A Multilayer Network-Based Framework for Handling and Comparing User Histories in X

A Multilayer Network-Based Framework for Handling and Comparing User Histories in X

Abstract

In this paper, we propose a framework that uses a multimodal multilayer network to build, manage and compare User Histories in X. A User History not only considers the contents of interest to the user, as is generally the case with a User Profile. In fact, it also records the set of interactions she has made with contents, the timestamps when they happened, and, ultimately, the history of her actions and behavior. This provides a more comprehensive view of the user, her preferences and needs and paves the way to a number of additional applications. Most of them need to compare two User Histories to calculate their similarity degree. Our framework proposes two approaches, one naive and one refined, to perform this task. We also mention an application that, along with others already proposed in the literature, can greatly benefit from User Histories with the aim of fostering people inclusiveness. Finally, we describe a set of experiments we conducted to evaluate the goodness of the proposed framework.

Keywords: User Histories; Multilayer Networks; Multimodal Networks; User Behavior; Online Social Platforms, X.

1 Introduction

In recent years, the explosive growth of Online Social Platforms (OSPs, for short) has led to an unprecedented accumulation of user-generated data. OSPs have become virtual ecosystems, where individuals interact, share information and engage in various activities. This has prompted researchers to carry out various investigations in this context, involving, for example, the structure and dynamics of the networks underlying OSPs, the influence one user can exert on others, the formation of opinions, the behavior of users and the study of their interactions [33, 32, 14, 13, 20]. The latter is certainly one of the most investigated topics in the context of Social Network Analysis, to analyze user activity patterns, engagement dynamics, behavioral characteristics, etc. [1, 4, 5, 52, 46, 39, 12]. Studying factors such as the behavior of a user in an OSP, the set of her interactions with the contents present therein, the types of interactions (e.g., posting, commenting, liking) with the contents preferred by her paves the way to a large number of applications [46, 11, 28, 31, 13, 17, 15] just as important as those that can be designed by knowing the contents preferred by her.

In this extremely dynamic scenario, a very challenging issue concerns the investigation of User Histories [28, 11, 46]. By this term we refer to a comprehensive record of the activities of a user in an OSP, her interactions, her contributions and the contents of her interest. Thus, a User History differs

from a User Profile in that the latter focuses more on the contents of interest to a user while the former, along with contents, considers the actions made by her on them and, ultimately, her behavior. Extracting meaningful insights from User Histories is a challenging task. Traditional approaches for analyzing OSP users often fail to capture the intricate dynamics and interdependence inherent in the vast amount of data concerning user interactions. They generally put a great emphasis on contents but little emphasis on interactions. Nevertheless, the latter play a key role in reconstructing user behavior beyond the User Profiles. As a final remark, the interactions of a user with the content posted in the OSP happen in specific timestamps. In other words, in reconstructing a User History, we cannot disregard the temporal aspect; rather, this is intrinsic in the concept of history. Taking all these aspects into account in a simple single-mode network is extremely difficult. Thus, we need to adopt more complex data structures, such as multimodal networks [50] and multilayer ones [44, 3].

In this paper, we aim to provide a contribution to address these issues. Specifically, we propose a new framework that uses the concepts of multimodal network and multilayer network to build, handle and compare User Histories in X[1]. This framework, with appropriate abstractions and generalizations, can be extended to other OSPs in the future. Initially, our framework adopts a multimodal single-layer network to represent the users of X, the contents with which they interacted, their interactions, and the timestamps at which these occurred. This representation is extremely compact and useful for storing all the information of interest related to a User History in an OSP. However, at the same time, it is difficult to manage and employ for the next analyses. For this reason, our framework uses a second data model, which consists of a multimodal multilayer network. In it, nodes represent users and contents, while edges denote the interactions that users performed with contents. The time interval of interest is divided into time slices, and the network has a layer for each time slice. An intra-layer edge indicates that a user performed an interaction with a content created in the same time slice. In contrast, an inter-layer edge denotes that a user made an interaction with a content posted in a previous time slice.

Besides defining a model for representing and managing User Histories, our framework defines an approach for comparing them. Regarding this issue, we have to consider that a huge number of user pairs of X should be potentially compared. A very detailed and refined comparison of two User Histories should take into account the interactions made by users, the contents with which these were made and the number of times they were made. However, this type of analysis is very expensive. Therefore, in order to preserve the efficiency of our framework, it makes sense to proceed in two steps. During the first step, which we call naive comparison, we consider only the interactions made by users and filter out all those pairs of users whose sets of interactions carried out by them have medium to low similarity. This task will keep only those pairs of users whose naive similarity is medium to high. On these we apply an approach for refined comparison of User Histories in order to determine their similarity degree much more accurately. Obviously, it is expected that most of the time there will be a concordance (albeit with some margin of difference) between the naive and the refined similarity degrees of two User Histories. However, it cannot be ruled out that there are some situations where this will not be the case. For this reason, we conducted a series of experiments designed to evaluate these and other aspects related to our framework.

[1]We recall that X is the new name of Twitter as of July 2023.

The knowledge of User Histories related to X and, more in general, to an OSP, as well as the ability to determine their similarity degree have many applications. They include all that can be thought for User Profiles, plus more specific ones taking into account the behaviors of users and the set of interactions performed by them. These include healthcare [2, 19], digital marketing [27, 48] and cyber security [18