(a) Original image **(b) Ground truth**

(c) FineTuned SalGAN **(d) TSGAN**

Fig. 16.13: Comparison of the predictions returned by our fine-tuned variant of SalGAN and TSGAN on a layout dense of images and texts

Gaze path prediction. As for gaze path prediction, first of all we decided to verify if the original PathGAN, which was explicitly conceived to operate on natural images, showed an acceptable performance on web pages, in order to compare our variants for gaze path prediction (i.e., NormalGAN and WGAN, described in Section 16.1.2) with it.

We recall that NormalGAN uses a content loss weight equal to 1.0, whereas WGAN sets the same parameter to 0.05. The adversarial loss of NormalGAN is set to 0.2, while the one of WGAN is set to 1. Each variant advantages one term of the objective function over the other. This is obtained thanks to the different combination of weight updates between generator and discriminator, which requires a different parameter tuning.

We performed both a quantitative and a qualitative comparison of PathGAN and our two variants. In order to carry out their quantitative evaluation, we leveraged Jarodzka's metrics [320], which define scanpaths as a series of geometric vectors (also called saccade vectors) and compare them across the following dimensions:

- *Vector shape*: it denotes the difference in shape between saccade vectors.
- *Vector direction*: it indicates the difference in direction (i.e., angle) between saccade vectors.
- *Vector length*: it represents the difference in amplitude between saccade vectors.
- *Vector position*: it denotes the distance between fixations.

- *Fixation duration*: it indicates the difference in duration between fixations.

All the measures above range in the real interval $[0,1]$. In fact, the first three measures are normalized by the screen diagonal, vector direction is normalized by π, whereas each fixation duration is normalized against the maximum value of the two durations being compared. The reasoning underlying these measures is the same: the higher the value, the closer saccade vectors.

Recall that each web page in our dataset was observed by 11 or 12 different users (see Section 16.2.2). As a consequence, given a web page, there is no single truth, but every path corresponding to a user who observed it was considered as a ground truth. Based on this choice, the prediction returned by the approach into evaluation was compared with each ground truth and, then, the average performance was computed. Finally, the performances associated with every image were averaged to obtain the evaluation of the approach on the whole dataset. In Table 16.6, we report the results returned by the original PathGAN, NormalGAN and WGAN, when no threshold was set on the fixation duration.

	Shape	*Direction*	*Length*	*Position*	*Duration*
Original PathGAN	0.652	0.421	0.850	0.435	0.295
NormalGAN	0.992	0.693	0.991	0.836	0.290
WGAN	0.993	0.699	0.992	0.840	0.310

Table 16.6: Performance of the original PathGAN, NormalGAN and WGAN when no threshold was set on the duration of fixations

This figure shows that, in the web page domain, the original PathGAN achieves much lower results in all benchmark metrics than NormalGAN and WGAN. As we know, this is due to the fact that the website domain is complex because it can contain several natural images simultaneously, along with text. This makes the direct application of PathGAN (designed for only one natural image at a time) not effective. Looking at NormalGAN and WGAN, it is possible to conclude that the vector shape and the path length are predicted very well by both approaches. In fact, the corresponding values are very high for both variants. The position of fixations is also high, compared to ground truths. On the other hand, direction similarity decreases significantly, even if it remains within an acceptable range. Both variants struggle to determine the duration of each fixation. Duration is by far the metric with the worst performance for both approaches, highlighting a common weakness of them. The values in Table 16.6 also say that WGAN is the best performing model. Indeed, it achieves the best score in all the five metrics. NormalGAN also performs well,

being behind WGAN for just some decimal points in every metric. Table 16.6 also highlights that the two approaches have the same strengths and weaknesses because they perform well and poorly in the same metrics.

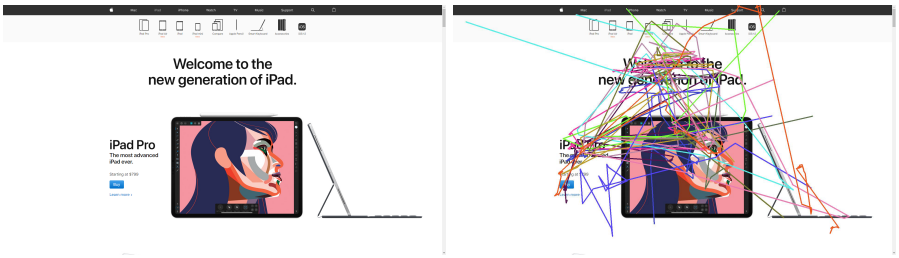
We performed a second quantitative evaluation by setting a threshold on the duration of fixations with the goal of improving results. The idea motivating this attempt was to eliminate the dummy fixations in the prediction generated by the network to return a path 63 fixations long, which is the same output length of the original PathGAN. We set a threshold on the duration of fixations equal to 0.0027; this means that all fixations with shorter duration were not considered as such. Since duration is normalized between 0 and 1, and the predicted path has a total duration of 4 seconds, we can compute the corresponding threshold expressed in seconds. In particular, a threshold of 0.0027 corresponds to about 10 milliseconds, i.e., one order of magnitude smaller than the average fixation duration [128]. This threshold has been conceived in such a way as to avoid losing important fixations in the final prediction. In Table 16.7, we show the results obtained.

	Shape	*Direction*	*Length*	*Position*	*Duration*
Original PathGAN	0.645	0.424	0.852	0.437	0.310
NormalGAN	0.992	0.699	0.991	0.838	0.308
WGAN	0.993	0.698	0.992	0.840	0.326

Table 16.7: Performance of the original PathGAN, NormalGAN and WGAN when a threshold equal to 0.0027 has been set on the duration of fixations

Similarly to Table 16.6, this table shows that the performance of the original PathGAN is much lower than that of NormalGAN and WGAN. This represents a further confirmation that the original PathGAN is not adequate to predict the gaze path of a user while she is surfing web pages. NormalGAN and WGAN represent two ways to increase its effectiveness. Clearly, we are not saying that these variants are the only ways to obtain this result. However, we can certainly say that they result in significant improvements over the original PathGAN whatever evaluation metrics are considered. As far as NormalGAN and WGAN are concerned, we can see that adding a threshold on the duration of fixations introduces only a slight improvement of results. In fact, the performance value regarding duration remains modest, even if a very small improvement is visible. Once again, WGAN confirms as the best approach, even if NormalGAN manages to shorten the distance from WGAN in most metrics.

After the quantitative evaluation, we proceeded with the qualitative one. In Figure 16.14, we report an example of prediction provided by NormalGAN and WGAN models.



(a) Original image **(b) Ground truth**

(c) NormalGAN **(d) WGAN**

Fig. 16.14: Comparison of the predictions returned by NormalGAN and WGAN on one of the web pages of our dataset

From the analysis of this figure, we can easily observe that WGAN performs better in this case. This qualitative conclusion can be drawn considering that:

- Most of the gaze paths of the ground truth pass through the left part of the area between the text and the image. This is because the eye is also caught by the text placed on the image left. The gaze path predicted by WGAN crosses the area between the text and the image in the same way, going to the left. The gaze path predicted by NormalGAN crosses the area between the text and the image from the right side. Therefore, it does not take into account all the gazes that, on the left, are captured by the text close to the figure.
- Most of the gaze paths of the ground truth cross the screen going overall from left to right. The gaze path of WGAN behaves in the same way. Instead, the gaze path of NormalGAN goes immediately from left to right and then back to left making almost a clockwise rotation. This behavior is not found in almost any of the gaze paths of the ground truth.

The metric values computed on this web page confirm again our qualitative conclusions. The direction similarity in WGAN is higher than in NormalGAN, with a

score of 0.74 against 0.70. The same trend between WGAN and NormalGAN can be also observed for all the other metrics. The values of position similarity are very good; it reaches 0.87 in both cases. Also in this experiment, duration does not return satisfactory values. Overall, both approaches show the same strengths and weaknesses.

In every image analyzed we found analogous trends and results. Therefore, we can conclude that WGAN behaves generally better than NormalGAN. This probably happens because Wasserstein training present in WGAN improves the final results in almost every aspect.

After having determined that WGAN is the best gaze path prediction approach for web pages, we must now verify if its absolute performance is anyway acceptable. In fact, it could happen that WGAN, although better than NormalGAN, has very low (and, therefore, unacceptable) performances. Unfortunately, past literature lacks GAN-based approaches to predict gaze paths on websites.

To do this verification, we used an approach that considers humans, their behavior and their evaluation. In particular, we leveraged the well-known One human baseline technique [90]. This technique, given N observers, tells us how well a fixation map of one of them, represented as a saliency map, predicts the fixation of the other $N-1$ observers. This verification task is performed for each of the N observers, and the results thus obtained are averaged. In this way, in turn, each individual is used to predict the behavior of all the others. In order to make an as accurate and complete as possible evaluation, which takes into account not only the average behavior of observers, but also the full range of their possible behaviors, we decided to specify more values for the prediction scores associated with humans. In particular, we considered the maximum, minimum and mean values.

In Table 16.8, we report the values of the evaluation metrics for One human baseline and WGAN. Since, with One human baseline, the evaluation of the gaze paths is transformed into an evaluation of the corresponding saliency map (see [90] for all details), the metrics we use are those related to saliency map prediction.

	NSS	*AUC-Judd*	*AUC-Borji*	*CC*	*KL*
One human baseline	Min: 0.55 Mean: 0.99 Max: 1.61	Min: 0.20 Mean: 0.22 Max: 0.26	Min: 0.49 Mean: 0.51 Max: 0.54	Min: 0.32 Mean: 0.44 Max: 0.59	Min: 4.29 Mean: 6.06 Max: 7.88
WGAN	0.71	0.66	0.61	0.34	7.67

Table 16.8: Comparison between One human baseline and WGAN

Table 16.8 shows that WGAN is able to achieve satisfactory results. For example, consider the *AUC-Judd* and *AUC-Borji* metrics. For them, the mean value reached through One human baseline is 0.22 and 0.51, respectively; instead, WGAN reaches 0.66 and 0.61, respectively. Interestingly, for these two metrics, WGAN achieves an even better performance than the maximum values reached by One human baseline.

On the other hand, WGAN performs below the mean, but still above the minimum for *KL*, *NSS* and *CC*. This means that the prediction of the length and duration of the gaze path made by WGAN is quite different from the values of the ground truth, even if predicted values are still acceptable.

A final contribution on this evaluation process is obtained by considering the qualitative evaluation of WGAN compared to One human baseline. In this case, the gaze path generated by WGAN has a shape close to ground truth. This clearly represents a very encouraging result for the research we have described.

Part III

Closing remarks

This part is dedicated to draw some conclusions on the approaches for networking both people and things. In particular, (i) we draw some conclusions in Chapter 17, and (ii) highlight some possible future developments in Chapter 18.

17

Conclusions

In this chapter, we draw some conclusions on our approaches in networking people and things described in the previous chapters of this thesis.

17.1 Networking people

In the previous chapters of this thesis, we have seen the motivations, the characteristics, the contributions and the results of our approaches to Networking people. Then, we have described each of them in detail. In this section, we provide some conclusion remarks for each of the proposed approaches.

Defining and detecting k-bridges. As for this context, we have introduced the concept of k-bridge and we have found that it enjoys the anti-monotone property.

Starting from this result, we have proposed an algorithm for detecting k-bridges from a social network. With Yelp as the main reference platform, we have discovered several features characterizing k-bridges and we have detected several knowledge patterns about them.

Afterwards, by performing on Reddit and the network of patent inventors some of the experiments we had already carried out on Yelp, we have seen that the properties and the knowledge patterns characterizing k-bridges, that we have found through Yelp, are general and not limited to this social network.

Finally, we have presented two use cases that could benefit from the presence of k-bridges; the former regards the application of k-bridges to find the best targets for a marketing campaign. The latter concerns the role of k-bridges to find new products/services to propose.

Detecting user stereotypes and their assortativity. In this scenario, we have presented an investigation on Reddit, whose aim was analyzing three aspects of this social platform that are interesting for both the theory and the practice.

First, we have examined related literature and we have described the dataset used for our investigation. Then, we have illustrated some preliminary analyses that allowed us to gather some (partially expected) information, useful to correctly carry out the following activities and interpret the corresponding results.

The first knowledge detected in our investigation is subreddit stereotypes. We have explained the way of proceeding that we followed to determine them, we have defined three macro-categories and, for each of them, a certain number of stereotypes. Finally, we have proposed three orthogonal taxonomies and we have classified the detected stereotypes according to them. We have proceeded in the same way performing the second main task of our investigation, namely the definition and the classification of author stereotypes.

Afterwards, we have focused on a more theoretical issue. In fact, analogously to what has been carried out for other social platforms, we have verified if Reddit is assortative, and in which way. We have found that a degree assortativity exists in Reddit and that it involves co-posters. Finally, we have presented several applications that could benefit from subreddit and author stereotypes.

Detecting backbones of information diffusers among different communities of a social platform. As for this context, we have presented an approach for finding information diffusers among different communities of a social platform. First, we have defined the reference scenario, which involves multiple communities in a social platform and a set of users that can act as bridges among communities. Then, we have proposed a model to represent this scenario. Afterwards, we have introduced the concept of disseminator bridge and we have proposed a new form of centrality, called disseminator centrality, specifically designed for the identification of disseminator bridges in the reference scenario. Thanks to this new centrality, we were able to propose a definition of backbone of disseminator bridges and an approach for its construction. We have also considered related literature and we have highlighted the differences between the approaches proposed in the past and ours. Finally, we have presented several experiments to evaluate the performance of our approach.

Investigating NSFW contents and their authors. As for this context, we have proposed two approaches, one based on the semantic and one on the structure of NSFW contents in Reddit.

We have seen that NSFW contents are frequent in this social medium and, despite this, there are very few studies on this subject in the past literature. We have tried to fill this gap and we have proposed an approach that investigates the phenomenon

of NSFW posts in Reddit by performing descriptive, co-posting and assortativity analyses.

In this way, we have derived three findings, which, together with the principles underlying our approach, are certainly the two main contributions of it. In fact, the findings reported provide valuable knowledge to better understand this phenomenon still little investigated. In addition, our way of proceeding defines a methodology that can be used to uncover the dynamics underlying NSFW contents in other social media.

We conducted our analysis on a dataset extracted from `pushshift.io` that contains posts and comments published on 449 NSFW adult subreddits from January 1^{st}, 2020 to March 31^{st}, 2020. The knowledge we were able to extract through our approach is interesting and pertinent, and provides an initial glimpse of light into a world that has been little investigated by researchers in the past. The pattern extraction is done considering not only their frequency, but also, and especially, their utility, according to suitable utility measures. Starting from extracted patterns, our approach constructs three social networks allowing the extraction of information about the users who publish and read NSFW adult posts and comments in Reddit, the texts generally present in them, and the language generally adopted. We conducted our analysis on the same dataset as before.

Investigating negative reviews and negative influencers. In this scenario, we have studied the negative reviews in Yelp proposing a multi-dimensional analysis where the dimensions we have considered are co-reviews, friendships and business categories.

First, we have proposed a preliminary analysis of Yelp data to understand the distribution of categories and reviews in the macro-categories of Yelp. Then, we have focused on three types of users, namely k-bridges, power users and double-life users.

Afterwards, we have seen that power users and double-life users are two subsets of k-bridges. We have also seen that there are two types of double-life users, namely the dl-users (whose double life concerns the amount of reviews made) and the sdl-users (whose double life regards the scores assigned to the businesses of the various macro-categories). As for the scores, we have seen that there is a very good correspondence between the number of stars assigned to the businesses and the polarity obtained by analyzing the review through sentiment analysis tools.

After that, we studied how users can influence each other in making negative reviews on the same businesses and/or on the same macro-categories. As for this aspect, we have seen that each user tends to greatly influence her friends and to be,

in turn, influenced by them. We have also seen that the influence exerted by bridges is greater than that exerted by non-bridges.

Finally, we have built a network that takes into account only negative reviewers and, by conducting a series of analyses and studies on this network, we have seen that the main negative influencers in Yelp are people who are sdl-users and, simultaneously, top users with regard to degree centrality and/or eigenvector centrality and/or page rank.

Investigating user behavior in a blockchain during a cryptocurrency speculative bubble. As for this context, we have illustrated how Social Network Analysis can be used to extract knowledge regarding the behavior of certain categories of users of a cryptocurrency blockchain during a speculative bubble. In particular, we have focused on the speculative bubble that involved Ethereum in the years 2017 and 2018. However, our way of proceeding can be applied to any speculative bubble of any cryptocurrency blockchain.

We first modeled Ethereum as a social network. Then, we defined some categories of users that we considered particularly interesting for our analysis. These categories were the Survivors, the Missings and the Entrants. Proceeding in this way, we have obtained several interesting results. First of all, we have determined the main characteristics distinguishing one category of users from the others. Next, we have found that there exist backbones linking users of specific categories in certain phases (pre-bubble, bubble, post-bubble). Finally, we have defined some guidelines that allowed us, during a certain phase, to determine who will be the main players in the next phase.

Representation, detection and usage of the content semantics of comments. As for this context, we have presented a data structure and a related approach for managing comment semantics in a social platform. Our data structure is network-based and is capable of handling more perspectives about content semantics. It is also easily extensible if additional perspectives are desired in the future. Our approach is based on the mining of text patterns from comments. This activity is carried out based not only on their frequency but also on their utility. The latter is expressed through a utility function that can be chosen according to the reference scenario and the user's needs. Our approach is also able to compute the semantic similarity degree of two sets of comments.

We have also examined several possible applications of our approach, namely: *(i)* the realization of content-based and collaborative filtering recommender systems; *(ii)* the construction of new user communities; *(iii)* and/or the identification of out-

liers. Finally, if applied on Reddit, our approach can also be used for building new subreddits.

Defining user spectra to classify user behaviors in cryptocurrencies. In this scenario, we proposed an automatic social network based approach to classify Ethereum users. First, we have seen that the classification of a user in Ethereum currently occurs only when she requests the validation of her smart contract to a provider in charge of this service, such as Etherscan. As a result, only a small fraction of Ethereum users is presently classified. Our approach is automatic and, therefore, can classify any Ethereum user. The classification of a user is based on her past behavior modeled through the time evolution of eight parameters forming a multivariate time series, which represents her spectrum. In order to compute the similarity between the spectrum of a user and that of a class, we had to fit the Eros distance to our context. We have also tested our approach on a dataset derived from Ethereum and obtained very satisfactory results in terms of both accuracy and computation time.

Extracting information from posts on COVID-19. As for this context, we have presented three approaches to extract information from posts on COVID-19 published on Reddit. The first approach is semi-automatic and incremental. It aims at building, and then updating, a classification of posts on Reddit. This classification allows us to define a hierarchy of classes each characterized by a set of keywords. The second approach is automatic and allows the identification of a set of themes concerning COVID-19. Each theme deals with homogeneous topics and has homogeneous posts associated with it. It can also be seen as the core for the realization of a virtual subreddit. The third approach is automatic and allows the construction of virtual communities of users having the same interests. It can be exploited to define a recommender system suggesting to a user other ones with similar interests or to allow Reddit to propose a new functionality aiming at creating communities of users with common interests. We applied the three approaches on the posts on COVID-19 published on Reddit between January and April 2020 and also reported the information discovered. Finally, we highlighted that the proposed approaches can be applied to analyze the posts about other emergencies published on Reddit.

Extracting time patterns from the lifespans on TikTok challenges. In this scenario, we have proposed an approach to extract time patterns from the lifespans of non-dangerous and dangerous TikTok challenges. We have seen that the patterns we found for the two types of challenges are different. As a consequence, the pres-

ence of a certain pattern can be a strong indicator on the (non) dangerousness of the corresponding challenge.

In light of our results, we can say that our goal of identifying a new model to classify challenges into dangerous and non-dangerous ones has been achieved. In fact, our approach has proved capable of distinguishing the two kinds of challenge. We point out again that it must be considered a first step in our overall research. Indeed, it is currently able to perform the classification near the end of the lifespan of a challenge, or at least after a presumably long period of time. However, a challenging issue for TikTok is to find a new mechanism for the early detection of dangerous challenges. This is very important in order to be able to detect and remove them before they are too successful and reach an exponential growth. Such early detection can be seen as the final goal of our research of which the approach proposed in this thesis represents the first step. In fact, we believe that if we were able to reduce the granularity of the time intervals, so as to make it much finer, we could verify the possibility of extending our approach to identify temporal patterns capable of distinguishing the two kinds of challenge already at the beginning of their lifespan. The early detection of dangerous challenges using time interval analysis could have important applications. For example, it could enrich the set of approaches used by TikTok to detect dangerous challenges for removing them. In addition, it could be used by government regulators to identify dangerous challenges and then ask TikTok to remove them. Last but not least, it could be used to offer a service reporting dangerous challenges or challenges with content "inappropriate" for young people. This service could be extremely valuable for parents and educators (we cannot forget that TikTok is currently the most popular social network among adolescents, and therefore among minors).

Investigating community evolutions in TikTok. As for this context, we have studied the different characteristics and evolutionary dynamics of the user communities participating in non-dangerous and dangerous TikTok challenges. This study led us to the identification of evolutionary patterns allowing us to discriminate the communities of users participating in the two types of challenges. This de facto represents a new approach to identify dangerous challenges in TikTok. Interestingly, our approach, based on the analysis of the behavior of hundreds or thousands of users participating in a challenge, is robust to the classical tricks used to bypass the current TikTok controls. The importance of the fast detection of dangerous challenges is also motivated by another relevant result we obtained, namely the fact that when these challenges begin to succeed, they tend to have an exponential growth of

their users, even much greater than that of the communities associated with non-dangerous challenges.

17.2 Networking things

In the previous chapters of this thesis, we have seen the motivations, the characteristics, the contributions and the results of our approaches to Networking things. Then, we have presented each of them in detail. In this section, we provide some conclusion remarks for each of the proposed approaches.

Networking wearable devices for fall detection in a workplace. As for this context, we proposed a new framework based on Sentient Multimedia Systems and Machine Learning to improve safety at work.

First, we provided a general overview of the proposed framework. Then, we presented a more detailed description of its three layers, namely Personal Devices, Area Devices and Safety Coordination Platform. After that, in order to give a very concrete idea of how our framework can operate in reality, we illustrated its specialization to a typical scenario of safety at work, which is fall detection.

With regard to this scenario, we described how our framework can be adopted to detect falls, activate alarms and coordinate rescue operations. In this description, we paid particular attention to Personal Devices as we introduced a new wearable device based on Machine Learning for fall detection in a workplace, which we designed, built and tested. Then, we took a look at Area Devices.

Finally, we saw how the Safety Coordination Platform can operate to identify a fall, establish its cause and severity and, based on this information, define how to trigger alarms and how to organize and activate a rescue management plan.

Anomaly detection and classification in Multiple IoT scenarios. As for this context, we have presented a first attempt to investigate and classify anomalies in a MIoT.

Our proposal consists of two main components. The first one is a new methodological framework that can make future investigations in this research field easier, more coherent and more uniform. Indeed, our framework extends existing methods to the case of anomaly detection in a MIoT, whilst also allowing the definition of new cases. Another important contribution is the extension to the anomaly detection in MIoT of the so-called forward problem and inverse problem, which have been largely investigated and employed in scientific literature but were never analyzed in this research field. We also introduced a use case on a smart lighting system for a MIoT deployed in a smart city.

Our experiments have provided interesting outcomes about the capability of detecting anomalies and their effects in a MIoT. For instance, they revealed that: *(i)* the effects of an anomaly on a node spread over the surrounding nodes, even if they rapidly decrease against the distance; *(ii)* the anomaly degree defined in our thesis is a parameter that really helps the detection of the anomalous object in a network; *(iii)* the number of nodes affected by an anomaly increases against the number of IoT in a roughly linear way; *(iv)* degree centrality and, even more, closeness centrality are really key parameters in the spread of anomalies in a MIoT.

Increasing protection and autonomy of smart objects in the IoT. In this scenario, we have proposed a two-tier Blockchain framework conceived to increase protection and autonomy of smart objects in the IoT.

First of all, we have seen the motivations underlying our decision to address this issue. Then, we have examined related literature and we have pointed out the main differences and novelties of our approach with respect to the past ones. Afterwards, we have proposed a reference model on which both our framework and the algorithms operating in it are based. Next, we have illustrated our approach to compute the trust of a smart object in another one, the reputation of a smart object in its community and the trust of a community in another one.

After this, we have presented the security model that can be activated by means of our framework. Finally, we have illustrated several experiments devoted to evaluate the performance of our approach and to compare it with two other ones already presented in the past literature.

Extending saliency maps and gaze prediction in an Industry 4.0 scenario. As for this context, we have proposed one fine-tuned variant of SalGAN and two fine-tuned variants of PathGAN conceived for extending saliency map and gaze path prediction from natural images to websites.

First of all, we have seen the motivations underlying our work and, in particular, why this is an important issue to address from both theoretical and application perspectives. Then, we have examined related literature and pointed out the small number of approaches for the evaluation of visual attention on website layouts. Afterwards, we have proposed our variants and described the underlying architecture. Next, we have presented our dataset, which is specifically built for the web domain.

After this, we have shown our experiments, carried out for saliency map and gaze path predictions, and we have compared our variants to the other already existing approaches. Finally, we have presented our prototype that implements all the functionalities discussed and allows the designer to access them easily.

18

Future works

In this chapter, we first present some possible future developments for all the approaches to networking people and things described in the previous two parts of this thesis. Afterwards, we discuss the possibility of defining an approach capable of handling an Internet of Everything scenario and, therefore, of simultaneously managing people and things.

18.1 Networking people

In this section, we illustrate some possible future developments of each approach concerning networking people that we presented in this thesis.

Defining and detecting k-bridges. As for this context, we plan to extend our research efforts in several directions. First of all, we would like to investigate other properties of k-bridges, for instance their capability of being influencers in a social context. Negative influencers are particularly interesting for us because this user stereotype is less studied in the literature even if its impact in real life is enormous. Then, we would like to extend the analysis of k-bridges from Yelp to other platforms similar to it, for instance TripAdvisor, to understand the analogies and the difference with Yelp. Afterwards, we would like to extend, realize and test the approaches described in the two use cases. Finally, we plan to realize a research campaign that performs a profile-based analysis of users for most of the challenges described in order to extract a deep knowledge about k-bridges and their behaviors in the social platforms they belong to.

Detecting user stereotypes and their assortativity. As for this context, we plan to develop our research along several directions. First of all, we would like to carry out a deep investigation on NSFW subreddits. In fact, in spite they are very numerous, few analyses on them have been performed in the past literature. Furthermore, we have seen that the merge, or at least the integration, of related subreddits could be

extremely beneficial. Therefore, we plan to define an approach that finds possible subreddits to merge or to integrate and, then, suggests the tasks necessary to carry out this activity. Last, but not least, we would like to define an approach to find duplicate accounts, i.e. two or more Reddit accounts belonging to the same person. We would like to understand the main motivations leading a user to adopt multiple accounts and verify if she has different behaviors in different accounts.

Detecting backbones of information diffusers among different communities of a social platform. The results presented in this scenario should not be considered as an endpoint but rather as a starting point for the research in this area. In fact, it is possible to think of several future developments.

More specifically, we can apply the approach proposed to other subreddits that cover different topics from COVID-19 or deal with heterogeneous topics. On the other hand, we can also apply our approach to other social platforms. This would allow us to assess whether and how information dissemination changes when passing from a social platform to another.

Another possible future development is the improvement of the network model in such a way as to include other types of interactions among users (e.g., sharing and liking). This would allow us to consider information dissemination from other points of view and possibly highlight the presence of "hidden" users that support information diffusion without exposing themselves openly.

Last but not least, we plan to define an approach to "evaluate" disseminator bridges from multiple perspectives, e.g., for their efficiency, effectiveness, trustworthiness, reputation, etc., based on their past behavior in disseminating the information of their interest.

Investigating NSFW contents and their authors. As for the first approach, concerning this issue, there are several possible developments of our research efforts. First, it is possible to apply the proposed approach to other social media managing NSFW contents. In addition, we could extend our study of NSFW posts by including an in-depth analysis of their content from a semantic point of view. Similarly, we could deepen our knowledge on the authors of NSFW posts applying sentiment analysis techniques to the posts they wrote or commented. Finally, we could consider to define a Machine Learning based approach to automatically identify and label NSFW posts, authors and communities, particularly when NSFW posts are not manually labeled by users. This last application can become extremely important to prevent NSFW contents from being sneakily and deceptively offered to unsuitable users (e.g., children).

As for the second approach, about this issue, the results obtained are not to be considered as a point of arrival but as a starting point. In particular, our approach could be applied to other specific categories of posts and subreddits, for example those dedicated to vegan users or luxury car lovers. Furthermore, the analysis of text patterns could be adopted within an automatic classification system capable of filtering out posts and comments having inopportune patterns/content. Again, the approach proposed could be extended to other social networks that manage NSFW content, also to those performing such a task in a less structured and explicit way than Reddit. Finally, we could think of an approach that integrates text and semantic analysis tools and utility patterns to build a knowledge base capable of automatically classifying new posts and directing them to the most suitable communities.

Investigating negative reviews and negative influencers. As for this context, we plan to extend our research in various directions. First of all, we think of analyzing the phenomenon of negative reviews in other social media, such as TripAdvisor, to understand the similarities and differences with respect to Yelp. Then, we plan to analyze other aspects and other peculiarities of Yelp. Last but not least, we think to define an approach that exploits the anti-monotonic property characterizing the definition of k-bridge to allow the extraction of negative influencers related to a business, a macro-category or a group of target users.

Investigating user behavior in a blockchain during a cryptocurrency speculative bubble. As for this context, the activities described in this thesis are not to be considered as a point of arrival. Instead, they are a starting point for further researches in this field. For example, we might perform further studies on user behavior, taking into account labels identifying the type of addresses in a blockchain. Based on these labels, we would like to define a classification approach that first constructs a profile for all users of each label and, then, employs that profile to classify non-labeled users. In addition, we could think of upgrading from predictive to prescriptive analysis by defining the characteristics that a new user must take over time in a blockchain for quickly becoming one of the main actors in it. Last, but not the least, we could investigate the text data sent along with transactions. Indeed, it would be possible to analyze the shared contents through Natural Language Processing techniques in order to detect additional features allowing a more precise definition of the profiles of the main players in the blockchain.

Representation, detection and usage of the content semantics of comments. As for this context, we plan to extend our research efforts in several directions. First,

we could investigate the possibility of using our approach to build a system that autonomously identifies offensive content of a certain type (cyberbullism, racism, etc.) in a set of comments (e.g., those of a certain user or community) on a social platform. To do so, we should first build a meaningful set of comments with characteristics similar to the ones we want to identify and remove. Then, we should construct a CS-Net $\mathcal{N}_c$ corresponding to these comments. At this point, given a new set $\mathcal{C}_n$ of comments, if the corresponding CS-Net $\mathcal{N}_n$ has a very high semantic similarity degree with $\mathcal{N}_c$, we can conclude that $\mathcal{C}_n$ is offensive and should be removed. Extending the previous idea further, we might consider building a virtual moderator. It could not only remove sets of offensive and inappropriate comments, but also favor the most relevant ones to a certain post or comment. Furthermore, it could associate each user with a reputation degree rewarding her when she publishes relevant comments and penalizing her when she submits irrelevant or offensive ones.

A further interesting issue to investigate regards the evolution of CS-Nets over time. In fact, such an analysis would allow us to identify new trends or topics that characterize a social platform.

Last, but not the least, we could use our approach in a sentiment analysis context. In fact, in the literature, there are several studies on how people with anxiety, and/or psychological and emotional disorders, write their posts or comments on social platforms. We could contribute to these studies by considering a set of comments published by users with such characteristics, constructing the corresponding CS-Nets and analyzing them in detail. We could also compare a CS-Net thus obtained with "template CS-Nets", representative of a certain emotional state, to possibly perform a suitable classification.

Defining user spectra to classify user behaviors in cryptocurrencies. As for this context, we plan to develop the research topics described in this thesis along several directions. First, we would like to extend our approach in order to classify Ethereum entities. We recall that, in the past literature, the term "entity" has been used to denote the set of addresses of a single user. Investigating the exploitation of multiple addresses by a single user is a challenging issue. Indeed, it is first necessary to understand why a user is doing it. Then, it is needed to evaluate if and when it makes sense considering the addresses all together or separately.

Afterwards, we aim at extending the way of proceeding underlying our approach in order to define a similar approach for Bitcoin and compare it with the ones already proposed for this blockchain.

A third extension might be in depth rather than in breadth. In fact, so far we have modeled user behavior by means of a spectrum comprising eight "structural"

features related to transactions made by users. None of these features takes transaction reasons into account. This information, although difficult to extract and process, could be a valuable source for understanding user behavior and being able to classify users more accurately. In the future, we plan to investigate this issue to understand whether the benefits brought by the analysis of transaction reasons outweigh the corresponding costs.

Finally, we believe it is possible to apply graph mining techniques on the social network modeling Ethereum. This could lead to the identification of possible recurring structures and motifs. The discovery of such structures could allow us to define an approach for the detection of ransom demands, fraud, blackmail spread over the network or, even, activities carried out in cooperation by a group of criminals.

Extracting information from posts on COVID-19. As for this context, we plan to extend the research proposed along various directions. First, we would like to generalize the proposed approaches so that they can also operate on other social networks, such as Quora, 4Chan and Digg, just to cite a few. Moreover, we plan to define collaborative filtering recommender systems exploiting the results of the three approaches discussed to suggest to users other users and subreddits with similar interests. Last but not least, we plan to apply sentiment analysis techniques to identify new forms of classification of Reddit users, which consider not only the content of their posts but also the sentiments used to express them.

Extracting time patterns from the lifespans on TikTok challenges. The early detection of dangerous challenges through the analysis of their lifespan is certainly the first future development on this context. In addition to this, we would like to further delve into the investigation of challenges through Social Network Analysis in order to find indicators capable of distinguishing the two types of challenges based on how the corresponding communities evolve over time. Last but not least, we would like to extend our analyses done for challenges to TikTok's trends. These certainly have some similarities with challenges. However, they also have several specificities. Consequently, it is presumable that many of the results found for challenges can be extended to trends by making the suitable changes taking their specificities into account.

Investigating community evolutions in TikTok. As for this context, we plan to further investigate the evolutionary dynamics of the communities associated with challenges using additional features and concepts derived from Social Network Analysis. Second, we plan to further study the distinction between dangerous and non-dangerous challenges by identifying additional criteria allowing the detection of a

dangerous challenge as soon as possible and in the most robust possible way. Last but not least, we could extend our analysis from TikTok challenges to TikTok trends. In fact, these last ones have certainly several analogies with challenges, but, at the same time, present also several differences. Consequently, we can assume that many of the results found for challenges can be extended to trends by making suitable modifications, which consider the peculiarities of trends with respect to challenges.

18.2 Networking things

In this section, we illustrate some possible future developments of each approach concerning networking things that we presented in this thesis.

Networking wearable devices for fall detection in a workplace. As for this context, we are planning to extend our work in several directions. First of all, we think of investigating metrics to evaluate Quality of Service (QoS) and Quality of Experience (QoE) from the worker perspective. Indeed, a continuous feedback from the users on the services they are employing and how they feel while working with our framework can help to identify some adjustments allowing an improvement in QoS and QoE.

Another interesting future development concerns the anonymization of data. In fact, in scenarios like these, workers are surrounded by smart objects. These are certainly useful to increase their safety but, on the other hand, they are able to store a lot of data about workers that, properly combined, could allow the extraction of sensitive information about them. In order to address this problem, some popular database anonymization techniques, such as k-anonymity, l-diversity and t-closeness, could be included in our framework, to ensure that no information can be traced back to the specific worker, unless it is not required.

Finally, it is also interesting including in our framework smart objects able to evaluate the biometric parameters of the worker. Indeed, these could be fundamental to improve the prediction of negative events, such as falls, and to evaluate the level of stress of the worker during her activity. In fact, all the actions leading to an excessive level of stress are to be considered at risk, as they could lead the worker to a drop in concentration that could have disastrous effects on safety.

Anomaly detection and classification in Multiple IoT scenarios. As for this context, we can foresee several developments of our research. First of all, we would like to extend our framework to social networking and/or social internetworking scenarios where humans and objects simultaneously operate. In fact, the investigation

of mixed networks, consisting of humans and smart/social objects, is attracting increasing interest among researchers. Afterwards, we plan to extend our studies on MIoT anomalies for predictive maintenance in such a way as to optimize the maintenance of production lines. Last, but not the least, we think that several results obtained for MIoT can be exploited, by applying a sort of "feedback", to identify new topics and new approaches for the investigation of human behavior in Online Social Networks.

Increasing protection and autonomy of smart objects in the IoT. As for this context, we can think of several developments of our of our research effort. For instance, we plan to combine our approach with other community-based ones conceived to ensure the privacy of smart objects and their owners, with the ultimate goal to define a single solution handling both privacy and security in IoT. Furthermore, we would like to extend our approach adding the possibility to protect the authenticity of the services offered by smart objects. In fact, we have not currently considered how nodes advertise and, then, deliver their services; therefore, we have not taken into account that they might lie about this. Last, but not the least, we plan to improve the computation of object and community reliability using machine learning techniques that can also predict the type of content the requester expects to receive, based on its past history. In this way, the reliability computation would depend not only on technological aspects but also on semantic evaluations.

Extending saliency maps and gaze prediction in an Industry 4.0 scenario. As for this context, we can foresee several developments of our research. For instance, it could be possible to fuse both saliency map and gaze path prediction and create a unique pipeline. In this way, we could exploit the saliency map prediction to generate the corresponding visual scanpath that, we argue, could be more accurate. Finally, it would be also interesting to evaluate the possibility of applying reinforcement learning in this scenario. Here, the challenge would involve the definition of a reward function able to highlight the correct aspects of web interfaces and, therefore, to ensure an appropriate training to the model.

18.3 Networking everything

In the previous sections, we have highlighted the main future developments or our research activities, with regards to the two research lines considered in this thesis, i.e., "Networking people" and "Networking things". However, the most relevant future development we could think of is the merge of this two research lines, in order

to reach the goal of proposing approaches capable of working in the context of the Internet of Everything (IoE).

IoE extends all main concepts of IoT to three more entities, i.e., people, processes and data. So, it aims at providing models, approaches and frameworks capable of merging this four aspects in order to add new capacities and experiences, especially to increase economic and technical potential. IoE is also thought to realize a stricter interaction between physical and virtual world; for this reason, even augmented reality, virtual reality and mixed reality will have an important role in this.

Obviously, IoE is going to require the solution of different problems. Some of them have already been addressed in the IoT context, but need to be redefined to adapt them to the new context; others, instead, are totally new. For example, it will be necessary to tackle the problems of cybersecurity and privacy protection in an innovative way. The pervasiveness of sensors and devices in spaces that also involve the intimacy of people raises privacy problems that are not easy to solve.

Data management will also have to take place in a very different way than in the current scenario. In fact, the combination of all these connected systems will involve an enormous amount of data exchanged on the network. Mobile data traffic is now in the order of exabytes per month. The management of this huge amount of data, many of which streamed, represents a crucial problem in the context of the IoE, to be addressed in a completely new way compared to the past. Paradoxically, in this new context, it would be important to limit the amount of information intended for archiving, bearing in mind that any information must be deleted when it is no longer of general use. This would go against the current trend, as well as with respect to the conception on data management that people and organizations currently have.

On this line, we could continue highlighting many other problems, perhaps even studied in the past, which must be defined or redefined in the context of IoE. We would like to point out that many of these issues are precisely those investigated in this thesis. Therefore, in addition to being in many ways a point of arrival, it is to be understood in many other ways as a starting point for the exploration of new challenging research horizons.

References

1. Six stereotypes you follow on Instagram. https://www.kaindefoecommunications.com/new-england-social-media-marketing/6-stereotypes-you-follow-on-instagram/, 2020.
2. The Stereotypes of Facebook. https://www.ericsson.com/en/blog/2011/9/facebook-stereotypes-which-type-are-you, 2020.
3. Extracting time patterns from the lifespans of TikTok challenges to characterize non-dangerous and dangerous ones, author=G. Bonifazi, S. Cecchini, E. Corradini, L. Giuliani, D. Ursino, and L. Virgili, journal=Social Network Analysis and Mining. 12(1):1–22, 2022. Springer.
4. W. Abdelghani, C.A. Zayani, I. Amous, and F. Sèdes. Trust management in social internet of things: a survey. In *Proc. of the International Conference on e-Business, e-Services and e-Society (IFIP'16)*, pages 430–441, Swansea, United Kingdom, 2016. Springer.
5. M. Abomhara and G.M. Køien. Security and privacy in the Internet of Things: Current status and open issues. In *Proc. of the International Conference on Privacy and Security in Mobile Systems (PRISMS'14)*, pages 1–8, Aalborg, Denmark, 2014. IEEE.
6. M. Abulaish, A. Kamal, and M.J. Zaki. A Survey of Figurative Language and Its Computational Detection in Online Social Networks. *ACM Transaction on the Web*, 14(1):3:1–3:52, 2020. ACM.
7. L. Adamic and E. Adar. Friends and Neighbors on the Web. *Social Networks*, 25(3):211–230, 2003. Elsevier.
8. M. Addlesee, R. Curwen, S. Hodges, J. Newman, P. Steggles, A. Ward, and A. Hopper. Implementing a sentient computing system. *Computer*, 34(8):50–56, 2001. IEEE.
9. M. Adnan, R. Alhajj, and J. G. Rokne. Identifying Social Communities by Frequent Pattern Mining. In *Proc. of the International Conference on Information Visualisation (IV'09)*, pages 413–418, Barcelona, Spain, 2009. IEEE Computer Society.
10. R. Agarwal and R. Srikant. Fast algorithms for mining association rules. In *Proc. of the International VLDB Conference (VLDB'94)*, pages 487–499, Santiago de Chile, Chile, 1994. Morgan Kaufmann.
11. C.C. Aggarwal, M. Bhuiyan, and M. Al Hasan. Frequent pattern mining algorithms: A survey. In J. Han C. Aggarwal, editor, *Frequent Pattern Mining*, pages 19–64. 2014. Springer, Cham.

12. R. Aggarwal, R. Gopal, A. Gupta, and H. Singh. Putting Money Where the Mouths Are: The Relation Between Venture Financing and Electronic Word-of-Mouth. *Information Systems Research*, 23(3):976–992, 2012. INFORMS.
13. M. Ahmed. Collective anomaly detection techniques for network traffic analysis. *Annals of Data Science*, 5(4):497–512, 2018. Springer.
14. M. Ahmed and A.N. Mahmood. Novel approach for network traffic pattern analysis using clustering-based collective anomaly detection. *Annals of Data Science*, 2(1):111–130, 2015. Springer.
15. M. Ahmed, A.N. Mahmood, and J. Hu. A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, 60:19–31, 2016. Elsevier.
16. M. Ahmed and A.S.S.M. Barkat Ullah. Infrequent pattern mining in smart healthcare environment using data summarization. *The Journal of Supercomputing*, 74(10):5041–5059, 2018. Springer.
17. Y.Y. Ahn, S. Han, H. Kwak, S. Moon, and H. Jeong. Analysis of topological characteristics of huge online social networking services. In *Proc. of the International Conference on World Wide Web (WWW'07)*, pages 835–844, Banff, Alberta, Canada, 2007. ACM.
18. E. Akbas and P. Zhao. Attributed graph clustering: An attribute-aware graph embedding approach. In *Proc. of the International Conference on Advances in Social Networks Analysis and Mining (ASONAM'17)*, pages 305–308, Sydney, Autralia, 2017.
19. L. Akoglu, M. McGlohon, and C. Faloutsos. Oddball: Spotting anomalies in weighted graphs. In *Proc. of the Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining, (PAKDD'10) Part II*, pages 410–421, Hyderabad, India, 2010. Lecture Notes in Computer Science, Springer.
20. L. Akoglu, H. Tong, and D. Koutra. Graph based anomaly detection and description: a survey. *Data Mining and Knowledge Discovery*, 29(3):626–688, 2015. Springer.
21. F. Al-Turjman and S. Alturjman. Context-sensitive access in industrial internet of things (iiot) healthcare applications. *IEEE Transactions on Industrial Informatics*, 14(6):2736–2744, 2018. IEEE.
22. F. Al-Turjman and S. Alturjman. 5G/IoT-enabled UAVs for multimedia delivery in industry-oriented applications. *Multimedia Tools and Applications*, 79(13-14):8627–8648, 2020. Springer.
23. F. Al-Turjman, H. Zahmatkesh, and R. Shahroze. An overview of security and privacy in smart cities' iot communications. *Transactions on Emerging Telecommunications Technologies*, page e3677, 2019. Wiley Online Library.
24. A. Al-Zoubi, J. Alqatawna, H. Faris, and M. A Hassonah. Spam profiles detection on social networks using computational intelligence methods: The effect of the lingual context. *Journal of Information Science*, 47(1):58–81, 2019. SAGE.
25. A. Alambo, M. Gaur, U. Lokala, U. Kursuncu, K. Thirunarayan, A. Gyrard, A. Sheth, R.S. Welton, and J. Pathak. Question answering for suicide risk assessment using Reddit. In *Proc. of the International Conference on Semantic Computing (ICSC'19)*, pages 468–473, Newport Beach, CA, USA, 2019. IEEE.

26. A. Alexandrov. Characteristics of single-item measures in Likert scale format. *The Electronic Journal of Business Research Methods*, 8(1):1–12, 2010.
27. S.A. Aljawarneh and R. Vangipuram. Garuda: Gaussian dissimilarity measure for feature representation and anomaly detection in internet of things. *The Journal of Supercomputing*, (11227):1–38, 2018. Springer US.
28. N. Alonso-López, P. Sidorenko-Bautistal, and F. Giacomelli. Beyond challenges and viral dance moves: TikTok as a vehicle for disinformation and fact-checking in Spain, Portugal, Brazil, and the USA. *Anàlisi*, 64:65–84, 2021.
29. M. Aloqaily, S. Otoum, I. Al Ridhawi, and Y. Jararweh. An intrusion detection system for connected vehicles in smart cities. *Ad Hoc Networks*, 90:101842, 2019. Elsevier.
30. K. Altun, B. Barshan, and O. Tunçel. Comparative study on classifying human activities with miniature inertial and magnetic sensors. *Pattern Recognition*, 43(10):3605–3620, 2010.
31. M. Alwan, P.J. Rajendran, S. Kell, D. Mack, S. Dalal, M. Wolfe, and R. Felder. A smart and passive floor-vibration based fall detector for elderly. In *Proc. of the International Conference on Information & Communication Technologies (ICICT'06)*, volume 1, pages 1003–1007, Damascus, Syria, 2006. IEEE.
32. F. Amato, V. Moscato, A. Picariello, and F. Piccialli. SOS: A multimedia recommender System for Online Social networks. *Future Generation Computer Systems*, 93:914–923, 2019. Elsevier.
33. B. Amiri, L. Hossain, J. W. Crawford, and R.T. Wigand. Community detection in complex networks: Multi–objective enhanced firefly algorithm. *Knowledge-Based Systems*, 46:1–11, 2013. Elsevier.
34. K. Anand, J. Kumar, and K. Anand. Anomaly detection in online social network: A survey. In *Proc. of the 2017 International Conference on Inventive Communication and Computational Technologies (ICICCT '17)*, pages 456–459, Coimbatore, India, 2017. IEEE.
35. G. Anania, A. Tognetti, N. Carbonaro, M. Tesconi, F. Cutolo, G. Zupone, and D. De Rossi. Development of a novel algorithm for human fall detection using wearable sensors. *Sensors*, pages 1336–1339, 2008. IEEE.
36. M.S. Anbarasi, V. Iswarya, M. Sindhuja, and S. Yogabindiya. Ontology oriented concept based clustering. *International Journal of Research in Engineering and Technology*, 3(2), 2014.
37. K.E. Anderson. Ask me anything: what is Reddit? 2015. Emerald.
38. S. Angelidis and M. Lapata. Multiple instance learning networks for fine-grained sentiment analysis. *Transactions of the Association for Computational Linguistics*, 6:17–31, 2018.
39. N. Antonakakis, I. Chatziantoniou, and D. Gabauer. Cryptocurrency market contagion: Market uncertainty, market complexity, and dynamic portfolios. *Journal of International Financial Markets, Institutions and Money*, 61:37–51, 2019. Elsevier.
40. M. Arslan, C. Cruz, and D. Ginhac. Semantic enrichment of spatio-temporal trajectories for worker safety on construction sites. *Personal and Ubiquitous Computing*, 23(5-6):749–764, 2019. Springer.

41. Q.M. Ashraf and M. H. Habaebi. Introducing autonomy in internet of things. In *Proc. of the International Conference on Applied Computer and Applied Computational Science (ACACOS'15)*, Kuala Lumpur, Malaysia, 2015.
42. C. Aslay, L.V.S. Lakshmanan, W. Lu, and X. Xiao. Influence maximization in online social networks. In *Proc. of the ACM International Conference on Web Search and Data Mining (WSDM'18)*, pages 775–776, Marina del Rey, CA, USA, 2018. ACM.
43. M. Assens, X. Giro i Nieto, K. McGuinness, and N.E. O'Connor. PathGAN: visual scanpath prediction with generative adversarial networks. In *Proc. of the European Conference on Computer Vision (ECCV'18)*, pages 406–422, Munich, Germany, 2018.
44. S. Asur and B.A. Huberman. Predicting the future with social media. In *Proc. of the International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT'10)*, volume 1, pages 492–499, Toronto, Ontario, Canada, 2010. IEEE.
45. L. Atzori, A. Iera, and G. Morabito. SIoT: Giving a social structure to the Internet of Things. *IEEE Communications Letters*, 15(11):1193–1195, 2011. IEEE.
46. L. Atzori, A. Iera, and G. Morabito. From "smart objects" to "social objects": The next evolutionary step of the Internet of Things. *IEEE Communications Magazine*, 52(1):97–105, 2014. IEEE.
47. L. Atzori, A. Iera, and G. Morabito. Understanding the Internet of Things: definition, potentials, and societal role of a fast evolving paradigm. *Ad Hoc Networks*, 56:122–140, 2017. Elsevier.
48. L. Atzori, A. Iera, G. Morabito, and M. Nitti. The Social Internet of Things (SIoT)– when social networks meet the Internet of Things: Concept, architecture and network characterization. *Computer networks*, 56(16):3594–3608, 2012. Elsevier.
49. C. Baek and M. Elbeck. Bitcoins as an investment or speculative vehicle? A first look. *Applied Economics Letters*, 22(1):30–34, 2015. Taylor & Francis.
50. R. Baghel and R. Dhir. A frequent concepts based document clustering algorithm. *International Journal of Computer Applications*, 4(5):6–12, 2010.
51. Y. Bai, Q. Li, Y. Fan, and S. Liu. Motif-h: a novel functional backbone extraction for directed networks. *Complex & Intelligent Systems*, pages 1–11, 2021. Springer.
52. U.A.B.U.A. Bakar, H. Ghayvat, S.F. Hasanm, and S.C. Mukhopadhyay. *Activity and Anomaly Detection in Smart Home: A Survey*, pages 191–220. Springer International Publishing, Cham, 2016.
53. G. Baldassarre, P. Lo Giudice, L. Musarella, and D. Ursino. The MIoT paradigm: main features and an "ad-hoc" crawler. *Future Generation Computer Systems*, 92:29–42, 2019. Elsevier.
54. S.M.H. Bamakan, I. Nurgaliev, and Q. Qu. Opinion leader detection: A methodological review. *Expert Systems with Applications*, 115:200–222, 2019. Elsevier.
55. J. Bandy and N. Diakopoulos. # TulsaFlop: A Case Study of Algorithmically-Influenced Collective Action on TikTok. *arXiv preprint arXiv:2012.07716*, 2020.
56. S. Bandyopadhyay, M. Sengupta, S. Maiti, and S. Dutta. A survey of middleware for Internet of Things. In *Recent trends in wireless and mobile networks*, pages 288–296. Springer, 2011.

57. F. Bao and R. Chen. Trust management for the Internet of Things and its application to service composition. In *Proc. of the International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'12)*, pages 1–6, San Francisco, CA, USA, 2012. IEEE.
58. F. Bao, R. Chen, and J. Guo. Scalable, adaptive and survivable trust management for community of interest based Internet of Things systems. In *Proc. of the IEEE International Symposium on Autonomous Decentralized Systems (ISADS'13)*, pages 1–7, Mexico City, Mexico, 2013. IEEE.
59. F. Bao and I. Cheny. Dynamic trust management for internet of things applications. In *Proc. of the International Workshop on Self-aware Internet of Things (ICAC'12)*, pages 1–6, San Jose, CA, USA, 2012. ACM.
60. M.S. Bartlett. The effect of non-normality on the t distribution. *Mathematical Proceedings of the Cambridge Philosophical Society*, 31(2):223–231, 1935. Cambridge University Press.
61. M. Bartoletti, S. Carta, T. Cimoli, and R. Saia. Dissecting Ponzi schemes on Ethereum: identification, analysis, and impact. *Future Generation Computer Systems*, 102:259–277, 2020. Elsevier.
62. M. Bartoletti, B. Pes, and S. Serusi. Data mining for detecting Bitcoin Ponzi schemes. In *Proc. of the International Crypto Valley Conference on Blockchain Technology (CVCBT '18)*, pages 75–84, Zug, Switzerland, 2018. IEEE.
63. Z. Batooli and M. Sayyah. Measuring social media attention of scientific research on novel coronavirus disease 2019 (COVID-19): An investigation on article-level metrics data of dimensions. *Preprint from Research Square*, 2020.
64. K. Bauman and A. Tuzhilin. Discovering contextual information from user reviews for recommendation purposes. In *Proc. of the International Workshop on New Trends in Content-Based Recommender Systems (CBRecSys @ RecSys 2014)*, pages 2–9, Foster City, CA, USA, 2014.
65. J. Baumgartner, S. Zannettou, B. Keegan, M. Squire, and J. Blackburn. The pushshift Reddit dataset. In *Proc. of the International AAAI Conference on Web and Social Media (ICWSM'20)*, volume 14, pages 830–839, Atlanta, GA, USA, 2020. AAAI Press.
66. M.G. Baydogan and G. Runger. Learning a symbolic representation for multivariate time series classification. *Data Mining and Knowledge Discovery*, 29(2):400–422, 2015. Springer.
67. M. Behniafar, A.R. Nowroozi, and H.R. Shahriari. A survey of anomaly detection approaches in internet of things. *The ISC International Journal of Information Security*, 10(2):79–92, 2018. Iranian Society of Cryptology.
68. J.L. Bender, M.-C. Jimenez-Marroquin, and A.R. Jadad. Seeking support on facebook: A content analysis of breast cancer groups. *Journal of Medical Internet Research*, 13(1):e16, 2011. JMIR Publications.
69. F. Benevenuto, T. Rodrigues, M. Cha, and V. Almeida. Characterizing user behavior in online social networks. In *Proc. of the ACM SIGCOMM Conference on Internet measurement*, pages 49–62, Chicago, IL, USA, 2009. ACM.

70. J. Berger, A.T. Sorensen, and S.J. Rasmussen. Positive effects of negative publicity: When negative reviews increase sales. *Marketing science*, 29(5):815–827, 2010. INFORMS.
71. M. Berlingerio, M. Coscia, F. Giannotti, A. Monreale, and D. Pedreschi. Foundations of Multidimensional Network Analysis. In *Proc. of the International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2011)*, pages 485–489, Kaohsiung, Taiwan, 2011. IEEE.
72. M. Berlingerio, D. Koutra, T. Eliassi-Rad, and C. Faloutsos. Netsimile: A scalable approach to size-independent network similarity. *arXiv preprint arXiv:1209.2684*, 2012.
73. M. Berlingerio, F. Pinelli, and F. Calabrese. Abacus: frequent pattern mining-based community discovery in multidimensional networks. *Data Mining and Knowledge Discovery*, 27(3):294–320, 2013.
74. J. Bernabé-Moreno, A. Tejeda-Lorente, C. Porcel, H. Fujita, and E. Herrera-Viedma. Quantifying the emotional impact of events on locations with social media. *Knowledge-Based Systems*, 146:44–57, 2018. Elsevier.
75. D.J. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In *Proc. of the International Conference on Knowledge Discovery in Databases (KDD'94)*, volume 10, pages 359–370, Seattle, WA, USA, 1994. AAAI Press.
76. D. Bertram. Likert scales. *Retrieved November*, 2:2013, 2007.
77. P.K. Bhanodia, A. Khamparia, B. Pandey, and S. Prajapat. Online social network analysis. In *Hidden Link Prediction in Stochastic Social Networks*, pages 50–63. IGI Global, 2019.
78. S. Bhatt, S. Padhee, A. Sheth, K. Chen, V. Shalin, D. Doran, and B. Minnery. Knowledge graph enhanced community detection and characterization. In *Proc. of the International Conference on Web Search and Data Mining (WSDM'19)*, pages 51–59, Melbourne, Australia, 2019.
79. A.Q. Bhatti, M. Umer, S. H. Adil, M. Ebrahim, D. Nawaz, and F. Ahmed. Explicit Content Detection System: An Approach towards a Safe and Ethical Environment. *Applied Computational Intelligence and Soft Computing*, page 1463546, 2018. Hindawi.
80. A.K. Bhowmick, S. Suman, and B. Mitra. Effect of information propagation on business popularity: A case study on yelp. In *Proc. of the International Conference on Mobile Data Management (MDM'17)*, pages 11–20, Daejeon, South Korea, 2017. IEEE.
81. R. Bian, Y. S. Koh, G. Dobbie, and A. Divoli. Identifying top-k nodes in social networks: A survey. *ACM Computing Surveys (CSUR)*, 52(1):1–33, 2019. ACM New York, NY, USA.
82. K. Bibi, S. Naz, and A. Rehman. Biometric signature authentication using machine learning techniques: Current trends, challenges and opportunities. *Multimedia Tools and Applications*, 79(1):289–340, 2020. Springer.
83. P.V. Bindu, P. Santhi Thilagam, and D. Ahuja. Discovering suspicious behavior in multilayer social networks. *Computers in Human Behavior*, 73:568–582, 2017. Elsevier.
84. K. Biswas and V. Muthukkumarasamy. Securing smart cities using blockchain technology. In *Proc. of the IEEE International Conference on High pPerformance Computing and Communications; IEEE International Conference on Smart City; IEEE International Conference on Data Science and Systems (HPCC/SmartCity/DSS 2016)*, pages 1392–1393, Sydney, Australia, 2016. IEEE.

85. B.M. Blau. Price dynamics and speculative trading in bitcoin. *Research in International Business and Finance*, 41:493–499, 2017. Elsevier.
86. P.J. Boczkowski, M. Matassi, and E. Mitchelstein. How young users deal with multiple platforms: The role of meaning-making in social media repertoires. *Journal of Computer-Mediated Communication*, 23(5):245–259, 2018. Oxford University Press.
87. P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017. MIT Press.
88. G. Bonifazi, E. Corradini, D. Ursino, and L. Virgili. A Social Network Analysis based approach to investigate user behavior during a cryptocurrency speculative bubble. *Journal of Information Science*, 2021. SAGE.
89. L. Bontemps, V.L. Cao, J. McDermott, and N. Le-Khac. Collective anomaly detection based on long short-term memory recurrent neural networks. In *Proc. of the International Conference on Future Data and Security Engineering (FDSE'16)*, pages 141–152, Can Tho City, Vietnam, 2016.
90. A. Borji. Saliency prediction in the deep learning era: Successes and limitations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2019. IEEE.
91. A. Borji and L. Itti. State-of-the-art in visual attention modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):185–207, 2012. IEEE.
92. A. Borji, H.R. Tavakoli, D.N. Sihite, and L. Itti. Analysis of scores, datasets, and models in visual saliency prediction. In *Proc. of the International Conference on Computer Vision (ICCV'13)*, pages 921–928, Sidney, Australia, 2013. IEEE.
93. C. Bothorel, J.D. Cruz, M. Magnani, and B. Micenkova. Clustering attributed graphs: models, measures and methods. *Network Science*, 3(3):408–444, 2015. Cambridge University Press.
94. M. Bouabdellah, N. Kaabouch, F. El Bouanani, and H. Ben-Azza. Network layer attacks and countermeasures in cognitive radio networks: A survey. *Journal of Information Security and Applications*, 38:40–49, 2018. Elsevier.
95. Z. Bouraoui, J. Camacho-Collados, and S. Schockaert. Inducing relational knowledge from BERT. In *Proc. of the International Conference on Artificial Intelligence (AAAI 2020)*, volume 34(05), pages 7456–7463, New York, NY, USA, 2020. Association for the Advancement of Artificial Intelligence.
96. E. Bouri, C.K.M. Lau, B. Lucey, and D. Roubaud. Trading volume and the predictability of return and volatility in the cryptocurrency market. *Finance Research Letters*, 29:340–346, 2019. Elsevier.
97. A.K. Bourke and G.M. Lyons. A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor. *Medical engineering & physics*, 30(1):84–90, 2008. Elsevier.
98. A. Boutet, H. Kim, and E. Yoneki. What's in Twitter, I know what parties are popular and who you are supporting now! *Social Network Analysis and Mining*, 3(4):1379–1391, 2013. Springer.
99. L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. Springer.

100. P. Bruce, A. Bruce, and P. Gedeck. *Practical Statistics for Data Scientist, Second Edition*. O'Reilly, Sebastopol, CA, USA, 2020.

101. C.M. Bruno. A Content Analysis of How Healthcare Workers Use TikTok. *Elon Journal of Undergraduate Research in Communications*, 11(2):5–16, 2020.

102. F. Buccafurri, L. Coppolino, S. D'Antonio, A. Garofalo, G. Lax, A. Nocera, and L. Romano. Trust-Based Intrusion Tolerant Routing in Wireless Sensor Networks. In *Proc. of the International Conference on Computer Safety, Reliability and Security (SAFECOMP 2014)*, pages 214–229, Firenze, Italy, 2014. Springer.

103. F. Buccafurri, V.D. Foti, G. Lax, A. Nocera, and D. Ursino. Bridge Analysis in a Social Internetworking Scenario. *Information Sciences*, 224:1–18, 2013. Elsevier.

104. F. Buccafurri, G. Lax, A.Nocera, and D. Ursino. A system for extracting structural information from Social Network accounts. *Software Practice & Experience*, 45(9):1251–1275, 2015. John Wiley & Sons.

105. F. Buccafurri, G. Lax, S. Nicolazzo, and A. Nocera. Accountability-Preserving Anonymous Delivery of Cloud Services. In *Proc. of the International Conference on Trust, Privacy and Security in Digital Business (TRUSTBUS 2015)*, pages 124–135. Springer, 2015.

106. F. Buccafurri, G. Lax, S. Nicolazzo, and A. Nocera. Comparing Twitter and Facebook user behavior: Privacy and other aspects. *Computers in Human Behavior*, 52:87–95, 2015. Elsevier.

107. F. Buccafurri, G. Lax, S. Nicolazzo, and A. Nocera. Interest Assortativity in Twitter. In *Proc. of the 12th International Conference on Web Information Systems and Technologies (WEBIST 2016)*, pages 239–246, Rome, Italy, 2016. "SCITEPRESS – Science and Technology Publications, Lda".

108. F. Buccafurri, G. Lax, A. Nocera, and D. Ursino. Supporting Information Spread in a Social Internetworking Scenario. *Post-Proceedings of the International Workshop on New Frontiers in Mining Complex Knowledge Patterns at ECML/PKDD 2012 (NFMCP 2012)*, 200–214. Lecture Notes in Artificial Intelligence, Springer.

109. F. Buccafurri, G. Lax, A. Nocera, and D. Ursino. Internetworking assortativity in Facebook. In *Proc. of the International Conference on Social Computing and its Applications (SCA 2013)*, pages 335–341, Karlsruhe, Germany, 2013. IEEE Computer Society.

110. F. Buccafurri, G. Lax, A. Nocera, and D. Ursino. Moving from social networks to social internetworking scenarios: The crawling perspective. *Information Sciences*, 256:126–137, 2014. Elsevier.

111. F. Buccafurri, G. Lax, A. Nocera, and D. Ursino. Discovering Missing Me Edges across Social Networks. *Information Sciences*, 319:18–37, 2015. Elsevier.

112. C. Buntain and J. Golbeck. Identifying Social Roles in Reddit Using Network Structure. In *Proc. of the International Conference on World Wide Web (WWW'14)*, page 615–620, Seoul, Korea, 2014. ACM.

113. S. Butterworth. On the theory of filter amplifiers. *Wireless Engineer*, 7(6):536–541, 1930.

114. J.W. Byers, M. Mitzenmacher, and G. Zervas. The groupon effect on yelp ratings: a root cause analysis. In *Proc. of the ACM Conference on Electronic Commerce (EC'12)*, pages 248–265, Valencia, Spain, 2012. ACM.

115. F. Cabitza, D. Fogli, and A. Piccinno. Fostering participation and co-evolution in sentient multimedia systems. *Journal of Visual Languages & Computing*, 25(6):684–694, 2014. Elsevier.

116. D. Cai, Z. Shao, X. He, X. Yan, and J. Han. Community mining from multi-relational networks. In *Proc. of the European Conference on Principles of Data Mining and Knowledge Discovery (PKDD'05)*, pages 445–452, Porto, Portugal, 2005. Springer.

117. D. Camacho, M. V. Luzón, and E. Cambria. New trends and applications in social media analytics. *Future Generation Computer Systems*, 114:318–321, 2021. Elsevier.

118. D. Camacho, A. Panizo-LLedot, G. Bello-Orgaz, A. Gonzalez-Pardo, and E. Cambria. The four dimensions of social network analysis: An overview of research methods, applications, and software tools. *Information Fusion*, 63:88–120, 2020. Elsevier.

119. R. Camino, C.F. Torres, M. Baden, and R. State. A data science approach for honeypot detection in Ethereum. *arXiv preprint arXiv:1910.01449*, 2019. arXiv.

120. U. Can and B. Alatas. A new direction in social network analysis: Online social network analysis problems and applications. *Physica A: Statistical Mechanics and its Applications*, 535:122372, 2019. Elsevier.

121. J.H. Canós, G. Alonso, and J. Jaén. A multimedia approach to the efficient implementation and use of emergency plans. *IEEE Multimedia*, 11(3):106–110, 2004. IEEE.

122. V. Carchiolo, A. Longheu, M. Malgeri, G. Mangioni, and M. Previti. Mutual Influence of Users Credibility and News Spreading in Online Social Networks. *Future Internet*, 13(5):107, 2021. Multidisciplinary Digital Publishing Institute.

123. M. Carpenter and M. Garner. NSFW: An Empirical Study of Scandalous Trademarks. *Cardozo Arts & Ent. LJ*, 33:321, 2015. HeinOnline.

124. L. Caruccio and S. Cirillo. Incremental Discovery of Imprecise Functional Dependencies. *Journal of Data and Information Quality (JDIQ)*, 2019. ACM.

125. L. Caruccio, S. Cirillo, V. Deufemia, and G. Polese. Incremental Discovery of Functional Dependencies with a Bit-vector Algorithm. In *Atti del Ventisettesimo Convegno Nazionale su Sistemi Evoluti per Basi di Dati (SEBD'19)*, Castiglione della Pescaia (GR), Italy, 2019.

126. E. Casilari, J. Santoyo-Ramón, and J. Cano-García. Analysis of public datasets for wearable fall detection systems. *Sensors*, 17(7):1513, 2017.

127. N. Cassavia, E. Masciari, C. Pulice, and D. Saccà. Discovering User Behavioral Features to Enhance Information Search on Big Data. *ACM Transactions on Interactive Intelligent Systems*, 7(2), 2017. ACM.

128. M.S. Castelhano and K. Rayner. Eye movements during reading, visual search, and scene perception: An overview. *Cognitive and cultural influences on eye movements*, 2175:3–33, 2008.

129. F. Cauteruccio, L. Cinelli, G. Fortino, C. Savaglio, and G. Terracina. Using sentiment analysis and automated reasoning to boost smart lighting systems. In *Proc. of the 12th International Conference in Internet and Distributed Computing Systems (IDCS 2019)*, volume 11874 of *LNCS*, pages 69–78, Naples, Italy, 2019. Springer.

130. F. Cauteruccio, E. Corradini, G. Terracina, D. Ursino, and L. Virgili. Investigating Reddit to detect subreddit and author stereotypes and to evaluate author assortativity. *Journal of Information Science*, 2021. SAGE.

131. F. Cauteruccio, E. Corradini, G. Terracina, D. Ursino, and L. Virgili. Extraction and analysis of text patterns from NSFW adult content in Reddit. *Data & Knowledge Engineering*, 138:101979, 2022. Elsevier.

132. F. Cauteruccio, G. Fortino, A. Guerrieri, A. Liotta, D.C. Mocanu, C. Perra, G. Terracina, and M.T. Vega. Short-long term anomaly detection in wireless sensor networks based on machine learning and multi-parameterized edit distance. *Information Fusion*, 52:13–30, 2019. Elsevier.

133. K. Chaccour, R. Darazi, A.H. El Hassans, and E. Andres. Smart carpet using differential piezoresistive pressure sensors for elderly fall detection. In *Proc. of the International Conference on Wireless and Mobile Computing, Networking and Communications (WIMOB'15)*, pages 225–229, Abu-Dhabi, United Arab Emirates, 2015. IEEE.

134. P. Chaim and M.P. Laurini. Is Bitcoin a bubble? *Physica A: Statistical Mechanics and its Applications*, 517:222–232, 2019. Elsevier.

135. H.L. Chan. CGU-BES Dataset for Fall and Activity of Daily Life. 8 2018.

136. W. Chan and A. Olmsted. Ethereum transaction graph analysis. In *Proc. of the International Conference for Internet Technology and Secured Transactions (ICITST'17)*, pages 498–500, Cambridge, MA, USA, 2017. IEEE.

137. V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computer Surveys*, 41(3):15:1–15:58, 2009. ACM.

138. I. Chandra, N. Sivakumar, C.B. Gokulnath, and P. Parthasarathy. IoT based fall detection and ambient assisted system for the elderly. *Cluster Computing*, 22(1):2517–2525, 2019. Springer.

139. Y.C. Chang, C.H. Ku, and C.H. Chen. Social media analytics: Extracting and visualizing Hilton hotel ratings and reviews from TripAdvisor. *International Journal of Information Management*, 48:263–279, 2019. Elsevier.

140. A. Chauhan, O. P. Malviya, M. Verma, and T. S. Mor. Blockchain and scalability. In *Proc. of the IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C 2018)*, pages 122–128, Lisbon, Portugal, 2018. IEEE.

141. S. Chawla, K. Garimella, A. Gionis, and D. Tsang. Backbone discovery in traffic networks. *International Journal of Data Science and Analytics*, 1(3):215–227, 2016. Springer.

142. E.T. Cheah and J. Fry. Speculative bubbles in Bitcoin markets? An empirical investigation into the fundamental value of Bitcoin. *Economics Letters*, 130:32–36, 2015. Elsevier.

143. P.C. Cheeseman and J.C. Stutz. Bayesian classification (AutoClass): theory and results. *Advances in knowledge discovery and data mining*, 180:153–180, 1996. Philadelphia, PA, USA.

144. C.Y.H. Chen and C.M. Hafner. Sentiment-induced bubbles in the cryptocurrency market. *Journal of Risk and Financial Management*, 12(2):53, 2019. Multidisciplinary Digital Publishing Institute.

145. D. Chen, H. Gao, L. Lu, and T. Zhou. Identifying influential nodes in large-scale directed networks: the role of clustering. *PloS one*, 8(10):e77455, 2013. Public Library of Science.

146. E. Chen, K. Lerman, and E. Ferrara. Tracking Social Media Discourse About the COVID-19 Pandemic: Development of a Public Coronavirus Twitter Data Set. *JMIR Public Health and Surveillance*, 6(2):e19273, 2020. JMIR Publications.

147. F. Chen and D.B. Neill. Non-parametric scan statistics for event detection and forecasting in heterogeneous social media graphs. In *Proc. of the International Conference on Knowledge Discovery and Data Mining (KDD'14)*, pages 1166–1175, New York, NY, USA, 2014. ACM.

148. G. Chen, B.D. Ward, C. Xie, W. Li, Z. Wu, J. Jones, M. Franczak, P. Antuono, and S. Li. Classification of Alzheimer disease, mild cognitive impairment, and normal cognitive status with large-scale network analysis based on resting-state functional MR imaging. *Radiology*, 259(1):213–221, 2011. Radiological Society of North America, Inc.

149. H. Chen, P. Han, B. Yu, and C. Gao. A new kind of session keys based on message scheme for sensor networks. In *Proc. of the International Asia-Pacific Microwave Conference (APMC'05)*, volume 1, pages 4–pp, Suzhou, China, 2005. IEEE.

150. H. Chen, H. Wu, X. Zhou, and C. Gao. Agent-based trust model in wireless sensor networks. In *Proc. of the ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD 2007)*, volume 3, pages 119–124, Qingdao, China, 2007. IEEE.

151. Q. Chen, C. Min, W. Zhang, X. Ma, R. Evans, et al. Factors driving citizen engagement with government TikTok accounts during the COVID-19 pandemic: Model development and analysis. *Journal of Medical Internet Research*, 23(2):e21463, 2021. JMIR Publications.

152. W. Chen, Z. Zheng, J. Cui, E. Ngai, P. Zheng, and Y. Zhou. Detecting Ponzi schemes on Ethereum: Towards healthier blockchain technology. In *Proc. of the International World Wide Web Conference (WWW'18)*, pages 1409–1418, Lyon, France, 2018. ACM.

153. W. Chen, Z. Zheng, E.C.H. Ngai, P. Zheng, and Y. Zhou. Exploiting blockchain data to detect smart Ponzi schemes on Ethereum. *IEEE Access*, 7:37575–37586, 2019. IEEE.

154. X. Chen, Z. Qin, Y. Zhang, and T. Xu. Learning to rank features for recommendation over multiple categories. In *Proc. of the International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'16)*, pages 305–314, New York, NY, USA, 2016. ACM.

155. X. Chen, K.D.B. Valdovinos, and J. Zeng. # PositiveEnergy Douyin: constructing "playful patriotism" in a Chinese short-video application. *Chinese Journal of Communication*, 14(1):97–117, 2021. Taylor & Francis.

156. X. Chen, Y. Yuan, and M. Ali Orgun. Using bayesian networks with hidden variables for identifying trustworthy users in social networks. *Journal of Information Science*, 46(5):600–615, 2019. SAGE.

157. Z. Chen, W. Hendrix, and N.F. Samatova. Community-based anomaly detection in evolutionary networks. *Journal of Intelligent Information Systems*, 39(1):59–85, 2012. Springer.

158. Z. Chen and W. Sun. Scanpath Prediction for Visual Attention using IOR-ROI LSTM. In *Proc. of the International Joint Conferences on Artificial Intelligence (IJCAI'18)*, pages 642–648, Stockholm, Sweden, 2018.

159. Z. Chen and Q. Zhang. A Survey Study on Successful Marketing Factors for Douyin (TikTok). In *Proc. of the International Conference on Human-Computer Interaction (HCI'21)*, pages 22–42, Washington DC, USA, 2021. Springer.

160. H. Cheng, X. Xing, X. Liu, and Q. Lv. ISC: An Iterative Social Based Classifier for Adult Account Detection on Twitter. *IEEE Transactions on Knowledge and Data Engineering*, 27(4):1045–1056, 2015. IEEE.

161. A. Cheung, E. Roca, and J. Su. Crypto-currency bubbles: an application of the Phillips–Shi–Yu (2013) methodology on Mt. Gox bitcoin prices. *Applied Economics*, 47(23):2348–2358, 2015. Taylor & Francis.

162. C.M.K. Cheung and M.K.O Lee. What Drives Consumers to Spread Electronic Word of Mouth in Online Consumer-Opinion Platforms. *Decision Support Systems*, 53(1):218–225, 2012. Elsevier.

163. C.M.K. Cheung and D.R. Thadani. The impact of Electronic Word-of-Mouth Communication: A Literature Analysis and Integrative Model. *Decision Support Systems*, 54(1):461—470, 2012. Elsevier.

164. C. Chew and G. Eysenbach. Pandemics in the age of Twitter: content analysis of tweets during the 2009 H1N1 outbreak. *PloS one*, 5(11), 2010. Public Library of Science.

165. J. Cho, S. Rager, J. O'Donovan, S. Adali, and B. D. Horne. Uncertainty-based false information propagation in social networks. *ACM Transactions on Social Computing*, 2(2):1–34, 2019. ACM New York, NY, USA.

166. N. Choudhary, C. Gautam, and V. Arya. Digital marketing challenge and opportunity with reference to TikTok - A new rising social media platform. *International Journal of Multidisciplinary Educational Research*, 9(10), 2020.

167. N. Chouhan, H.K. Saini, and S.C. Jain. Internet of Things: Illuminating and Study of Protection and Justifying Potential Countermeasures. In *Soft Computing and Signal Processing*, pages 21–27. Springer, 2019.

168. S. Chowdhury, M. Khanzadeh, R. Akula, F. Zhang, S. Zhang, H. Medal, M. Marufuzzaman, and L. Bian. Botnet detection using graph-based feature clustering. *Journal of Big Data*, 4(1):1–23, 2017. Springer.

169. M.S. Christian, J.C. Bradley, J.C. Wallace, and M.J. Burke. Workplace safety: a meta-analysis of the roles of person and situation factors. *Journal of Applied Psychology*, 94(5):1103, 2009. American Psychological Association.

170. K. Christidis and M. Devetsikiotis. Blockchains and smart contracts for the Internet of Things. *IEEE Access*, 4:2292–2303, 2016. IEEE.

171. P. Chunaev. Community detection in node-attributed social networks: a survey. *Computer Science Review*, 37:100286, 2020. Elsevier.

172. M. Cinelli, W. Quattrociocchi, A. Galeazzi, C.M. Valensise, E. Brugnoli, A.L. Schmidt, P. Zola, F. Zollo, and A. Scala. The COVID-19 social media infodemic. *arXiv preprint arXiv:2003.05004*, 2020.

173. S. Cirillo, D. Desiato, and B. Breve. Chravat-chronology awareness visual analytic tool. In *Proc. of the International Conference Information Visualisation (IV'19)*, pages 255–260, Paris, France, 2019. IEEE.

174. J. Cohen. Trusses: Cohesive subgraphs for social network analysis. *National security agency technical report*, 16(3.1), 2008.

175. M. Coletto, L.M. Aiello, C. Lucchese, and F. Silvestri. Adult content consumption in online social networks. *Social Network Analysis and Mining*, 7(1):28:1–28:21, 2017. Springer.

176. T. Connie, M. Al-Shabi, and M. Goh. Smart content recognition from images using a mixture of convolutional neural networks. In *IT Convergence and Security 2017*, pages 11–18. 2018. Springer.

177. S. Corbet, B. Lucey, A. Urquhart, and L. Yarovaya. Cryptocurrencies as a financial asset: A systematic analysis. *International Review of Financial Analysis*, 62:182–199, 2019. Elsevier.

178. S. Corbet, B. Lucey, and L. Yarovaya. Datestamping the Bitcoin and Ethereum bubbles. *Finance Research Letters*, 26:81–88, 2018. Elsevier.

179. E. Corradini, A. Nocera, D. Ursino, and L. Virgili. Defining and detecting k-bridges in a social network: the Yelp case, and more. *Knowledge-Based Systems*, 187:104820, 2020. Elsevier.

180. E. Corradini, A. Nocera, D. Ursino, and L. Virgili. Investigating the phenomenon of NSFW posts in Reddit. *Information Sciences*, 566:140–164, 2021. Elsevier.

181. D. Correa, L. A. Silva, M. Mondal, F. Benevenuto, and K. P. Gummadi. The many shades of anonymity: Characterizing anonymous social media content. In *Proc. of the International AAAI Conference on Web and Social Media (ICWSM 2015)*, pages 71–80, Oxford, UK, 2015. AAAI.

182. A. Coutrot, J.H. Hsiao, and A.B. Chan. Scanpath modeling and classification with hidden Markov Models. *Behavior research methods*, 50(1):362–379, 2018. Springer.

183. R. Cramery, R. Gennaroz, and B. Schoenmakersx. A secure and optimally efficient multi-authority election scheme. *European transactions on Telecommunications*, 8(5):481–490, 1997.

184. R. Cucchiara, A. Prati, and R. Vezzani. A multi-camera vision system for fall detection and alarm generation. *Expert Systems*, 24(5):334–345, 2007. Wiley Online Library.

185. Y. Cui. An Evaluation of Yelp Dataset. *arXiv preprint arXiv:1512.06915*, 2015.

186. T.O. Cunha, I. Weber, H. Haddadi, and G.L. Pappa. The effect of social feedback in a Reddit weight loss community. In *Proc. of the International Conference on Digital Health Conference (ICDHT'16)*, pages 99–103, Bordeaux, France, 2016. Springer.

187. W. Dai, G.Z Jin, J. Lee, and M. Luca. Optimal aggregation of consumer ratings: an application to yelp.com. *NBER Working Paper Series*, page 18567, 2012.

188. A.K. Dalai and S.K. Jena. Wdtf: A technique for wireless device type fingerprinting. *Wireless Personal Communications*, 97(2):1911–1928, 2017.

189. K. Darwish, P. Stefanov, M.J. Aupetit, and P. Nakov. Unsupervised User Stance Detection on Twitter. In *Proc. of the International Conference on Web and Social Media (ICWSM 2020)*, pages 141–152, Atlanta, GA, USA, 2020. AAAI Press.

190. T. Dasu, Y.Kanza, and D. Srivastava. Unchain your blockchain. In *Proc. of the International Symposium on Foundations and Applications of Blockchain (FAB'18)*, volume 1, pages 16–23, Los Angeles, CA, USA, 2018.

191. S. Datta and E. Adar. Extracting Inter-Community Conflicts in Reddit. In *Proc. of the International Conference on Web and Social Media (ICWSM 2019)*, pages 146–157, Munich, Germany, 2019. AAAI.

192. I. Davidson, A. Gourru, J. Velcin, and Y. Wu. Behavioral differences: insights, explanations and comparisons of French and US Twitter usage during elections. *Social Network Analysis and Mining*, 10(1):1–27, 2020. Springer.

193. D. Davis, R. Lichtenwalter, and N.V. Chawla. Multi-relational link prediction in heterogeneous information networks. In *Proc. of the International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2011)*, pages 281–288, Kaohsiung, Taiwan, 2011. IEEE.

194. M. Davis. "This is For You": An Anthropological Approach to Relationships to TikTok and its Algorithm. Technical report, University of Chicago, 2021.

195. P.V.A. de Freitas, G.N.P. Santos, A.J.G. Busson, A.L.V. Guedes, and S. Colcher. A baseline for NSFW video detection in e-learning environments. In *Proc. of the Brazillian Symposium on Multimedia and the Web (WebMedia 2019)*, pages 357–360, Rio de Janeiro, Brazil, 2019. ACM.

196. P. De Meo, G. Quattrone, G. Terracina, and D. Ursino. Integration of XML Schemas at various "severity" levels. *Information Systems*, 31(6):397–434, 2006.

197. R. DeJordy and D. Halgin. Introduction to ego network analysis. *Boston MA: Boston College and the Winston Center for Leadership & Ethics*, 2008.

198. J. Deng, R. Han, and S. Mishra. A performance evaluation of intrusion-tolerant routing in wireless sensor networks. In *Information Processing in Sensor Networks*, pages 349–364, 2003. Springer.

199. C. Diamantini, P. Lo Giudice, D. Potena, E. Storti, and D. Ursino. An approach to extracting topic-guided views from the sources of a data lake. *Information Systems Frontiers*, 23(1):243–262, 2021. Springer Nature.

200. C. Diamantini, A. Nocera, D. Potena, E. Storti, and D. Ursino. Find the Right Peers: Building and Querying Multi-IoT Networks Based on Contexts. In *Proc. of the International Conference on Flexible Query Answering Systems (FQAS'19)*, pages 302–313, Amantea, Italy, 2019. Springer.

201. G. Diraco, A. Leone, and P. Siciliano. An active vision system for fall detection and posture recognition in elderly healthcare. In *Proc. of the Design, Automation & Test in Europe Conference & Exhibition (DATE'10)*, pages 1536–1541, Dresden, Germany, 2010. IEEE.

202. Y. Djenouri, A. Belhadi, P. Fournier-Viger, and J.C. Lin. Fast and effective cluster-based information retrieval using frequent closed itemsets. *Information Sciences*, 453:154–167, 2018. Elsevier.

203. C. Donato, P. Lo Giudice, R. Marretta, D. Ursino, and L. Virgili. A well-tailored centrality measure for evaluating patents and their citations. *Journal of Documentation*, 75(4):750–772, 2019. Emerald.

204. S. Dorogovtsev, A. Goltsev, and J. Mendes. K-core organization of complex networks. *Physical Review Letters*, 96(4):040601, 2006. APS.

205. A. Dorri, S. Kanhere, R. Jurdak, and P. Gauravaram. LSB: A Lightweight Scalable Blockchain for IoT security and anonymity. *Journal of Parallel and Distributed Computing*, 134:180–197, 2019.

206. A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram. Blockchain for IoT security and privacy: The case study of a smart home. In *Proc. of the International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops'17)*, pages 618–623, Kona, HI, USA, 2017. IEEE.

207. A. Dorri, S.S. Kanhere, and R. Jurdak. Towards an optimized blockchain for IoT. In *Proc. of the International Conference on Internet-of-Things Design and Implementation (IoTDI'17)*, pages 173–178, Pittsburgh, PA, USA, 2017. IEEE.

208. A. Dorri, S.S. Kanhere, R. Jurdak, and P. Gauravaram. Lsb: A lightweight scalable blockchain for iot security and privacy. *Journal of Parallel and Distributed Computing*, 134:180–197, 2017. Elsevier.

209. J. R. Douceur. The Sybil attack. In *Proc. of the International Workshop on Peer-To-Peer Systems (IPTPS'02)*, pages 251–260, Cambridge, MA , USA, 2002. Springer.

210. R.Y. Dougnon, P. Fournier-Viger, and R. Nkambou. Inferring user profiles in online social networks using a partial social graph. In *Proc. of Canadian Conference on Artificial Intelligence*, pages 84–99, Halifax, Nova Scotia, Canada, 2015. Springer.

211. G. Douzas and F. Bacao. Effective data generation for imbalanced learning using conditional generative adversarial networks. *Expert Systems with applications*, 91:464–471, 2018. Elsevier.

212. M. Du, R. Christensen, W. Zhang, and F. Li. Pcard: Personalized restaurants recommendation from card payment transaction records. In *Proc. of the World Wide Web Conference (WWW 2019)*, pages 2687–2693, San Francisco, CA, USA, 2019. ACM.

213. L. Duan and Y. Xiong. Big data analytics and business analytics. *Journal of Management Analytics*, 2(1):1–21, 2015. Taylor & Francis.

214. R. Duan, X. Chen, and T. Xing. A QoS architecture for IOT. In *Proc. of the International Conference on Internet of Things and International Conference on Cyber, Physical and Social Computing (CPSCom'11)*, pages 717–720, Dalian, China, 2011. IEEE.

215. A.D. Dubey. Twitter Sentiment Analysis during COVID-19 Outbreak. *Available at SSRN 3572023*, 2020.

216. R.E. Dubrofsky and M.M. Wood. Posting racism and sexism: Authenticity, agency and self-reflexivity in social media. *Communication and Critical/Cultural Studies*, 11(3):282–287, 2014. Routledge.

217. A.D. Dwivedi, G. Srivastava, S. Dhar, and R. Singh. A decentralized privacy-preserving healthcare blockchain for IoT. *Sensors*, 19(2):326, 2019. Multidisciplinary Digital Publishing Institute.

218. M. C. De Donato E. Corradini D. Ursino E. Anceschi, G. Bonifazi and L. Virgili. SaveMeNow. AI: a Machine Learning based wearable device for fall detection in a workplace. In *Enabling AI Applications in Data Science*, pages 493–514. 2021. Springer.

219. A. Nocera D. Ursino E. Corradini, S. Nicolazzo and L. Virgili. A two-tier Blockchain framework to increase protection and autonomy of smart objects in the IoT. *Computer Communications*, 181:338–356, 2022. Elsevier.

220. A. Scopelliti D. Ursino E. Corradini, G. Porcino and L. Virgili. Fine-tuning SalGAN and PathGAN for extending saliency map and gaze path prediction from natural images to websites. *Expert Systems with Applications*, 191:116282, 2022. Elsevier.

221. D. Ursino E. Corradini, A. Nocera and L. Virgili. Defining and detecting k-bridges in a social network: the yelp case, and more. *Knowledge-Based Systems*, 195:105721, 2020. Elsevier.

222. D. Ursino E. Corradini, A. Nocera and L. Virgili. Investigating negative reviews and detecting negative influencers in Yelp through a multi-dimensional social network based model. *International Journal of Information Management*, 60:102377, 2021. Elsevier.

223. D. Ursino E. Corradini, A. Nocera and L. Virgili. Investigating the phenomenon of NSFW posts in Reddit. *Information Sciences*, 566:140–164, 2021. Elsevier.

224. F. Ebrahimi, A. Asemi, A. Nezarat, and A. Ko. Developing a mathematical model of the co-author recommender system using graph mining techniques and big data applications. *Journal of Big Data*, 8(1):1–15, 2021. Springer.

225. A. ElBahrawy, L. Alessandretti, A. Kandler, R. Pastor-Satorras, and A. Baronchelli. Evolutionary dynamics of the cryptocurrency market. *Royal Society Open Science*, 4(11):170623, 2017. The Royal Society Publishing.

226. EU ENISA. Baseline Security Recommendations for IoT in the context of Critical Information Infrastructures, 2017.

227. Y. Eom. Premium and speculative trading in bitcoin. *Finance Research Letters*, page 101505, 2020. Elsevier.

228. S. Eraslan, Y. Yesilada, and S. Harper. Scanpath trend analysis on web pages: Clustering eye tracking scanpaths. *ACM Transactions on the Web*, 10(4):1–35, 2016.

229. Z. Ertem, A. Veremyev, and S. Butenko. Detecting large cohesive subgroups with high clustering coefficients in social networks. *Social Networks*, 46:1–10, 2016. Elsevier.

230. T. Van Erven and P. Harremos. Rényi divergence and Kullback-Leibler divergence. *IEEE Transactions on Information Theory*, 60(7):3797–3820, 2014. IEEE.

231. A. Esfandyari, M. Zignani, S. Gaito, and G.P. Rossi. User identification across online social networks in practice: Pitfalls and solutions. *Journal of Information Science*, 44(3):377–391, 2018. SAGE Publications.

232. E. Corradini G. Terracina D. Ursino L. Virgili C. Savaglio A. Liotta F. Cauteruccio, L. Cinelli and G. Fortino. A framework for anomaly detection and classification in Multiple IoT scenarios. *Future Generation Computer Systems*, 114:322–335, 2021. Elsevier.

233. G. Terracina D. Ursino F. Cauteruccio, E. Corradini and L. Virgili. Investigating Reddit to detect subreddit and author stereotypes and to evaluate author assortativity. *Journal of Information Science*, page 0165551520979869, 2020. SAGE Publications Sage UK: London, England.

234. G. Terracina D. Ursino F. Cauteruccio, E. Corradini and L. Virgili. Extraction and analysis of text patterns from NSFW adult content in Reddit. *Data & Knowledge Engineering*, 138:101979, 2022. Elsevier.

235. S. Fakhraei, J.R. Foulds, M.V.S. Shashanka, and L. Getoor. Collective Spammer Detection in Evolving Multi-Relational Social Networks. In *Proc. of the International Conference on Knowledge Discovery and Data Mining (KDD'15)*, pages 1769–1778, Sydney, Australia, 2015. ACM.

236. C. Fan, Y. Jiang, Y. Yang, C. Zhang, and A. Mostafavi. Crowd or Hubs: information diffusion patterns in online social networks in disasters. *International Journal of Disaster Risk Reduction*, 46:101498, 2020. Elsevier.

237. M. Fernández and G. Valiente. A graph distance metric combining maximum common subgraph and minimum common supergraph. *Pattern Recognition Letters*, 22(6-7):753–758, 2001. Elsevier.

238. M. Ferrara, D. Fosso, D. Lanatà, R. Mavilia, and D. Ursino. A Social Network Analysis based approach to extracting knowledge patterns about innovation geography from patent databases. *International Journal of Data Mining, Modelling and Management*, 10(1):23–71, 2018. Inderscience.

239. B. Ferwerda and M. Schedl. Personality-Based User Modeling for Music Recommender Systems. In *Proc. of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD 2016)*, pages 254–257, Riva del Garda, Italy, 2016. Springer International Publishing.

240. A. Fiallos, C. Fiallos, and S. Figueroa. Tiktok and Education: Discovering Knowledge through Learning Videos. In *Proc. of the International Conference on eDemocracy & eGovernment (ICEDEG'21)*, pages 172–176, Quito, Ecuador, 2021. IEEE.

241. M. Fire and C. Guestrin. The rise and fall of network stars: Analyzing 2.5 million graphs to reveal how high-degree vertices emerge over time. *Information Processing & Management*, 57(2):102041, 2020. Elsevier.

242. A. Floris and L. Atzori. Quality of Experience in the Multimedia Internet of Things: Definition and practical use-cases. In *Proc. of the IEEE International Conference on Communication Workshop (ICCW'15)*, pages 1747–1752, London, United Kingdom, 2015. IEEE.

243. J. Fogel and S. Zachariah. Intentions to use the yelp review website and purchase behavior after reading reviews. *Journal of Theoretical and Applied Electronic Commerce Research*, 12(1):53–67, 2017.

244. C. Forman, A. Ghose, and B. Wiesenfeld. Examining the Relationship Between Reviews and Sales: The Role of Reviewer Identity Disclosure in Electronic Markets. *Information Systems Research*, 19(3):291—-313, 2008. INFORMS.

245. P. Fouque, G. Poupard, and J. Stern. Sharing decryption in the context of voting or lotteries. In *Proc. of the International Conference on Financial Cryptography (FC'00)*, pages 90–104, Anguilla, Anguilla, 2000. Springer.

246. P. Fournier-Viger, J. C.-W. Lin, R. U. Kiran, Y.S. Koh, and R. Thomas. A survey of sequential pattern mining. *Data Science and Pattern Recognition*, 1(1):54–77, 2017.

247. P. Fournier-Viger, J.C.W. Lin, R. Nkambou, B. Vo, and V.S. Tseng. *High-Utility Pattern Mining*. 2019. Springer.

248. P. Fournier-Viger, J.C.W. Lin, B. Vo, T.T. Chi, J. Zhang, and H.B. Le. A survey of itemset mining. *WIREs Data Mining and Knowledge Discovery*, 7(4):e1207, 2017. Wiley.

249. D.W. Franks, J. Noble, P. Kaufmann, and S. Stagl. Extremism propagation in social networks with hubs. *Adaptive Behavior*, 16(4):264–274, 2008.

250. N. Fritz and A. Gonzales. Privacy at the Margins| not the normal trans story: negotiating trans narratives while crowdfunding at the margins. *International Journal of Communication*, 12:20, 2018.

251. A. Fronzetti Colladon, B. Guardabascio, and R. Innarella. Using social network and semantic analysis to analyze online travel forums and forecast tourism demand. *Decision Support Systems*, 123:113075, 2019. Elsevier.

252. J. Fry and E.T. Cheah. Negative bubbles and shocks in cryptocurrency markets. *International Review of Financial Analysis*, 47:343–352, 2016. Elsevier.

253. D. Ursino G. Bonifazi, E. Corradini and L. Virgili. A social network analysis–based approach to investigate user behaviour during a cryptocurrency speculative bubble. *Journal of Information Science*, page 01655515211047428, 2021. SAGE Publications Sage UK: London, England.

254. D. Ursino G. Bonifazi, E. Corradini and L. Virgili. Defining user spectra to classify Ethereum users based on their behavior. *Journal of Big Data*, 9(1):1–39, 2022.

255. D. Ursino G. Bonifazi, E. Corradini and L. Virgili. New Approaches to Extract Information from Posts on COVID-19 Published on Reddit. *International Journal of Information Technology & Decision Making*, pages 1–47, 2022. World Scientific.

256. E. Corradini L. Giuliani D. Ursino G. Bonifazi, S. Cecchini and L. Virgili. Investigating community evolutions in TikTok dangerous and non-dangerous challenges. *Journal of Information Science*, page 01655515221116519, 2022. SAGE Publications Sage UK: London, England.

257. E. Corradini M. Marchetti A. Pierini G. Terracina D. Ursino G. Bonifazi, F. Cauteruccio and L. Virgili. An approach to detect backbones of information diffusers among different communities of a social platform. *Data & Knowledge Engineering*, 140:102048, 2022. Elsevier.

258. E. Corradini M. Marchetti G. Terracina D. Ursino G. Bonifazi, F. Cauteruccio and L. Virgili. Representation, detection and usage of the content semantics of comments in a social platform. *Journal of Information Science*, page 01655515221087663, 2022. SAGE Publications Sage UK: London, England.

259. L. Gadár and J. Abonyi. Frequent pattern mining in multidimensional organizational networks. *Scientific Reports*, 9(1):1–12, 2019. Nature Publishing Group.

260. W. Gan, C. Lin, P. Fournier-Viger, H. Chao, V. Tseng, and P. Yu. A Survey of Utility-Oriented Pattern Mining. *IEEE Transactions on Knowledge and Data Engineering*, 33(4):1306–1327, 2021. IEEE.

261. S. Ganeriwal, L.K. Balzano, and M.B. Srivastava. Reputation-based framework for high integrity sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 4(3):1–37, 2008.

262. S. Ganeriwal, R. Kumar, C.C. Han, S. Lee, and M.B. Srivastava. Location & Identity based Secure Event Report Generation for Sensor Networks. *NESL Technical Report*, 2004.

263. F. Gao, K. Musial, and B. Gabrys. A community bridge boosting social network link prediction model. In *Proc. of the International Conference on Advances in Social Networks Analysis and Mining (ASONAM'17)*, pages 683–689, Sydney, Australia, 2017.

264. X. Gao, B. Xiao, D. Tao, and X. Li. A survey of graph edit distance. *Pattern Analysis and applications*, 13(1):113–129, 2010. Springer.

265. A. Garcia-Diaz, V. Leboran, X.R. Fdez-Vidal, and X.M. Pardo. On the relationship between optical variability, visual saliency, and eye fixations: A computational approach. *Journal of vision*, 12(6):17–17, 2012. The Association for Research in Vision and Ophthalmology.

266. P. Garcia-Teodoro, J.E. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez. Anomaly-based network intrusion detection: Techniques, systems and challenges. *Computers & Security*, 28(1-2):18–28, 2009. Elsevier.

267. S. Garg, K. Kaur, S. Batra, G. Kaddoum, N. Kumar, and A. Boukerche. A multi-stage anomaly detection scheme for augmenting the security in iot-enabled applications. *Future Generation Computer Systems*, 104:105–118, 2020. Elsevier.

268. R. Genuer, J.M. Poggi, and C. Tuleau-Malot. Variable selection using random forests. *Pattern Recognition Letters*, 31(14):2225–2236, 2010. Elsevier.

269. H.U. Gerber and G. Pafum. Utility functions: from risk theory to finance. *North American Actuarial Journal*, 2(3):74–91, 1998. Taylor & Francis.

270. J.C. Gerlach, G. Demos, and D. Sornette. Dissection of Bitcoin's multiscale bubble history from January 2012 to February 2018. *Royal Society Open Science*, 6(7):180643, 2019. The Royal Society.

271. R.M. Gibson, A. Amira, N. Ramzan, P. Casaseca de-la Higuera, and Z. Pervez. Multiple comparator classifier framework for accelerometer-based fall detection and diagnostic. *Applied Soft Computing*, 39:94–103, 2016.

272. P. Lo Giudice, L. Musarella, G. Sofo, and D. Ursino. An approach to extracting complex knowledge patterns among concepts belonging to structured, semi-structured and unstructured sources in a data lake. *Information Sciences*, 478:606–626, 2019. Elsevier.

273. P. Gogoi, D.K. Bhattacharyya, B. Borah, and J. K. Kalita. A survey of outlier detection methods in network anomaly identification. *The Computer Journal*, 54(4):570–588, 2011. Elsevier.

274. J.H. Goldberg and J.I. Helfman. Visual scanpath representation. In *Proc. of the Symposium on Eye-Tracking Research & Applications (ETMA'10)*, pages 203–210, Austin, Texas, USA, 2010. ACM.

275. N. Gozzi, M. Tizzani, M. Starnini, F. Ciulla, D. Paolotti, A. Panisson, and N. Perra. Collective response to media coverage of the covid-19 pandemic on reddit and wikipedia: Mixed-methods analysis. *Journal of Medical Internet Research*, 22(10):e21597, 2020. JMIR.

276. M.S. Granovetter. The strength of weak ties. *American Journal of Sociology*, 78(6):1360–1380, 1973. JSTOR.

277. A. Grewal and J. Lin. The evolution of content analysis for personalized recommendations at Twitter. In *Proc. of the International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR '18)*, pages 1355–1356, Ann Arbor, MI, USA, 2018. ACM.

278. Peerless Research Group. Sensors in Distribution: On the Cusp of New Performance Efficiencies. `https://www.logisticsmgmt.com/wp_content/honeywell_wp_sensors_022316b.pdf`, 2015.

279. J. Gu, W. Feng, J. Zeng, H. Mamitsuka, and S. Zhu. Efficient semisupervised MEDLINE document clustering with MeSH-semantic and global-content constraints. *IEEE Transactions on Cybernetics*, 43(4):1265–1276, 2012. IEEE.

280. Y. Gu, J. Chang, Y. Zhang, and Y. Wang. An element sensitive saliency model with position prior learning for web pages. In *Proc. of the International Conference on Innovation in Artificial Intelligence (ICIAI'19)*, pages 157–161, London, England, 2019.

281. L. Guan, B. Hao, Q. Cheng, P.S.F. Yip, and T. Zhu. Identifying Chinese microblog users with high suicide probability using internet-based profile and linguistic features: classification model. *JMIR mental health*, 2(2):e4227, 2015. JMIR Publications Inc., Toronto, Canada.

282. S. Guan, H. Ma, and Y. Wu. Attribute-Driven Backbone Discovery. In *Proc. of the International Conference on Knowledge Discovery & Data Mining (KDD'19)*, pages 187–195, Anchorage, AK, USA, 2019.

283. I.D. Guedalia, J. Guedalia, R.P. Chandhok, and S. Glickfield. Methods to discover, configure, and leverage relationships in Internet of Things (IoT) networks, feb 20 2018. US Patent 9,900,171.

284. J. Guerreiro and P. Rita. How to predict explicit recommendations in online reviews using text mining and sentiment analysis. *Journal of Hospitality and Tourism Management*, 43:269–272, 2020. Elsevier.

285. L. Gui, Y. Zhou, R. Xu, Y. He, and Q. Lu. Learning representations from heterogeneous network for sentiment classification of product reviews. *Knowledge-Based Systems*, 124:34–45, 2017. Elsevier.

286. A. Guimaraes, O. Balalau, E. Terolli, and G. Weikum. Analyzing the Traits and Anomalies of Political Discussions on Reddit. In *Proc. of the International Conference on Web and Social Media (ICWSM 2019)*, pages 205–213, Munich, Germany, 2019. AAAI.

287. A. Gulati and M. Eirinaki. With a Little Help from My Friends (and Their Friends): Influence Neighborhoods for Social Recommendations. In *Proc. of the World Wide Web Conference (WWW'19)*, pages 2778–2784, San Francisco, CA, USA, 2019. ACM.

288. I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville. Improved Training of Wasserstein GANs, 2017.

289. R. K. Gunupudi, M. Nimmala, N. Gugulothu, and S. R. Gali. Clapp: A self constructing feature clustering approach for anomaly detection. *Future Generation Computer Systems*, 74:417–429, 2017. Elsevier.

290. J. Guo, I.R. Chen, and J.J.P. Tsai. A survey of trust computation models for service management in Internet of Things systems. *Computer Communications*, 97:1–14, 2017. Elsevier.

291. A. Gupta, J. Gautam, and A. Kumar. A survey on methodologies used for semantic document clustering. In *Proc. of the International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS'17)*, pages 671–675, Chennai, India, 2017. IEEE.

292. M. Haenlein, E. Anadol, T. Farnsworth, H. Hugo, J. Hunichen, and D. Welte. Navigating the New Era of Influencer Marketing: How to be Successful on Instagram, TikTok, & Co. *California Management Review*, 63(1):5–25, 2020.

293. W. Hamilton, J. Zhang, C. Danescu-Niculescu-Mizil, D. Jurafsky, and J. Leskovec. Loyalty in Online Communities. In *Proc. of the International Conference on Web and Social Media (ICWSM 2017)*, pages 540–543, Montreal, Canada, 2017. AAAI.

294. J. Han, M. Kamber, and J. Pei. *Data Mining: Concepts and Techniques - Third Edition*. 2011. Morgan Kaufmann notes.

295. J. Han, J. Pei, Y. Yin, and R. Mao. Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach. *Data Mining and Knowledge Discovery*, 8(1):53–87, 2004. Springer.

296. J. Harel, C. Koch, and P. Perona. Graph-based visual saliency. In *Proc. of the International Conference on Neural Information Processing Systems (NIPS'07)*, pages 545–552, Cambridge, MA, USA, 2007. MIT Press.

297. M. Hauswirth, A. Datta, and K. Aberer. Handling identity in peer-to-peer systems. In *14th International Workshop on Database and Expert Systems Applications, 2003. Proceedings.*, pages 942–946. IEEE, 2003.

298. D.M. Hawkins. *Identification of outliers / D.M. Hawkins*. Chapman and Hall London ; New York, New York, 1980.

299. R. He, X. Li, G. Chen, G. Chen, and Y. Liu. Generative adversarial network-based semi-supervised learning for real-time risk warning of process industries. *Expert Systems with Applications*, 150:113244, 2020. Elsevier.

300. F. Hendrikx, K. Bubendorfer R., and Chard. Reputation systems: A survey and taxonomy. *Journal of Parallel and Distributed Computing*, 75:184–197, 2015. Elsevier.

301. J. Herrman. How TikTok is rewriting the world. *The New York Times*, 10, 2019.

302. J. Hessel, C. Tan, and L. Lee. Science, AskScience, and BadScience: On the Coexistence of Highly Related Communities. In *Proc. of the International Conference on Web and Social Media (ICWSM 2016)*, pages 171–180, Cologne, Germany, 2016. AAAI.

303. C. Emma Hilton. Unveiling self-harm behaviour: what can social media site Twitter tell us about self-harm? A qualitative exploration. *Journal of clinical nursing*, 26(11-12):1690–1704, 2017. Wiley Online Library.

304. A.O. Hirschman. The paternity of an index. *The American Economic Review*, 54(5):761–762, 1964.

305. Y.C. Ho, J. Wu, and Y. Tan. Disconfirmation Effect on Online Rating Behavior: A Structural Model. *Information Systems Research*, 28(3):626––642, 2008. INFORMS.

306. S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. MIT Press.

307. K. Hoffman, D. Zage, and C. Nita-Rotaru. A survey of attack and defense techniques for reputation systems. *ACM Computing Surveys*, 42(1):1–31, 2009. ACM New York, NY, USA.

308. B. D. Horne, S. Adali, and S. Sikdar. Identifying the social signals that drive online discussions: A case study of reddit communities. In *2017 26th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–9, 2017. IEEE.

309. M.S. Hossain and R.A. Angryk. GDClust: A graph-based document clustering technique. In *Proc. of the International Conference on Data Mining Workshops (ICDMW'07)*, pages 417–422, Washington, DC, USA, 2007. IEEE.

310. M. Hossin and M.N. Sulaiman. A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining & Knowledge Management Process*, 5(2):1, 2015. Academy & Industry Research Collaboration Center (AIRCC).

311. L. Hu, A. Sun, and Y. Liu. Your neighbors affect your ratings: on geographical neighborhood influence to rating prediction. In *Proc. of the International ACM SIGIR Conference on Research & development in information retrieval (SIGIR'14)*, pages 345–354, Gold Coast, Queensland, Australia, 2014. ACM.

312. B. Huang, Z. Liu, J. Chen, A. Liu, Q. Liu, and Q. He. Behavior pattern clustering in blockchain networks. *Multimedia Tools and Applications*, 76(19):20099–20110, 2017.

313. L. Huang, R.X. Li, K.M. Wen, and X.W. Gu. A Self Training Semi-Supervised Truncated Kernel Projection Machine for Link Prediction. *Advanced Materials Research*, 580:369–373, 2012. Trans Tech Publications Inc.

314. S. Huh, S. Cho, and S. Kim. Managing IoT devices using blockchain platform. In *Proc. of the International Conference on Advanced Communication Technology (ICACT'17)*, pages 464–467, PyeongChang, Korea, 2017. IEEE.

315. K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Söderström. Detecting Spacecraft Anomalies Using LSTMs and Nonparametric Dynamic Thresholding. In *Proc. of the International Conference on Knowledge Discovery and Data Mining (KDD'18)*, pages 387–395, London, UK, 2018. ACM.

316. F. Hussain, M.B. Umair, M. Ehatisham ul Haq, I.M. Pires, T. Valente, N.M. Garcia, and N. Pombo. An Efficient Machine Learning-based Elderly Fall Detection Algorithm. *arXiv preprint 1911.11976*, 2019.

317. C.J. Hutto and E. Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proc. of the International AAAI Conference on Weblogs and Social Media (ICWSM'14)*, pages 216–225, Ann Arbor, MI, USA, 2014.

318. L. Jain and R. Katarya. Discover opinion leader in online social network using firefly algorithm. *Expert Systems with Applications*, 122:1–15, 2019. Elsevier.

319. A. Jana and S. Bhattacharya. Design and validation of an attention model of web page users. *Advances in Human-Computer Interaction*, 2015:1–14, 2015. Hindawi.

320. H. Jarodzka, K. Holmqvist, and M. Nyström. A vector-based, multidimensional scanpath similarity measure. In *Proc. of the Symposium on Eye Tracking Research & Applications (ETRA'10)*, pages 211–218, Austin, TX, USA, 2010. ACM.

321. X. Ji, S.A. Chun, P. Cappellari, and J. Geller. Linking and using social media data for enhancing public health analytics. *Journal of Information Science*, 43(2):221–245, 2017. SAGE Publications Sage UK: London, England.

322. H. Jian and H. Chen. A portable fall detection and alerting system based on k-NN algorithm and remote medicine. *China Communications*, 12(4):23–31, 2015. IEEE.

323. L. Jiang and X. Zhang. BCOSN: A blockchain-based decentralized online social network. *IEEE Transactions on Computational Social Systems*, 6(6):1454–1466, 2019. IEEE.

324. M. Jiang, X. Boix, G. Roig, J. Xu, L. Van Gool, and Q. Zhao. Learning to predict sequences of human visual fixations. *IEEE Transactions on Neural Networks and Learning Systems*, 27(6):1241–1252, 2016. IEEE.

325. L. Jin, Y. Chen, T. Wang, P. Hui, and A.V. Vasilakos. Understanding user behavior in online social networks: A survey. *IEEE Communications Magazine*, 51(9):144–150, 2013. IEEE.

326. S. Josephson and M.E. Holmes. Visual attention to repeated internet images: testing the scanpath theory on the world wide web. In *Proc. of the Symposium on Eye tracking Research & Applications (ETMA'02)*, pages 43–49, New Orleans, LA, USA, 2002. ACM.

327. A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov. Fasttext. zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*, 2016.

328. M. Jourdan, S. Blandin, L. Wynter, and P. Deshpande. Characterizing entities in the bitcoin blockchain. In *Proc. of the International Conference on Data Mining Workshops (ICDMW'18)*, pages 55–62, Singapore, 2018. IEEE.

329. M.A. Just and P.A. Carpenter. Eye fixations and cognitive processes. *Cognitive psychology*, 8(4):441–480, 1976. Elsevier.

330. V. Jyothsna and V.V. Rama Prasad. Article: A review of anomaly based intrusion detection systems. *International Journal of Computer Applications*, 28(7):26–35, 2011. IJCA Journal.

331. H. Kalodner, M. Möser, K. Lee, S. Goldfeder, M. Plattner, A. Chator, and A. Narayanan. Blocksci: Design and applications of a blockchain analysis platform. In *Proc. of the International Security Symposium (USENIX'20)*, pages 2721–2738, 2020. USENIX Association.

332. B. Kaluža and M. Luštrek. Fall detection and activity recognition methods for the confidence project: a survey. In *Proc. of che International Multi-Conference Information Society (IS'09)*, volume A, pages 22–25, Ljubljana, Slovenia, 2009.

333. Y.S. Kang, J. Min, J. Kim, and H. Lee. Roles of alternative and self-oriented perspectives in the context of the continued use of social network sites. *International Journal of Information Management*, 33(3):496–511, 2013. Elsevier.

334. K. K. Kapoor, K. Tamilmani, N. P Rana, P. Patil, Y. K Dwivedi, and S. Nerur. Advances in social media research: past, present and future. *Information Systems Frontiers*, 20(3):531–558, 2018.

335. D.M. Karantonis, M.R. Narayanan, M. Mathie, N.H. Lovell, and B.G. Celler. Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring. *IEEE Transactions on Information Technology in Biomedicine*, 10(1):156–167, 2006. IEEE.

336. F. Karim, S. Majumdar, H. Darabi, and S. Harford. Multivariate LSTM-FCNs for time series classification. *Neural Networks*, 116:237–245, 2019. Elsevier.

337. C. Karlof, N. Sastry, and D. Wagner. TinySec: a link layer security architecture for wireless sensor networks. In *Proc. of the International Conference on Embedded Networked Sensor Systems (SenSys'04)*, pages 162–175, Baltimore, MD, USA, 2004. ACM.

338. W. Kasper and M. Vela. Sentiment analysis for hotel reviews. In *Proc. of the International Computational Linguistics-Applications Conference*, volume 231527, pages 45–52, Jachranka, Poland, 2011.

339. A.L. Kavanaugh, D.D. Reese, J.M. Carroll, and M.B. Rosson. Weak ties in networked communities. *The Information Society*, 21(2):119–131, 2005.

340. K. Kaviya, C. Roshini, V. Vaidhehi, and J.D. Sweetlin. Sentiment analysis for restaurant rating. In *Proc. of the International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM'17)*, pages 140–145, Chennai, India, 2017. IEEE.

341. C. Ke-Jia, Z. Pei, Y. Zinong, and L. Yun. iBridge: Inferring bridge links that diffuse information across communities. *Knowledge-Based Systems*, 192, 2020. Elsevier.

342. Y. Keneshloo, S. Wang, E.-H. Sam Han, and N. Ramakrishnan. Predicting the Popularity of News Articles. In *Proc. of the International Conference on Data Mining (SDM'19)*, pages 441–449, Miami, FL, USA, 2016. SIAM.

343. M. Kennedy. If the rise of the TikTok dance and e-girl aesthetic has taught us anything, it's that teenage girls rule the internet right now: TikTok celebrity, girls and the Coronavirus crisis. *European Journal of Cultural Studies*, 23(6):1069–1076, 2020. SAGE.

344. D.J. Ketchen and C.L. Shook. The application of cluster analysis in strategic management research: an analysis and critique. *Strategic Management Journal*, 17(6):441–458, 1996. Wiley Online Library.

345. M. Keyvanpour, M. Ebrahimi, N.G. Nayebi, O. Ormandjieva, and C.Y. Suen. Automated identification of child abuse in chat rooms by using data mining. In O.E. Isafiade and A.B. Bagula, editors, *Data Mining Trends and Applications in Criminal Science and Investigations*, pages 245–274. 2016. IGI Global.

346. M.A. Khan and K. Salah. IoT security: Review, blockchain solutions, and open challenges. *Future Generation Computer Systems*, 82:395–411, 2018. Elsevier.

347. A. Khasawneh, M.K. Chalil, E. Dixon, P. Wiśniewski, H. Zinzow, and R. Roth. Examining the self-harm and suicide contagion effects related to the portrayal of the blue whale challenge on youtube and twitter (preprint). *JMIR Mental Health*, 2020.

348. A. Khasawneh, K.C. Madathil, H. Zinzow, P. Rosopa, G. Natarajan, K. Achuthan, and M. Narasimhan. Factors contributing to adolescents' and young adults' participation in web-based challenges: survey study. *JMIR Pediatrics and Parenting*, 4(1):e24988, 2021. JMIR Publications Inc., Toronto, Canada.

349. N.H. Khoa, P.T. Duy, H.D. Hoang, D.T.T. Hien, and V.H. Pham. Forensic analysis of TikTok application to seek digital artifacts on Android smartphone. In *Proc. of the International Conference on Computing and Communication Technologies (RIVF'20)*, pages 1–5, Ho Chi Minh City, Vietnam, 2020. IEEE.

350. H. Kim, R. Cetin-Atalay, and E. Gelenbe. G-Network Modelling Based Abnormal Pathway Detection in Gene Regulatory Networks. In *Proc. of the International Symposium on Computer and Information Sciences (ISCIS'11)*, pages 257–263, London, UK, 2011.

351. H. Kim and E. Gelenbe. Anomaly detection in gene expression via stochastic models of gene regulatory networks. *BMC Genomics*, 10(3):S26, 2009. BMC Genomics.

352. J. Kim, J. Bae, and M. Hastak. Emergency information diffusion on online social media during storm Cindy in US. *International Journal of Information Management*, 40:153–165, 2018. Elsevier.

353. J. Kim and M. Hastak. Social network analysis: Characteristics of online social networks after a disaster. *International Journal of Information Management*, 38(1):86–96, 2018.

354. J. Kim, U. Yun, E. Yoon, J. Chun-Wei Lin, and P. Fournier-Viger. One scan based high average-utility pattern mining in static and dynamic databases. *Future Generation Computer Systems*, 111:143–158, 2020. Elsevier.

355. Mitchell K.J., M. Wells, G. Priebe, and M.L. Ybarra. Exposure to websites that encourage self-harm and suicide: Prevalence rates and association with actual thoughts of self-harm and thoughts of suicide in the united states. *Journal of adolescence*, 37(8):1335–1344, 2014. Elsevier.

356. D. Klug. "It took me almost 30 minutes to practice this". Performance and Production Practices in Dance Challenge Videos on TikTok. *arXiv preprint arXiv:2008.13040*, 2020.

357. D. Klug, Y. Qin, M. Evans, and G. Kaufman. Trick and Please. A Mixed-Method Study On User Assumptions About the TikTok Algorithm. In *Proc. of the International Web Science Conference (WebSci'21)*, pages 84–92, Southampton, England, UK, 2021.

358. J. Knoll and J. Matthes. The Effectiveness of Celebrity Endorsements: A Meta-Analysis. *Journal of the Academy of Marketing Science*, 45(1):55–75, 2017. Springer.

359. Y. Ko, D. Chae, and S. Kim. Influence maximisation in social networks: A target-oriented estimation. *Journal of Information Science*, 44(5):671–682, 2018. SAGE Publications.

360. A. Konev, R. Khaydarova, M. Lapaev, L. Feng, L. Hu, M. Chen, and I. Bondarenko. CHPC: A complex semantic-based secured approach to heritage preservation and secure IoT-based museum processes. *Computer Communications*, 148:240–249, 2019.

361. T. Koohi-Var and M. Zahedi. Cross-domain graph based similarity measurement of workflows. *Journal of Big Data*, 5(1):1–16, 2018. Springer.

362. G. Kou and Y. Peng. An application of latent semantic analysis for text categorization. *International Journal of Computers Communications & Control*, 10(3):357–369, 2015. CCC Publications.

363. G. Kou, P. Yang, Y. Peng, F. Xiao, Y. Chen, and F.E. Alsaadi. Evaluation of feature selection methods for text classification with small datasets using multiple criteria decision-making methods. *Applied Soft Computing*, 86:105836, 2020. Elsevier.

364. Y. Kou, C.M. Gray, A.L. Toombs, and R.S. Adams. Understanding Social Roles in an Online Community of Volatile Practice: A Study of User Experience Practitioners on Reddit. *ACM Transactions on Social Computing*, 1(4):17:1–17:22, 2018. ACM.

365. E.L. Koua, A.M. MacEachren, and M. Kraak. Evaluating the usability of visualization methods in an exploratory geovisualization environment. *International Journal of Geographical Information Science*, 20(4):425–448, 2006. Taylor & Francis.

366. P. Kouvaris, E. Pirogova, H. Sanadhya, A. Asuncion, and A. Rajagopal. Text enhanced recommendation system model based on yelp reviews. *SMU Data Science Review*, 1(3):8, 2018.

367. N. Kumar and I. Benbasat. Research note: the influence of recommendations and consumer reviews on evaluations of websites. *Information Systems Research*, 17(4):425–439, 2006. INFORMS.

368. N. Kumar, A. Singh, A. Handa, and S.K. Shukla. Detecting Malicious Accounts on the Ethereum Blockchain with Supervised Learning. In *Proc. of the International Symposium on Cyber Security Cryptography and Machine Learning (CSCML'20)*, pages 94–109, Be'er Sheva, Israel, 2020. Springer.

369. S. Kumar, J. Cheng, and J. Leskovec. Antisocial Behavior on the Web: Characterization and Detection. In *Proc. of the International Conference on World Wide Web Companion*, page 947–950, Geneva, Switzerland, 2017. International World Wide Web Conferences Steering Committee.

370. S. Kumar, W.L. Hamilton, J. Leskovec, and D. Jurafsky. Community Interaction and Conflict on the Web. In *Proc. of the World Wide Web Conference (WWW 2018)*, pages 933–943, Lyon, France, 2018. ACM.

371. S. Kumar, BS Panda, and D. Aggarwal. Community detection in complex networks using network embedding and gravitational search algorithm. *Journal of Intelligent Information Systems*, 57(1):51–72, 2021. Springer.

372. M. Kummerer, T.S. Wallis, L.A. Gatys, and M. Bethge. Understanding low-and high-level contributions to fixation prediction. In *Proc. of the IEEE International Conference on Computer Vision (ICCV'17)*, pages 4789–4798, Venezia, Italy, 2017.

373. B. Kwolek and M. Kepski. Human fall detection on embedded platform using depth maps and wireless accelerometer. *Computer Methods and Programs in Biomedicine*, 117(3):489–501, 2014.

374. K.H. Kwon, C. Chris Bang, M. Egnoto, and H. Raghav Rao. Social media rumors as improvised public opinion: semantic network analyses of twitter discourses during korean saber rattling 2013. *Asian Journal of Communication*, 26(3):201–222, 2016. Routledge.

375. L. Kiffer and D. Levin and A. Mislove. Analyzing ethereum's contract topology. In *Proc. of the Internet Measurement Conference (IMC'18)*, pages 494–499, Boston, MA, USA, 2018. ACM.

376. N. Labraoui, M. Gueroui, and L. Sekhri. A risk-aware reputation-based trust management in wireless sensor networks. *Wireless Personal Communications*, 87(3):1037–1055, 2016.

377. S. Lahiri, S.R. Choudhury, and C. Caragea. Keyword and keyphrase extraction using centrality measures on collocation networks. *arXiv preprint arXiv:1401.6571*, 2014.

378. C.F. Lai, S.Y. Chang, H.C. Chao, and Y.M. Huang. Detection of cognitive injured body region using multiple triaxial accelerometers for elderly falling. *IEEE Sensors Journal*, 11(3):763–770, 2010. IEEE.

379. Y. Lama, D. Hu, A. Jamison, S.C. Quinn, and D.A. Broniatowski. Characterizing trends in Human Papillomavirus Vaccine discourse on Reddit (2007-2015): an observational study. *JMIR Public Health and Surveillance*, 5(1):e12480, 2019. JMIR Publications.

380. L. Lao, Z. Li, S. Hou, B. Xiao, S. Guo, and Y. Yang. A survey of IoT applications in blockchain systems: Architecture, consensus, and traffic modeling. *ACM Computing Surveys (CSUR)*, 53(1):1–32, 2020. ACM New York, NY, USA.

381. C.E. Lawson. Platform vulnerabilities: harassment and misogynoir in the digital attack on Leslie Jones. *Information, Communication & Society*, 21(6):818–833, 2018. Taylor & Francis.

382. A. Leavitt and J.A. Clark. Upvoting hurricane Sandy: event-based news production processes on a social news site. In *Proc. of the International Conference on Human Factors in Computing Systems (SIGCHI'14)*, pages 1495–1504, Toronto, Canada, 2014. ACM.

383. C. Lee, S. Maharjan, K. Ko, and J.W.K. Hong. Toward Detecting Illegal Transactions on Bitcoin Using Machine-Learning Methods. In *Proc. of the International Conference on Blockchain and Trustworthy Systems (BlockSys'19)*, pages 520–533, Guangzhou, China, 2019. Springer.

384. C. Lee, S. Maharjan, K. Ko, J. Woo, and J.W.K. Hong. Machine Learning Based Bitcoin Address Classification. In *Proc. of the International Conference on Blockchain and Trustworthy Systems (BlockSys'20)*, pages 517–531, Dali, China, 2020. Springer.

385. K. Lee, J. Ham, S. Yang, and C. Koo. Can You Identify Fake or Authentic Reviews? An fsQCA Approach. In *Information and Communication Technologies in Tourism 2018*, pages 214–227, Jonkoping, Sweden, 2018. Springer.

386. X. Lei and X. Qian. Rating prediction via exploring service reputation. In *Proc. of the International Workshop on Multimedia Signal Processing (MMSP'15)*, pages 1–6, Xiamen, China, 2015. IEEE.

387. B.I. Lerman, S.P. Lewis, M. Lumley, G.J. Grogan, C.C. Hudson, and E. Johnson. Teen depression groups on Facebook: a content analysis. *Journal of Adolescent Research*, 32(6):719–741, 2017. SAGE Publications Sage CA: Los Angeles, CA.

388. J. Leskovec, L.A. Adamic, and B.A. Huberman. The dynamics of viral marketing. *ACM Transactions on the Web*, 1(1):5, 2007. ACM.

389. J. Li, L. Huang, T. Bai, Z. Wang, and H. Chen. CDBIA: A dynamic community detection method based on incremental analysis. In *Proc. of the International Conference on Systems and Informatics (ICSAI'12)*, pages 2224–2228, Yantai, China, 2012. IEEE.

390. J. Li, L. Su, B. Wu, J. Pang, C. Wang, Z. Wu, and Q. Huang. Webpage saliency prediction with multi-features fusion. In *Proc. of the IEEE International Conference on Image Processing (ICIP'16)*, pages 674–678, Phoenix, Arizona, USA, 2016. IEEE.

391. M. Li, X. Wang, K. Gao, and S. Zhang. A survey on information diffusion in online social networks: Models and methods. *Information*, 8(4):118, 2017. Multidisciplinary Digital Publishing Institute.

392. M.X. Li, C.H. Tan, K.K. Wei, and K.L. Wang. Sequentiality of Product Review Information Provision: An Information Foraging Perspective. *MIS Q.*, 41(3):867–892, 2017. Management Information Systems Research Center.

393. W. Li, S. Tug, W. Meng, and Y. Wang. Designing collaborative blockchained signature-based intrusion detection in iot environments. *Future Generation Computer Systems*, 96:481–489, 2019. Elsevier.

394. X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen. A survey on the security of blockchain systems. *Future Generation Computer Systems*, 107:841–853, 2020. Elsevier.

395. Y. Li, Y. Cai, H. Tian, G. Xue, and Z. Zheng. Identifying Illicit Addresses in Bitcoin Network. In *Proc. of the International Conference on Blockchain and Trustworthy Systems (BlockSys '19)*, pages 99–111, Guangzhou, China, 2020. Springer.

396. Y. Li, M. Guan, P. Hammond, and L.E. Berrey. Communicating COVID-19 information on TikTok: a content analysis of TikTok videos from official accounts featured in the COVID-19 information hub. *Health Education Research*, 2021. Oxford University Press.

397. Y. Li and Y. Zhang. Webpage Saliency Prediction with Two-Stage Generative Adversarial Networks. *arXiv preprint arXiv:1805.11374*, 2018.

398. Y. Lim and B. Van Der Heide. Evaluating the wisdom of strangers: The perceived credibility of online consumer reviews on Yelp. *Journal of Computer-Mediated Communication*, 20(1):67–82, 2014. Oxford University Press.

399. J. Lin, E.J. Keogh, A.W. Fu, and H. Van Herle. Approximations to magic: Finding unusual medical time series. In *Proc. of the 18th IEEE Symposium on Computer-Based Medical Systems (CBMS 2005), 23-24 June 2005, Dublin, Ireland*, pages 329–334, 2005. IEEE Computer Society.

400. J. Lin, Z. Shen, and C. Miao. Using blockchain technology to build trust in sharing LoRaWAN IoT. In *Proc. of the International Conference on Crowd Science and Engineering (ICCSE'17)*, pages 38–43, Beijing, China, 2017. ACM.

401. X. Lin and X. Wang. Examining gender differences in people's information-sharing decisions on social networking sites. *International Journal of Information Management*, 50:45–56, 2020. Elsevier.

402. Y.J. Lin, P.W. Wu, C.H. Hsu, I.P. Tu, and S.W. Liao. An evaluation of bitcoin address classification based on transaction history summarization. In *Proc. of the IEEE International Conference on Blockchain and Cryptocurrency (ICBC'19)*, pages 302–310, Seoul, South Korea, 2019. IEEE.

403. J.R. Linabary and D.J. Corple. Privacy for whom?: A feminist intervention in online research practice. *Information, Communication & Society*, 22(10):1447–1463, 2019. Taylor & Francis.

404. W. Lippmann. *Public Opinion*. 1922. Macmillan.

405. H. Liu and P. Singh. ConceptNet — a practical commonsense reasoning tool-kit. *BT technology journal*, 22(4):211–226, 2004. Springer.

406. H. Liu, D. Xu, Q. Huang, W. Li, M. Xu, and S. Lin. Semantically-based human scanpath estimation with HMMs. In *Proc. of the IEEE International Conference on Computer Vision (ICCV'13)*, pages 3232–3239, Sydney, NSW, Australia, 2013.

407. L. Liu, B. Chen, B. Qu, L. He, and X. Qiu. Data driven modeling of continuous time information diffusion in social networks. In *Proc. of the International Conference on Data Science in Cyberspace (DSC'17)*, pages 655–660, Shenzhen, China, 2017. IEEE.

408. N. Liu and J. Han. A deep spatial contextual long-term recurrent convolutional network for saliency detection. *IEEE Transactions on Image Processing*, 27(7):3264–3274, 2018. IEEE.

409. R. Liu, S. Wan, Z. Zhang, and X. Zhao. Is the introduction of futures responsible for the crash of Bitcoin? *Finance Research Letters*, 34:101259, 2020. Elsevier.

410. J. Lu, S. Sridhar, R. Pandey, M.A. Hasan, and G. Mohler. Redditors in recovery: text mining Reddit to investigate transitions into drug addiction. *arXiv preprint arXiv:1903.04081*, 2019.

411. M. Luca. Reviews, reputation, and revenue: The case of Yelp.com. *Harvard Business School Working Paper*, 12-016, 2016.

412. M. Luca and G. Zervas. Fake it till you make it: Reputation, competition, and yelp review fraud. *Management Science*, 62(12):3412–3427, 2016.

413. A. Lujain, H. Alhamarna, Y. AlWawi, Y. ElSayed, and H. Harb. Analysis of the representation of the 2019 Lebanese protests and the 2020 Beirut explosion on TikTok. *KIU Interdisciplinary Journal of Humanities and Social Sciences*, 1(3):53–72, 2020.

414. X. Luo. Quantifying the long-term impact of negative word of mouth on cash flows and stock prices. *Marketing Science*, 28(1):148–165, 2009. INFORMS.

415. J. Ma and Y. Luo. The classification of rumour standpoints in online social network based on combinatorial classifiers. *Journal of Information Science*, 46(2):191–204, 2020. SAGE.

416. X. Ma and H. Xue. Intelligent smart city parking facility layout optimization based on intelligent IoT analysis. *Computer Communications*, 153:145–151, 2020.

417. I. Maduako, M. Wachowicz, and T. Hanson. STVG: an evolutionary graph framework for analyzing fast-evolving networks. *Journal of Big Data*, 6(1):1–24, 2019. Springer.

418. W. Maharani, Adiwijaya, and A.A. Gozali. Degree centrality and eigenvector centrality in twitter. In *Proc. of the International Conference on Telecommunication Systems Services and Applications (TSSA'14)*, pages 1–5, Bali, Indonesia, 2014. IEEE.

419. R. Mahmoud, T. Yousuf, F. Aloul, and I. Zualkernan. Internet of things (IoT) security: Current status, challenges and prospective measures. In *Proc. of the International Confer-*

ence for Internet Technology and Secured Transactions (ICITST'15), pages 336–341, London, United Kingdom, 2015. IEEE.

420. A. Mahmoudi, A.A. Bakar, M. Sookhak, and M.R. RYaakub. A Temporal User Attribute-Based Algorithm to Detect Communities in Online Social Networks. *IEEE Access*, 8:154363–154381, 2020. IEEE.
421. H. Mahyar, R. Hasheminezhad, E. Ghalebi, A. Nazemian, R. Grosu, A. Movaghar, and H. R. Rabiee. Identifying central nodes for information flow in social networks using compressive sensing. *Social Network Analysis and Mining*, 8(1):1–24, 2018. Springer.
422. M. Maia, J. Almeida, and V. Almeida. Identifying user behavior in online social networks. In *Proc. of the International Workshop on Social Network Systems*, pages 1–6, Glasgow, Scotland, UK, 2008. ACM.
423. A. Majeed and A. Al-Yasiri. Formulating a global identifier based on actor relationship for the internet of things. In *Interoperability, Safety and Security in IoT*, pages 79–91. Springer, 2016.
424. K. Malang, S. Wang, Y. Lv, and A. Phaphuangwittayakul. Skeleton Network Extraction and Analysis on Bicycle Sharing Networks. *International Journal of Data Warehousing and Mining*, 16(3):146–167, 2020. IGI Global.
425. J. Malbon. Taking fake online consumer reviews seriously. *Journal of Consumer Policy*, 36(2):139–157, 2013. Springer.
426. B. Markines, C. Cattuto, and F. Menczer. Social spam detection. In *Proc. of the International Workshop on Adversarial Information Retrieval on the Web (AirWeb'09)*, pages 41–48, Madrid, Spain, 2009.
427. M. Marples. The 'devious lick" TikTok challenge has students stealing toilets and vandalizing bathrooms, 2021. available online at: `https://www.cnn.com/2021/09/18/health/devious-licks-tiktok-challengewellness/index.html`.
428. G. Marra, F. Ricca, G. Terracina, and D. Ursino. Information Diffusion in a Multi-Social-Network Scenario: A framework and an ASP-based analysis. *Knowledge and Information Systems*, 48(3):619–648, 2016. Springer.
429. S. Marti and H. Garcia-Molina. Taxonomy of trust: Categorizing P2P reputation systems. *Computer Networks*, 50(4):472–484, 2006. Elsevier.
430. S. Maslov and S. Redner. Promise and pitfalls of extending Google's PageRank algorithm to citation networks. *Journal of Neuroscience*, 28(44):11103–11105, 2008. Society for Neuroscience.
431. M.J. Mathie, A.C.F. Coster, N.H. Lovell, and B.G. Celler. Accelerometry: providing an integrated, practical method for long-term, ambulatory monitoring of human movement. *Physiological measurement*, 25(2):R1, 2004. IOP Publishing.
432. J.M. Matías, T. Rivas, JE Martín, and J. Taboada. A machine learning methodology for the analysis of workplace accidents. *International Journal of Computer Mathematics*, 85(3-4):559–578, 2008. Taylor & Francis.
433. J.N. Matias. Going dark: Social factors in collective action against platform operators in the Reddit blackout. In *Proc. of the International Conference on Human Factors in Computing Systems (ACM CHI 2016)*, pages 1138–1151, San Jose, CA, USA, 2016. ACM.

434. S. Mazhari, S.M. Fakhrahmad, and H. Sadeghbeygi. A user-profile-based friendship recommendation solution in social networks. *Journal of Information Science*, 41(3):284–295, 2015. SAGE.

435. M. McPherson, L. Smith-Lovin, and J.M. Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27:415–444, 2001. JSTOR.

436. A.N. Medvedev, R. Lambiotte, and J.C. Delvenne. The Anatomy of Reddit: An Overview of Academic Research. In *Dynamics On and Of Complex Networks III*, pages 183–204, Cham, 2019. Springer International Publishing.

437. J. Meese. "It belongs to the Internet": Animal images, attribution norms and the politics of amateur media production. *M/C Journal*, 17(2):1–3, 2014. M/C.

438. K.Z. Meral. Social Media Short Video-Sharing TikTok Application and Ethics: Data Privacy and Addiction Issues. In *Multidisciplinary Approaches to Ethics in the Digital Era*, pages 147–165. IGI Global, 2021.

439. O. Le Meur and T. Baccino. Methods for comparing scanpaths and saliency maps: strengths and weaknesses. *Behavior research methods*, 45(1):251–266, 2013. Springer.

440. K. De Miguel, A. Brunete, M. Hernando, and E. Gambao. Home camera-based fall detection system for the elderly. *Sensors*, 17(12):2864, 2017. MDPI.

441. R. Mihalcea and P. Tarau. Textrank: Bringing order into text. In *Proc. of the International Conference on Empirical Methods in Natural Language Processing (EMNLP'14)*, pages 404–411, Qatar, Qatar, 2004. Association for Computational Linguistics.

442. A.G. Miller. WordNet: A lexical database for English. *Communications of the ACM*, 38(11):39–41, 1995.

443. M. Miller, T. Banerjee, R. Muppalla, W. Romine, and A. Sheth. What are people tweeting about Zika? An exploratory study concerning its symptoms, treatment, transmission, and prevention. *JMIR Public Health and Surveillance*, 3(2):e38, 2017. JMIR Publications.

444. A. Mislove, M. Marcon, K.P. Gummadi, P. Druschel, and B. Bhattacharjee. Measurement and analysis of online social networks. In *Proc. of the ACM SIGCOMM International Conference on Internet Measurement (IMC'07)*, pages 29–42, San Diego, CA, USA, 2007. ACM.

445. A. Mislove, B. Viswanath, K.P. Gummadi, and P. Druschel. You are who you know: inferring user profiles in online social networks. In *Proc. of the third ACM International Conference on Web Search and Data Mining (WSDM'10)*, pages 251–260, New York, NY, USA, 2010. ACM Press.

446. L. Mitchell. *A Phenomenological study of social media: boredom and interest on Facebook, Reddit, and 4chan*. 2012. University of Victoria, British Columbia, Canada.

447. Thomas Moellers. IOTA-based Business Model Configurations. *https://www.alexandria.unisg.ch/257117/*, 2018.

448. S.A. Moosavi, M. Jalali, N. Misaghian, S. Shamshirband, and M.H. Anisi. Community detection in social networks using user frequent pattern mining. *Knowledge and Information Systems*, 51(1):159–186, 2017. Springer.

449. D. Morrison and C. Hayes. Here, have an upvote: Communication behaviour and karma on Reddit. *Informatik*, pages 2258–2268, 2013. Gesellschaft für Informatik eV.

450. M. Mubashir, L. Shao, and L. Seed. A survey on fall detection: Principles and approaches. *Neurocomputing*, 100:144–152, 2013. Elsevier.

451. A. Mukherjee, V. Venkataraman, B. Liu, and N. Glance. What yelp fake review filter might be doing? In *Proc. of the International AAAI Conference on Weblogs and Social Media (ICDSM'13)*, Boston, MA, USA, 2013.

452. J.L. Muñoz-Sánchez, C. Delgado, A. Sánchez-Prada, E. Parra-Vidales, D. De Leo, and M. Franco-Martín. Facilitating factors and barriers to the use of emerging technologies for suicide prevention in europe: multicountry exploratory study. *JMIR mental health*, 5(1):e7784, 2018. JMIR Publications Inc., Toronto, Canada.

453. C. Murray, L. Mitchell, J. Tuke, and M. Mackay. Symptom extraction from the narratives of personal experiences with COVID-19 on Reddit. *arXiv preprint arXiv:2005.10454*, 2020.

454. N.G. Nair, A. Saeed, M.I. Biswas, M. Abu-Tair, P.K. Chouhan, I. Cleland, J. Rafferty, C. Nugent, P. Morrow, and M.H. Zoualfaghari. Evaluation of an IoT Framework for a Workplace Wellbeing Application. In *Proc. of the International Conference on Ubiquitous Intelligence and Computing (UIC'19)*, pages 1783–1788, Leicester, UK, 2019. IEEE.

455. M. Nakayama and Y. Wan. The cultural impact on social commerce: A sentiment analysis on yelp ethnic restaurant reviews. *Information & Management*, 56(2):271–279, 2019. Elsevier.

456. H. Nam, Y.V. Joshi, and P.K. Kannan. Harvesting brand information from social tags. *Journal of Marketing*, 81(4):88–108, 2017.

457. B. K. Narayanan and M. Nirmala. Adult content filtering: Restricting minor audience from accessing inappropriate Internet content. *Education and Information Technologies*, 23(6):2719–2735, 2018. Springer.

458. R. Navigli and S.P. Ponzetto. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250, 2012. Elsevier.

459. A. Neal, M.A. Griffin, and P.M. Hart. The impact of organizational climate on safety climate and individual behavior. *Safety science*, 34(1-3):99–109, 2000. Elsevier.

460. N. Nesa, T. Ghosh, and I. Banerjee. Non-parametric sequence-based learning approach for outlier detection in iot. *Future Generation Computer Systems*, 82:412–421, 2018. Elsevier.

461. E. Newell, D. Jurgens, H.M. Saleem, H. Vala, J. Sassine, C. Armstrong, and D. Ruths. User Migration in Online Social Networks: A Case Study on Reddit During a Period of Community Unrest. In *Proc. of the International Conference on Web and Social Media (ICWSM 2016)*, pages 279–288, Cologne, Germany, 2016. AAAI.

462. M.E.J. Newman. Clustering and preferential attachment in growing networks. *Physical Review E*, 64(2):025102, 2001. APS.

463. M.E.J. Newman. Properties of highly clustered networks. *Physical Review E*, 68(2):026121, 2003. APS.

464. M.E.J. Newman. Power laws, Pareto distributions and Zipf's law. *Contemporary physics*, 46(5):323–351, 2005. Taylor & Francis.

465. N. Newman, R. Fletcher, A. Schulz, S. Andı, and R.K. Nielsen. Reuters Institute Digital News Report 2020. Reuters Institute and University of Oxford, 2020. available online at: https://reutersinstitute.politics.ox.ac.uk/sites/default/files/2020-06/DNR_2020_FINAL.pdf.

466. A. Neyaz, A. Kumar, S. Krishnan, J. Placker, and Q. Liu. Security, privacy and steganographic analysis of FaceApp and TikTok. *International Journal of Computer Science and Security*, 14(2):38–59, 2020.

467. L.H.X. Ng, J.Y.H. Tan, J.H. Darryl, and R.K.W. Lee. Will you dance to the challenge? predicting user participation of TikTok challenges. In *Proc. of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM'21)*, pages 356–360, Virtual event, The Netherlands, 2021.

468. T.V. Nguyen, N.T. Tran, and S. Le Thanh. An anomaly-based network intrusion detection system using deep learning. In *Proc. of the 2017 International Conference on System Science and Engineering (ICSSE)*, pages 210–214, Ho Chi Minh City, Vietnam, 2017. IEEE.

469. V.A. Nguyen, T.H. Le, and T.H. Nguyen. Single camera based fall detection using motion and human shape features. In *Proc. of the Symposium on Information and Communication Technology (SoICT'16)*, pages 339–344, Ho Chi Minh, Vietnam, 2016.

470. S. Nicolazzo, A. Nocera, D. Ursino, and L. Virgili. A privacy-preserving approach to prevent feature disclosure in an iot scenario. In *Future Generation Computer Systems*, volume 105, pages 1–8, 2019. IEEE.

471. S. Nicolazzo, A. Nocera, D. Ursino, and L. Virgili. A Privacy-Preserving Approach to Prevent Feature Disclosure in an IoT Scenario. *Future Generation Computer Systems*, 105:502–512, 2020. Elsevier.

472. K. Nishimoto and K. Matsuda. Informal communication support media for encouraging knowledge-sharing and creation in a community. *International Journal of Information Technology & Decision Making*, 6(03):411–426, 2007. World Scientific.

473. A. Nocera and D. Ursino. PHIS: a system for scouting potential hubs and for favoring their "growth" in a Social Internetworking Scenario. *Knowledge-Based Systems*, 36:288–299, 2012. Elsevier.

474. P. Nokhiz and F. Li. Understanding rating behavior based on moral foundations: The case of Yelp reviews. In *Proc. of the International Conference on Big Data (Big Data 2017)*, pages 3938–3945, Boston, MA, USA, 2017. IEEE.

475. B. Nour, K. Sharif, F. Li, H. Moungla, and Y. Liu. A unified hybrid information-centric naming scheme for IoT applications. *Computer Communications*, 150:103–114, 2020.

476. O. Novo. Blockchain meets IoT: An architecture for scalable access management in IoT. *IEEE Internet of Things Journal*, 5(2):1184–1195, 2018. IEEE.

477. J.H. Oh, J.Y. Hong, and J.G. Baek. Oversampling method using outlier detectable generative adversarial network. *Expert Systems with Applications*, 133:1–8, 2019. Elsevier.

478. Y. Okada, K. Masui, and Y. Kadobayashi. Proposal of Social Internetworking. In *Proc. of the International Human.Society@Internet Conference (HSI 2005)*, pages 114–124, Asakusa, Tokyo, Japan, 2005. Lecture Notes in Computer Science, Springer.

479. R. S. Olson and Z. P. Neal. Navigating the massive world of reddit: Using backbone networks to map user interests in social media. *PeerJ Computer Science*, 1:e4, 2015. PeerJ Inc.

480. T. O'Neill. 'Today I Speak': Exploring How Victim-Survivors Use Reddit. *International Journal for Crime, Justice and Social Democracy*, 7(1):44, 2018. Queensland University of Technology.

481. P. Oser, F. Kargl, and S. Lüders. Identifying devices of the internet of things using machine learning on clock characteristics. In *International conference on security, privacy and anonymity in computation, communication and storage*, pages 417–427. Springer, 2018.

482. P. Otte, M. de Vos, and J. Pouwelse. TrustChain: A Sybil-resistant scalable blockchain. *Future Generation Computer Systems*, 107(48):770–780, 2017. Elsevier.

483. A.T. Özdemir and B. Barshan. Detecting falls with wearable sensors using machine learning techniques. *Sensors*, 14(6):10691–10708, 2014. MDPI.

484. L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank Citation Ranking: Bringing Order to the Web. In *Proc. of the Seventh International World-Wide Web Conference (WWW 1998)*, pages 161–172, Brisbane, Australia, 1998. Elsevier.

485. X. Page, P. Wisniewski, B.P. Knijnenburg, and M. Namara. Social media's have-nots: an era of social disenfranchisement. *Internet Research*, 2018. Emerald Publishing Limited.

486. H. Haddad Pajouh, R. Javidan, R. Khayami, D. Ali, and K.R. Choo. A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in iot backbone networks. *IEEE Transactions on Emerging Topics in Computing*, pages 1–1, 2019. IEEE.

487. L. Palopoli, D. Saccà, G. Terracina, and D. Ursino. Uniform techniques for deriving similarities of objects and subschemes in heterogeneous databases. *IEEE Transactions on Knowledge and Data Engineering*, 15(2):271–294, 2003.

488. J. Pan, C. Canton Ferrer, K. McGuinness, N.E. O'Connor, J. Torres, E. Sayrol, and X. Giro i Nieto. Salgan: Visual saliency prediction with generative adversarial networks. *arXiv preprint arXiv:1701.01081*, 2017.

489. N. Pannurat, S. Thiemjarus, and E. Nantajeewarawat. A hybrid temporal reasoning framework for fall monitoring. *IEEE Sensors Journal*, 17(6):1749–1759, 2017. IEEE.

490. A. Parikh, C. Behnke, M. Vorvoreanu, B. Almanza, and D. Nelson. Motives for reading and articulating user-generated restaurant reviews on yelp. com. *Journal of Hospitality and Tourism Technology*, 5(2):160–176, 2014.

491. A.A. Parikh, C. Behnke, B. Almanza, D. Nelson, and M. Vorvoreanu. Comparative content analysis of professional, semi-professional, and user-generated restaurant reviews. *Journal of Foodservice Business Research*, 20(5):497–511, 2017.

492. J. Pater and E. Mynatt. Defining digital self-harm. In *Proc. of the ACM Conference on Computer Supported Cooperative Work and Social Computing (CSCW'17)*, pages 1501–1513, Potland, Oregon, USA, 2017.

493. G.C. Patton, C. Coffey, H. Romaniuk, A. Mackinnon, J.B. Carlin, , L. Degenhardt, C.A. Olsson, and P. Moran. The prognosis of common mental disorders in adolescents: a 14-year prospective cohort study. *The Lancet*, 383(9926):1404–1411, 2014. Elsevier.

494. T. Pay. Totally automated keyword extraction. In *Proc. of the International Conference on Big Data (Big Data 2016)*, pages 3859–3863, Washington, D.C., USA, 2016. IEEE.

495. K. Pearson. Note on Regression and Inheritance in the Case of Two Parents. *Proceedings of the Royal Society of London*, 58:240–242, 1895. The Royal Society.

496. G. Peeters and J. Czapinski. Positive-negative asymmetry in evaluations: The distinction between affective and informational negativity effects. *European review of social psychology*, 1(1):33–60, 1990. Taylor & Francis.

497. M. Pennacchiotti and A. Popescu. Democrats, republicans and starbucks afficionados: user classification in Twitter. In *Proc. of the International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 430–438, San Diego, CA, USA, 2011. ACM.

498. R.G. Pensa, G. Di Blasi, and L. Bioglio. Network-aware privacy risk estimation in online social networks. *Social Network Analysis and Mining*, 9(1):1–15, 2019. Springer.

499. A. Perrig, R. Szewczyk, J.D. Tygar, V. Wen, and D.E. Culler. SPINS: Security protocols for sensor networks. *Wireless networks*, 8(5):521–534, 2002. Springer.

500. R.J. Peters, A. Iyer, L. Itti, and C. Koch. Components of bottom-up gaze allocation in natural images. *Vision research*, 45(18):2397–2416, 2005. Elsevier.

501. R.C. Phillips and D. Gorse. Predicting cryptocurrency price bubbles using social media data and epidemic modelling. In *Proc. of the International Symposium Series on Computational Intelligence (SSCI'17)*, pages 1–7, Honolulu, HI, USA, 2017. IEEE.

502. R. Di Pietro, X. Salleras, M. Signorini, and E. Waisbard. A blockchain-based Trust System for the Internet of Things. In *Proc. of the ACM International Symposium on Access Control Models and Technologies (SACMAT'18)*, pages 77–83, Indianapolis, IN, USA, 2018. ACM.

503. S. Popov. The tangle. *White paper*, 1:3, 2018.

504. D. Praveena and P. Rangarajan. A machine learning application for reducing the security risks in hybrid cloud networks. *Multimedia Tools and Applications*, 79(7-8):5161–5173, 2020. Springer.

505. G. Pujolle. An autonomic-oriented architecture for the internet of things. In *Proc. of the IEEE John Vincent Atanasoff 2006 International Symposium on Modern Computing (JVA'06)*, pages 163–168, Sofia, Bulgaria, 2006. IEEE.

506. M. Qin, D. Jin, K.Lei, B. Gabrys, and K. Musial-Gabrys. Adaptive community detection incorporating topology and content in social networks. *Knowledge-Based Systems*, 161:342–356, 2018. Elsevier.

507. D. Qiu, H. Li, and Y. Li. Identification of active valuable nodes in temporal online social network with attributes. *International Journal of Information Technology & Decision Making*, 13(04):839–864, 2014. World Scientific.

508. J. Qiu, Y. Li, and Z. Lin. Does Social Commerce Work in Yelp? An Empirical Analysis of Impacts of Social Relationship on the Purchase Decision-making. In *Proc. of the Pacific Asia Conference on Information Systems (PACIS'18)*, page 16, Yokohama, Japan, 2018.

509. J. Qiu, Y. Li, and Z. Lin. Detecting Social Commerce: An Empirical Analysis on Yelp. *Journal of Electronic Commerce Research*, 21(3):168–179, 2020. Journal of Electronic Commerce Research.

510. J. Qiu, Q. Wu, G. Ding, Y. Xu, and S-Feng. A survey of machine learning for big data processing. *EURASIP Journal on Advances in Signal Processing*, 2016(1):67, 2016. Springer.

511. Z. Qiyang and H. Jung. Learning and sharing creative skills with short videos: A case study of user behavior in tiktok and bilibili. In *Proc. of the International Association of Societies of Design Research Conference (IASDR'19)*, Manchester, UK, 2019.

512. J. Quevedo, M. Antunes, D. Corujo, D. Gomes, and R.L. Aguiar. On the application of contextual iot service discovery in information centric networks. *Computer Communications*, 89:117–127, 2016. Elsevier.

513. E. Rahm and P.A. Bernstein. A survey of approaches to automatic schema matching. *VLDB Journal*, 10(4):334–350, 2001.

514. T.B.A. Rakib and L.K. Soon. Using the Reddit corpus for cyberbully detection. In *Proc. of the Asian Conference on Intelligent Information and Database Systems (ACIIDS'18)*, pages 180–189, Dong Hoi City, Vietnam, 2018. Springer.

515. G. Ramponi, M. Brambilla, S. Ceri, F. Daniel, and M. Di Giovanni. Content-based characterization of online social communities. *Information Processing & Management*, page 102133, 2019. Elsevier.

516. S. Rani and M. Mehrotra. Community detection in social networks: literature review. *Journal of Information & Knowledge Management*, 18(02):1–28, 2019. World Scientific.

517. S. R. Ranjan. Centrality measures: A tool to identify key actors in social networks. In *Principles of Social Networking*, pages 1–27. Springer Singapore, 2021.

518. S. Ranshous, C.A. Joslyn, S. Kreyling, K. Nowak, N.F. Samatova, C.L. West, and S. Winters. Exchange pattern mining in the bitcoin transaction directed hypergraph. In *Proc. of the International Conference on Financial Cryptography and Data Security (FC'17)*, pages 248–263, Malta, 2017. Springer.

519. M. Rehman, N. Javaid, M. Awais, M. Imran, and N. Naseer. Cloud based secure service providing for IoTs using blockchain. In *Proc. of the IEEE Global Communications Conference (GLOBCOM 2019)*, pages 1–7, Puako, Hawaii, USA, 2019.

520. F. Reid and M. Harrigan. An analysis of anonymity in the bitcoin system. In *Security and privacy in social networks*, pages 197–223. Springer, 2013.

521. A. Reihanian, M.R. Feizi-Derakhshi, and H.S. Aghdasi. Overlapping community detection in rating-based social networks through analyzing topics, ratings and links. *Pattern Recognition*, 81:370–387, 2018. Elsevier.

522. M. Assens Rein, X. Giro i Nieto, K. McGuinness, and N.E. O'Connor. Saltinet: Scan-path prediction on 360 degree images using saliency volumes. In *Proc. of the IEEE International Conference on Computer Vision Workshops (ICCV'17)*, pages 2331–2338, Venezia, Italy, 2017. IEEE.

523. N. Riche, M. Duvinage, M. Mancas, B. Gosselin, and T. Dutoit. Saliency and human fixations: State-of-the-art and study of comparison metrics. In *Proc. of the IEEE International Conference on Computer Vision (ICCV'13)*, pages 1153–1160, Sidney, Australia, 2013. IEEE.

524. H. Rimminen, J. Lindström, M. Linnavuo, and R. Sepponen. Detection of falls among the elderly by a floor sensor using the electric near field. *IEEE Transactions on Information Technology in Biomedicine*, 14(6):1475–1476, 2010. IEEE.

525. A. Robert, J.M. Suelves, M. Armayones, and S. Ashley. Internet use and suicidal behaviors: internet as a threat or opportunity? *Telemedicine and e-Health*, 21(4):306–311, 2015. Mary Ann Liebert.

526. F. Role and M. Nadif. Beyond cluster labeling: Semantic interpretation of clusters' contents using a graph representation. *Knowledge-Based Systems*, 56:141–155, 2014. Elsevier.

527. S. Romeo, A. Tagarelli, and D. Ienco. Semantic-based multilingual document clustering via tensor modeling. *Available at* `https://hal.archives-ouvertes.fr/hal-01130094/`, 2014.

528. S. Rose, D. Engel, N. Cramer, and W. Cowley. Automatic keyword extraction from individual documents. *Text mining: applications and theory*, 1:1–20, 2010. Wiley, New York.

529. R. Roth, P. Ajithkumar, G. Natarajan, K. Achuthan, P. Moon, H. Zinzow, and K.C. Madathil. A Study of Adolescents' and Young Adults' TikTok Challenge Participation in South India. *Human Factors in Healthcare*, page 100005, 2022. Elsevier.

530. M. Roy, N. Moreau, C. Rousseau, A. Mercier, A. Wilson, and L. Atlani-Duault. Ebola and localized blame on social media: analysis of Twitter and Facebook conversations during the 2014–2015 Ebola epidemic. *Culture, Medicine, and Psychiatry*, 44(1):56–79, 2020. Springer.

531. S.J. Russell and P. Norvig. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,, 2016.

532. S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *The Cryptography Mailing List*, 2008.

533. W. Saadeh, M.A.B. Altaf, and M.S.B. Altaf. A high accuracy and low latency patient-specific wearable fall detection system. In *Proc. of the International Conference on Biomedical & Health Informatics (BHI'17)*, pages 441–444, Orlando, FL, USA, 2017. IEEE.

534. A.M. Sabatini, G. Ligorio, A. Mannini, V. Genovese, and L. Pinna. Prior-to and post-impact fall detection using inertial and barometric altimeter measurements. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 24(7):774–783, 2015. IEEE.

535. O. Said and M. Masud. Towards Internet of Things: Survey and future vision. *International Journal of Computer Networks*, 5(1):1–17, 2013. Computer Science Journals.

536. N.Y. Saiyad, H.B. Prajapati, and V.K. Dabhi. A survey of document clustering using semantic approach. In *Proc. of the International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT'16)*, pages 2555–2562, Chennai, India, 2016. IEEE.

537. A. Salinca. Business reviews classification using sentiment analysis. In *Proc. of the International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC'15)*, pages 247–250, Timisoara, Romania, 2015. IEEE.

538. S. Salvador, P. Chan, and J. Brodie. Learning states and rules for time series anomaly detection. In *Proc. of the Seventeenth International Florida Artificial Intelligence Research Society Conference, Miami Beach, Florida, USA*, pages 306–311, 2004. AAAI Press.

539. M. Samaniego and R. Deters. Blockchain as a Service for IoT. In *Proc. of the International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 433–436, Chengdu, China, 2016. IEEE.

540. M. Samaniego and R. Deters. Using blockchain to push software-defined IoT components onto edge hosts. In *Proc. of the International Conference on Big Data and Advanced Wireless Technologies (BDAW'16)*, page 58, Blagoevgrad, Bulgaria, 2016. ACM.

541. K.S. Sandeep. Mobile fog based secure cloud-iot framework for enterprise multimedia security. *Multimedia Tools and Applications*, 79(15-16):10717–10732, 2020. Springer.

542. N.S. Sattar and S.M. Arifuzzaman. Community Detection using Semi-supervised Learning with Graph Convolutional Network on GPUs. In *Proc. of the International Conference on Big Data (Big Data 2020)*, pages 5237–5246, Los Alamitos, CA, USA, 2020. IEEE Computer Society.

543. M. Sattari and K. Zamanifar. A spreading activation-based label propagation algorithm for overlapping community detection in dynamic social networks. *Data & Knowledge Engineering*, 113:155–170, 2018. Elsevier.

544. S. Saurabh, S. Madria, A. Mondal, A.S. Sairam, and S. Mishra. An analytical model for information gathering and propagation in social networks using random graphs. *Data & Knowledge Engineering*, 129:101852, 2020. Elsevier.

545. D. Savage, X. Zhang, X. Yu, P. Chou, and Q. Wang. Anomaly detection in online social networks. *Social Networks*, 39:62–70, 2014. Elsevier.

546. A. Saxena, R. Gera, I. Bermudez, D. Cleven, E.T. Kiser, and T. Newlin. Twitter Response to Munich July 2016 Attack: Network Analysis of Influence. *Frontiers in Big Data*, 2:17, 2019. Frontiers.

547. P. Schäfer and U. Leser. Multivariate time series classification with WEASEL+ MUSE. *arXiv preprint arXiv:1711.11343*, 2017.

548. C. G. Schmidt and S. M. Wagner. Blockchain and supply chain relations: A transaction cost theory perspective. *Journal of Purchasing and Supply Management*, 25(4):100552, 2019.

549. N. Schrading, C.O. Alm, R. Ptucha, and C. Homan. An analysis of domestic abuse discourse on 2389. In *Proc. of the International Conference on Empirical Methods in Natural Language Processing (EMNLP'15)*, pages 2577–2583, Lisbon, Portugal, 2015. Association for Computational Linguistics.

550. D. Schuff and S. Mudambi. What makes a helpful online review? A study of customer reviews on Amazon.com. *Social Science Electronic Publishing*, 34(1):185–200, 2012. Elsevier.

551. J. Sedding and D. Kazakov. WordNet-based text document clustering. In *Proc. of the International Workshop on RObust Methods in Analysis of Natural Language Data (ROMAND 2004)*, pages 104–113, Geneva, Switzerland, 2004.

552. H.J. Seo and P. Milanfar. Static and space-time visual saliency detection by self-resemblance. *Journal of vision*, 9(12):15–15, 2009. The Association for Research in Vision and Ophthalmology.

553. J.C. Medina Serrano, O. Papakyriakopoulos, and S. Hegelich. Dancing to the partisan beat: a first analysis of political communication on TikTok. In *Proc. of the International Web Science Conference (WebSci'20)*, pages 257–266, Southampton, England, UK, 2020.

554. C. Shahabi and D. Yan. Real-time Pattern Isolation and Recognition Over Immersive Sensor Data Streams. In *Proc. of the International Conference on Multimedia Modeling (MMM'03)*, pages 93–113, Taipei, Taiwan, 2003.

555. A.R. Shahid, N. Pissinou, C. Staier, and R. Kwan. Sensor-Chain: A Lightweight Scalable Blockchain Framework for Internet of Things. In *Proc. of the International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 1154–1161, Atlanta, GE, USA, 2019. IEEE.

556. M. Shao, J. Li, F. Chen, H. Huang, S. Zhang, and X. Chen. An efficient approach to event detection and forecasting in dynamic multivariate social media networks. In *Proc. of the 26th International Conference on World Wide Web*, pages 1631–1639, Perth, Australia, 2017. ACM.

557. M. Sharma, K. Yadav, N. Yadav, and K.C. Ferdinand. Zika virus pandemic-analysis of Facebook as a social media health information platform. *American Journal of Infection Control*, 45(3):301–302, 2017. Elsevier.

558. V. Sharma, I. You, and R. Kumar. Isma: Intelligent sensing model for anomalies detection in cross platform osns with a case study on iot. *IEEE Access*, 5:3284–3301, 2017. IEEE.

559. A. Sheikhahmadi and M.A. Nematbakhsh. Identification of multi-spreader users in social networks for viral marketing. *Journal of Information Science*, 43(3):412–423, 2017. SAGE Publications Sage UK: London, England.

560. S. Shekhar, C. Lu, and P. Zhang. Detecting graph-based spatial outliers: algorithms and applications (a summary of results). In *Proc. of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, San Francisco, CA, USA, August 26-29, 2001*, pages 371–376, 2001. ACM.

561. A. K. Shelton and P. Skalski. Blinded by the light: Illuminating the dark side of social network use through content analysis. *Computers in Human Behavior*, 33:339–348, 2014. Elsevier.

562. C. Shen, X. Huang, and Q. Zhao. Predicting eye fixations on webpage with an ensemble of early features and high-level representations from deep network. *IEEE Transactions on Multimedia*, 17(11):2084–2093, 2015. IEEE.

563. C. Shen and Q. Zhao. Webpage saliency. In *Proc. of the European Conference on Computer Vision (ECCV'14)*, pages 33–46, Zurich, Switzerland, 2014. Springer.

564. J. Shen, J. Zhou, Y. Xie, S. Yu, and Q. Xuan. Identity Inference on Blockchain using Graph Neural Network. In *Proc. of the International Conference on Blockchain and Trustworthy Systems (BlockSys21)*, pages 3–17, Virtual Location, 2021. Springer.

565. M. Shen, X. Tang, L. Zhu, X. Du, and M. Guizani. Privacy-Preserving Support Vector Machine Training over Blockchain-Based Encrypted IoT Data in Smart Cities. *IEEE Internet of Things Journal*, 2019. IEEE.

566. W. Shen, Y.J. Hu, and J.R. Ulmer. Competing for Attention: An Empirical Study of Online Reviewers' Strategic Behavior. *MIS Q.*, 39(3):683–696, 2015. Management Information Systems Research Center.

567. X. Shi, B.L. Tseng, and L.A. Adamic. Looking at the blogosphere topology through different lenses. In *Proc. of the International Conference on Weblogs and Social Media (ICWSM'07)*, Boulder, CO, USA, 2007.

568. N. Shrivastava, A. Majumder, and R. Rastogi. Mining (social) network graphs to detect random link attacks. In *Proc. of the 24th International Conference on Data Engineering, ICDE 2008, April 7-12, 2008, Cancún, Mexico*, pages 486–495, 2008. IEEE Computer Society.

569. S. Sicari, A. Rizzardi, L.A. Grieco, and A. Coen-Porisini. Security, privacy and trust in Internet of Things: The road ahead. *Computer Networks*, 76:146–164, 2015. Elsevier.

570. W.F. Silvano and R. Marcelino. Iota Tangle: A cryptocurrency to communicate Internet-of-Things data. *Future Generation Computer Systems*, 112:307–319, 2020. Elsevier.

571. B. Silveira, H.S. Silva, F. Murai, and A.C.C. da Silva. Predicting user emotional tone in mental disorder online communities. *Future Generation Computer Systems*, 125:641–651, 2021. Elsevier.

572. D. Simon. *Evolutionary optimization algorithms*. John Wiley & Sons, 2013.

573. D. Simon, S. Sridharan, S. Sah, R. Ptucha, C. Kanan, and R. Bailey. Automatic scanpath generation with deep recurrent neural networks. In *Proc. of the Symposium on Applied Perception (SAP'16)*, pages 130–130, Anaheim, USA, 2016.

574. K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *Proc. of the International Conference on Learning Representations (ICLR'14)*, 2013. ICLR Press.

575. E. Simpson and B. Semaan. For You, or For"You"? Everyday LGBTQ+ Encounters with TikTok. *Proc. of the International Conference on Human-Computer Interaction (HCI'21)*, 4(CSCW3):1–34, 2021. ACM.

576. P. Singer, F. Flöck, C. Meinhart, E. Zeitfogel, and M. Strohmaier. Evolution of Reddit: From the Front Page of the Internet to a Self-Referential Community? In *Proc. of the International Conference on World Wide Web (WWW 2014)*, page 517–522, Seoul, Korea, 2014. ACM.

577. A.P. Singh and J. Dangmei. Understanding the generation Z: the future workforce. *South-Asian Journal of Multidisciplinary Studies*, 3(3):1–5, 2016.

578. L. Singh, S. Bansal, L. Bode, C. Budak, G. Chi, K. Kawintiranon, C. Padden, R. Vanarsdall, E. Vraga, and Y. Wang. A first look at COVID-19 information and misinformation sharing on Twitter. *arXiv preprint arXiv:2003.13907*, 2020.

579. M. Singh, D. Bansal, and S. Sofat. Behavioral analysis and classification of spammers distributing pornographic content in social media. *Social Network Analysis and Mining*, 6(1):41:1–41:18, 2016. Springer.

580. R. Singh, J. Woo, N. Khan, J. Kim, H.J. Lee, H.A. Rahman, J. Park, J. Suh, M. Eom, and N. Gudigantala. Applications of machine learning models on yelp data. *Asia Pacific Journal of Information Systems*, 29(1):117–143, 2019.

581. V.K. Singh, H.A. Rashwan, S. Romani, F. Akram, N. Pandey, M.M.K. Sarker, A. Saleh, M. Arenas, M. Arquez, and D. Puig. Breast tumor segmentation and shape classification in mammograms using generative adversarial and convolutional neural network. *Expert Systems with Applications*, 139:112855, 2020. Elsevier.

582. T. Sodani and S. Mendenhall. Binge-Swiping Through Politics: TikTok's Emerging Role in American Government. *Journal of Student Research*, 10(2), 2021.

583. A. Soliman, J. Hafer, and F. Lemmerich. A Characterization of Political Communities on Reddit. In *Proc. of the ACM Conference on Hypertext and Social Media (HT'19)*, page 259–263, Hof, Germany, 2019. ACM.

584. V.N. Soloviev and A. Belinskiy. Complex systems theory and crashes of cryptocurrency market. In *Proc. of the International Conference on Information and Communication Technologies in Education, Research, and Industrial Applications (ICTERI'18)*, pages 276–297, Kyiv, Ukraine, 2018. Springer.

585. S. Somin, G. Gordon, and Y. Altshuler. Network analysis of ERC20 tokens trading on ethereum blockchain. In *Proc. of the International Conference on Complex Systems (ICCS'18)*, pages 439–450, Cambridge, MA, USA, 2018. Springer.

586. S. Souravlas, S. Anastasiadou, and S. Katsavounis. A survey on the recent advances of deep community detection. *Applied Sciences*, 11(16):7179, 2021. Multidisciplinary Digital Publishing Institute.

587. A. Srinivasan, J. Teitelbaum, and J. Wu. DRBTS: distributed reputation-based beacon trust system. In *Proc. of the IEEE International Symposium on Dependable, Autonomic and Secure Computing (DASC'06)*, pages 277–283, Indianapolis, IN, USA, 2006. IEEE.

588. J.D. Still. Web page attentional priority model. *Cognition, Technology & Work*, 19(2-3):363–374, 2017. Springer.

589. J.D. Still and C.M. Masciocchi. A saliency model predicts fixations in web interfaces. In *Proc. of the International Workshop on Model Driven Development of Advanced User Interfaces (MDDAUI'10)*, page 25, Atlanta, GA, USA, 2010.

590. C. Stokel-Walker. TikTok's global surge. *New Scientist*, 245(3273):31, 2020. Elsevier.

591. Y.A. Strekalova. Health risk information engagement and amplification on social media: News about an emerging pandemic on Facebook. *Health Education & Behavior*, 44(2):332–339, 2017. SAGE Publication.

592. Y. Su, B.J. Baker, J.P. Doyle, and M. Yan. Fan engagement in 15 seconds: Athletes' relationship marketing during a pandemic via TikTok. *International Journal of Sport Communication*, 13(3):436–446, 2020.

593. R.P. Subbanarasimha, S. Srinivasa, and S. Mandyam. Invisible Stories That Drive Online Social Cognition. *IEEE Transactions on Computational Social Systems*, pages 1–14, 2020. IEEE.

594. A. Sucerquia, J.D. López, and J.F. Vargas-Bonilla. SisFall: A fall and movement dataset. *Sensors*, 17(1):198, 2017. MDPI.

595. S. Sudrich, J. De Melo Borges, and M. Beigl. Anomaly detection in evolving heterogeneous graphs. In *Proc. of the International Conference on Internet of Things (iThings) and*

IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), pages 1147–1149, Exeter, UK, 2017. IEEE Computer Society.

596. H. Sun, N. Ruan, and H. Liu. Ethereum Analysis via Node Clustering. In *Proc. of the International Conference on Network and System Security (NSS'19)*, pages 114–129, Sapporo, Japan, 2019. Springer.
597. Y. Sun and J.D.G. Paule. Spatial analysis of users-generated ratings of yelp venues. *Open Geospatial Data, Software and Standards*, 2(1):5, 2017.
598. M. Suran and D.K. Kilgo. Freedom from the press? How anonymous gatekeepers on Reddit covered the Boston Marathon bombing. *Journalism Studies*, 18(8):1035–1051, 2017. Taylor & Francis.
599. S. Sussman, R. Garcia, T. B. Cruz, L. Baezconde-Garbanati, M. A. Pentz, and J. B Unger. Consumers' perceptions of vape shops in southern california: an analysis of online yelp reviews. *Tobacco induced diseases*, 12(1):22, 2014.
600. M. Szell, R. Lambiotte, and S. Thurner. Multirelational organization of large-scale social networks in an online world. *Proceedings of the National Academy of Sciences*, 107(31):13636–13641, 2010.
601. A.M. Tabar, A. Keshavarz, and H. Aghajan. Smart home care network using sensor fusion and distributed vision-based reasoning. In *Proc. of the International Workshop on Video Surveillance & Sensor Networks (VSSN'06)*, pages 145–154, Santa Barbara, CA, USA, 2006.
602. T. Tamura, T. Yoshimura, M. Sekine, M. Uchida, and O. Tanaka. A wearable airbag to prevent fall injuries. *IEEE Transactions on Information Technology in Biomedicine*, 13(6):910–914, 2009. IEEE.
603. C. Tan and L. Lee. All Who Wander: On the Prevalence and Characteristics of Multi-Community Engagement. In *Proc. of the International Conference on World Wide Web (WWW 2015)*, page 1056–1066, Florence, Italy, 2015. ACM.
604. K. Tan and K.M. Wegmann. Social–Emotional Learning and Contemporary Challenges for Schools: What Are Our Students Learning from Us? *Children & Schools*, 44(1):3–5, 2021. Oxford Academic.
605. G. Tang, Y. Xia, E. Cambria, P. Jin, and T.F. Zheng. Document representation with statistical word senses in cross-lingual document clustering. *International Journal of Pattern Recognition and Artificial Intelligence*, 29(02):1559003, 2015. World Scientific.
606. H. Tang, Y. Jiao, B. Huang, C. Lin, S. Goyal, and B. Wang. Learning to classify blockchain peers according to their behavior sequences. *IEEE Access*, 6:71208–71215, 2018. IEEE.
607. M. Thelwall. Can social news websites pay for content and curation? The SteemIt cryptocurrency model. *Journal of Information Science*, 44(6):736–751, 2018. SAGE Publications.
608. C. Thirumalai, S. Mohan, and G. Srivastava. An efficient public key secure scheme for cloud and IoT security. *Computer Communications*, 150:634–643, 2020.
609. K. Tiidenberg. Boundaries and conflict in a NSFW community on tumblr: The meanings and uses of selfies. *New Media & Society*, 18(8):1563–1578, 2016. Sage Publications.

610. P.L. Ting, S.L. Chen, H. Chen, and W.C. Fang. Using big data and text analytics to understand how customer experiences posted on yelp. com impact the hospitality industry. *Contemporary Management Research*, 13(2), 2017. Academy of Taiwan Information Systems Research.

611. A.J.P. Tixier, M.R. Hallowell, B. Rajagopalan, and D. Bowman. Application of machine learning to construction injury prediction. *Automation in construction*, 69:102–114, 2016. Elsevier.

612. K. Toyoda, T. Ohtsuki, and P.T. Mathiopoulos. Multi-class bitcoin-enabled service identification based on transaction history summarization. In *Proc. of the IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 1153–1160, Halifax, NS, Canada, 2018. IEEE.

613. M. Tsvetovat and A. Kouznetsov. *Social Network Analysis for Startups: Finding connections on the social web*. Sebastopol, CA, USA, 2011. O'Reilly Media, Inc.

614. T. Tucker. Online word of mouth: characteristics of Yelp.com reviews. *Elon Journal of Undergraduate Research in Communications*, 2(1):37–42, 2011.

615. M.M. Tulu, M.E. Mkiramweni, R. Hou, S. Feisso, and T. Younas. Influential nodes selection to enhance data dissemination in mobile social networks: A survey. *Journal of Network and Computer Applications*, page 102768, 2020. Elsevier.

616. A.S. Uban, B. Chulvi, and P. Rosso. An emotion and cognitive based analysis of mental health disorders from social media data. *Future Generation Computer Systems*, 124:480–494, 2021. Elsevier.

617. D. Ursino and L. Virgili. An approach to evaluate trust and reputation of things in a Multi-IoTs scenario. *Computing*, 102:2257–2298, 2020. Springer.

618. G.M. Van Koningsbruggen, T. Hartmann, A. Eden, and H. Veling. Spontaneous hedonic reactions to social media cues. *Cyberpsychology, Behavior, and Social Networking*, 20(5):334–340, 2017. Mary Ann Liebert, Inc. USA.

619. J.M. Vanerio and P Casas. Ensemble-learning approaches for network security and anomaly detection. In *Proc. of the Workshop on Big Data Analytics and Machine Learning for Data Communication Networks, Big-DAMA@SIGCOMM 2017*, pages 1–6, Los Angeles, CA, USA, 2017. ACM.

620. M. Vasek and T. Moore. Analyzing the Bitcoin Ponzi scheme ecosystem. In *Proc. of the International Conference on Financial Cryptography and Data Security (FC'18)*, pages 101–112, Nieuwport, Curaçao, 2018. International Financial Cryptography Association.

621. A. M. Vegni, V. Loscri, and A. Benslimane. SOLVER: A Framework for the Integration of Online Social Networks with Vehicular Social Networks. *IEEE Network*, 34(1):204–213, 2020. IEEE.

622. M. De Veirman, S. De Jans, E. Van den Abeele, and L. Hudders. Unravelling the power of social media influencers: a qualitative study on teenage influencers as commercial content creators on social media. In *The regulation of social media influencers*. 2020. Edward Elgar Publishing.

623. A. Verma and D. Sen. HMM-based Convolutional LSTM for Visual Scanpath Prediction. In *Proc. of the European Signal Processing Conference (EUSIPCO'19)*, pages 1–5, La Coruna, Spain, 2019. IEEE.

624. F. Victor. Address clustering heuristics for Ethereum. In *Proc. of the International Conference on Financial Cryptography and Data Security (FC'20)*, pages 617–633, Kota Kinabalu, Malaysia, 2020. Springer.

625. P. Vikatos, P. Gryllos, and C. Makris. Marketing campaign targeting using bridge extraction in multiplex social network. *Artificial Intelligence Review*, 53(1):703–724, 2020. Springer.

626. C. Villavicencio, S. Schiaffino, J.A. Diaz-Pace, and A. Monteserin. Group recommender systems: A multi-agent solution. *Knowledge-Based Systems*, 164:436–458, 2019. Elsevier.

627. B. Viswanath, A. Mislove, M. Cha, and K.P. Gummadi. On the evolution of user interaction in Facebook. In *Proc. of the ACM Workshop on Online Social Networks (WOSN'09)*, pages 37–42, Barcelona, Spain, 2009. ACM.

628. D. Wang, D. Pedreschi, C. Song, F. Giannotti, and A.L. Barabási. Human mobility, social ties, and link prediction. In *Proc. of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'11)*, pages 1100–1108, San Diego, California, USA, 2011. ACM.

629. F. Wang, Z. Wang, Z. Li, and J.R. Wen. Concept-based Short Text Classification and Ranking. In *Proc. of the International Conference on Information and Knowledge Management (CIKM'14)*, pages 1069–1078, Shangai, China, 2014. ACM.

630. M. Wang, H. Ichijo, and B. Xiao. Cryptocurrency Address Clustering and Labeling. *arXiv preprint arXiv:2003.13399*, 2020.

631. N. Wang, H. Wang, Y. Jia, and Y. Yin. Explainable recommendation via multi-task learning in opinionated text data. In *Proc. of the International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR'18)*, pages 165–174, Ann Arbor, MI, USA, 2018. ACM.

632. P. Wang and Y. Wen. Speculative bubbles and financial crises. *American Economic Journal: Macroeconomics*, 4(3):184–221, 2012.

633. T. Wang, G. Zhang, A. Liu, M. Z. A. Bhuiyan, and Q. Jin. A secure iot service architecture with an efficient balance dynamics based on cloud and edge computing. *IEEE Internet of Things Journal*, 6(3):4831–4843, 2019. IEEE.

634. W. Wang, C. Chen, Y. Wang, T. Jiang, F. Fang, and Y. Yao. Simulating human saccadic scanpaths on natural images. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11)*, pages 441–448, Colorado Springs, CO, USA, 2011. IEEE.

635. B. Wei and L. Chenxi. Study on the Win-Win Strategy of Douyin and Its Users. In *Proc. of the International Conference on Information Systems and Computer Aided Education (ICISCAE'20)*, pages 183–186, Dalian, China, 2020. IEEE.

636. Q. Wei and Z. Jin. Service discovery for internet of things: a context-awareness perspective. In *Proc. of the Fourth Asia-Pacific Symposium on Internetware (Internetware)*, pages 1–6, Qingdao, China, 2012.

637. G. Weimann and N. Masri. Research note: spreading hate on TikTok. *Studies in Conflict & Terrorism*, pages 1–14, 2020. Taylor & Francis.

638. T. Weninger. An exploration of submissions and discussions in social news: mining collective intelligence of Reddit. *Social Network Analysis and Mining*, 4:173–192, 2014. Springer.

639. F. Wilcoxon. Individual Comparisons by Ranking Methods. In *Breakthroughs in statistics*, pages 196–202. 1992. Springer.

640. S. Wold, K. Esbensen, and P. Geladi. Principal Component Analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987. Elsevier.

641. M. Wood, E. Rose, and C. Thompson. Viral justice? Online justice-seeking, intimate partner violence and affective contagion. *Theoretical Criminology*, 23(3):375–393, 2019. SAGE Publications Sage UK: London, England.

642. J. Wu, Q. Yuan, D. Lin, W. You, W. Chen, C. Chen, and Z. Zheng. Who are the phishers? Phishing scam detection on Ethereum via network embedding. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pages 1–11, 2020. IEEE.

643. L. Wu, Q. Zhang, C. Chen, K. Guo, and D. Wang. Deep learning techniques for community detection in social networks. *IEEE Access*, 8:96016–96026, 2020. IEEE.

644. P. Wu, Z. Lu, Q. Zhou, Z. Lei, X. Li, M. Qiu, and P.C.K. Hung. Bigdata logs analysis based on seq2seq networks for cognitive Internet of Things. *Future Generation Computer Systems*, 90:477–488, 2019. Elsevier.

645. S.W. Wu, Z. Wu, S. Chen, G. Li, and S. Zhang. Community detection in blockchain social networks. *Journal of Communications and Information Networks*, 6(1):59–71, 2021. Primera Publisher.

646. Z. Wu, J. Cao, J. Wu, Y. Wang, and C. Liu. Detecting Genuine Communities from Large-Scale Social Networks: A Pattern-Based Method. *The Computer Journal*, 57(9):1343–1357, 2014. Oxford University Press.

647. L. Xu, X. Yan, and Z. Zhang. Research on the causes of the "Tik Tok" app becoming popular and the existing problems. *Journal of Advanced Management Science*, 7(2), 2019.

648. Y. Xu, H. Xu, D. Zhang, and Y. Zhang. Finding overlapping community from social networks based on community forest model. *Knowledge-Based Systems*, 109:238–255, 2016. Elsevier.

649. Q. Xuan, X. Shu, Z. Ruan, J. Wang, C. Fu, and G. Chen. A self-learning information diffusion model for smart social networks. *IEEE Transactions on Network Science and Engineering*, 7(3):1466–1480, 2019.

650. J. Yan, M. Zhu, H. Liu, and Y. Liu. Visual saliency detection via sparsity pursuit. *IEEE Signal Processing Letters*, 17(8):739–742, 2010. IEEE.

651. Z. Yan, P. Zhang, and A.V. Vasilakos. A survey on trust management for Internet of Things. *Journal of Network and Computer Applications*, 42:120–134, 2014. Elsevier.

652. J. Yang, J. Zhang, and Y. Zhang. First Law of Motion: Influencer Video Advertising on TikTok. *Available at SSRN 3815124*, 2021.

653. K. Yang and C. Shahabi. A PCA-based similarity measure for multivariate time series. In *Proc. of the International Workshop on Multimedia Databases (MMDB'04)*, pages 65–74, Washington, DC, USA, 2004. ACM.

654. L. Yang, X. Cao, D. He, C. Wang, X. Wang, and W. Zhang. Modularity based community detection with deep learning. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI'16)*, volume 16, pages 2252–2258, New York City, NY, USA, 2016.

655. W. Yang, Y. Wang, Z. Lai, Y. Wan, and Z. Cheng. Security Vulnerabilities and Countermeasures in the RPL-based Internet of Things. In *Proc. of the International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC'18)*, pages 49–495, Henan, China, 2018. IEEE.

656. Y. Yang, N. Chawla, Y. Sun, and J. Hani. Predicting links in multi-relational and heterogeneous networks. In *Proc. of the International Conference on Data Mining (ICDM'12)*, pages 755–764, Bruxelles, Belgium, 2012. IEEE.

657. H. Yao, H.J. Hamilton, and L. Geng. A unified framework for utility-based measures for mining itemsets. In *Proc. of the ACM SIGKDD Workshop on Utility-Based Data Mining (UBDM'06)*, pages 28–37, Philadelphia, PA, USA, 2006. ACM.

658. O.S. Yaya, A.E. Ogbonna, and O.E. Olubusoye. How persistent and dynamic interdependent are pricing of Bitcoin to other cryptocurrencies before and after 2017/18 crash? *Physica A: Statistical Mechanics and its Applications*, 531:121732, 2019. Elsevier.

659. O.S. Yaya, E.A. Ogbonna, and R. Mudida. Market Efficiency and Volatility Persistence of Cryptocurrency during Pre-and Post-Crash Periods of Bitcoin: Evidence based on Fractional Integration. *International Journal of Finance and Economics*, 2020. John Wiley & Sons.

660. D. Yin, S. Mitra, and H. Zhang. When do consumers value positive vs. negative reviews? An empirical investigation of confirmation bias in online word of mouth. *Information Systems Research*, 27(1):131–144, 2016. INFORMS.

661. J. Yli-Huumo, D. Ko, S. Choi, S. Park, and K. Smolander. Where is current research on blockchain technology? A systematic review. *PloS one*, 11(10):e0163477, 2016. PloS ONE.

662. M. Yoo, S. Lee, and T. Ha. Semantic network analysis for understanding user experiences of bipolar and depressive disorders on reddit. *Information Processing & Management*, 56(4):1565–1575, 2019. Elsevier.

663. W. Yu, J. Li, M. Z. A. Bhuiyan, R. Zhang, and J. Huai. Ring: Real-time emerging anomaly monitoring system over text streams. *IEEE Transactions on Big Data*, 5(4):506–519, 2019. IEEE.

664. Y. Yu, K. Li, W. Zhou, and P. Li. Trust mechanisms in wireless sensor networks: Attack analysis and countermeasures. *Journal of Network and computer Applications*, 35(3):867–880, 2012.

665. Y. Yu, H. Yan, H. Guan, and H. Zhou. Deephttp: Semantics-structure model with attention for anomalous http traffic detection and pattern mining. *CoRR*, abs/1810.12751, 2018. IEEE.

666. Q. Yuan, B. Huang, J. Zhang, J. Wu, H. Zhang, and X. Zhang. Detecting Phishing Scams on Ethereum Based on Transaction Records. In *Proc. of the International Symposium on Circuits and Systems (ISCAS'20)*, pages 1–5, Seville, Spain, 2020. IEEE.

667. B.B. Zarpelão, R.S. Miani, C.T. Kawakani, and S.C. de Alvarenga. A survey of intrusion detection in internet of things. *Journal of Network and Computer Applications*, 84:25–37, 2017. Elsevier.

668. J. Zeng, M.S. Schäfer, and J. Allgaier. Reposting "till albert einstein is TikTok famous": The memetic construction of science on TikTok. *International Journal of Communication*, 15:3216–3247, 2020. University of Southern California.

669. B. Zhang, L. Zhang, C. Mu, Q. Zhao, Q. Song, and X. Hong. A most influential node group discovery method for influence maximization in social networks: a trust-based perspective. *Data & Knowledge Engineering*, 121:71–87, 2019. Elsevier.

670. D. Zhang, J. Yin, X. Zhu, and C. Zhang. User Profile Preserving Social Network Embedding. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI'17)*, pages 3378–3384, Melbourne, Australia, 2017. ijcai.org.

671. J. Zhang and S. Sclaroff. Exploiting surroundedness for saliency detection: a boolean map approach. *IEEE transactions on pattern analysis and machine intelligence*, 38(5):889–902, 2015. IEEE.

672. J.S. Zhang, B.C. Keegan, Q. Lv, and C. Tan. A tale of two communities: Characterizing reddit response to covid-19 through/r/china_flu and/r/coronavirus. *arXiv preprint arXiv:2006.04816*, 2020.

673. K.Z. Zhang, S.J. Zhao, C.M. Cheung, and M.K. Lee. Examining the influence of online reviews on consumers' decision-making: A heuristic–systematic model. *Decision Support Systems*, 67:78–89, 2014. Elsevier.

674. T. Zhang, J. Wang, L. Xu, and P. Liu. Fall detection by wearable sensor and one-class SVM algorithm. *Intelligent computing in signal processing and pattern recognition*, pages 858–863, 2006. Springer.

675. Y. Zhang, D. Raychadhuri, R. Ravindran, and G. Wang. ICN based Architecture for IoT. `https://tools.ietf.org/html/draft-zhang-iot-icn-challenges-02`, 2013. IRTF contribution.

676. Y. Zhang, S. Shi, S. Guo, X. Chen, and Z. Piao. Audience management, online turbulence and lurking in social networking services: A transactional process of stress perspective. *International Journal of Information Management*, 56:102233, 2021. Elsevier.

677. Z. Zhang, Q. Li, D. Zeng, and H. Gao. User community discovery from multi-relational networks. *Decision Support Systems*, 54(2):870–879, 2013. Elsevier.

678. Z. Zhang and K. Wang. A trust model for multimedia social networks. *Social Network Analysis and Mining*, 3(4):969–979, 2013. Springer.

679. J. Zhao, J. Wu, X. Feng, H. Xiong, and K. Xu. Information propagation in online social networks: a tie-strength perspective. *Knowledge and Information Systems*, 32(3):589–608, 2012. Springer.

680. K. Zhao and L. Ge. a survey on the internet of things security. In *Proc. of the International Conference on Computational Intelligence and sSecurity (CISIS'13)*, pages 663–667, Leshan, China, 2013. IEEE.

681. Z. Zhao. Analysis on the "Douyin (Tiktok) Mania" Phenomenon Based on Recommendation Algorithms. In *Proc. of the International Conference on New Energy Technology and Industrial Development (NETID'20)*, volume 235, page 03029, Dali, China, 2021. EDP Sciences.

682. Z. Zhao, C. Li, X. Zhang, F. Chiclana, and E. Herrera Viedma. An incremental method to detect communities in dynamic evolving social networks. *Knowledge-Based Systems*, 163:404–415, 2019. Elsevier.

683. D. Zhelonkin and N. Karpov. Training Effective Model for Real-Time Detection of NSFW Photos and Drawings. In *Proc. of the International Conference on Analysis of Images, Social Networks and Texts (AIST 2019)*, pages 301–312, Kazan, Russia, 2019. Springer.

684. Z. Zheng, S. Xie, H.N. Dai, X. Chen, and H. Wang. Blockchain challenges and opportunities: A survey. *International Journal of Web and Grid Services*, 14(4):352–375, 2018. Inderscience.

685. H. Zhou, Y. Zhang, and J. Li. An overlapping community detection algorithm in complex networks based on information theory. *Data & Knowledge Engineering*, 117:183–194, 2018. Elsevier.

686. M. Zhou, X. Cai, Q. Liu, and W. Fan. Examining continuance use on social network and micro-blogging sites: Different roles of self-image and peer influence. *International Journal of Information Management*, 47:215–232, 2019. Elsevier.

687. C. Zhu, J. Ma, D. Zhang, X.Han, and X. Niu. Hierarchical document classification based on a backtracking algorithm. In *Proc. of the International Conference on Fuzzy Systems and Knowledge Discovery (FSKD'08)*, volume 2, pages 467–471, Jinan, China, 2008. IEEE.

688. C. Zhu, X. Xu, W. Zhang, J. Chen, and R. Evans. How health communication via Tik Tok makes a difference: a content analysis of Tik Tok accounts run by Chinese Provincial Health Committees. *International Journal of Environmental Research and Public Health*, 17(1):192, 2020. Multidisciplinary Digital Publishing Institute.

689. L. Zhu, N.J. Westers, S.E. Horton, J.D. King, A. Diederich, S.M. Stewart, and B.D. Kennard. Frequency of exposure to and engagement in nonsuicidal self-injury among inpatient adolescents. *Archives of suicide research*, 20(4):580–590, 2016. Taylor & Francis.

690. X. Zhuang, J. Huang, G. Potamianos, and M. Hasegawa-Johnson. Acoustic fall detection using Gaussian mixture models and GMM supervectors. In *Proc. of the International Conference on Acoustics, Speech and Signal Processing (ICASSP'09)*, pages 69–72, Taipei, Taiwan, 2009. IEEE.

691. F. Zola, M. Eguimendia, J.L. Bruse, and R.O. Urrutia. Cascading Machine Learning to Attack Bitcoin Anonymity. In *Proc. of the International Conference on Blockchain (ICBC'19)*, pages 10–17, Atlanta, GA, USA, 2019. IEEE.

692. D. Zulli and D.J. Zulli. Extending the Internet meme: Conceptualizing technological mimesis and imitation publics on the TikTok platform. *New Media & Society*, page 1461444820983603, 2020. SAGE.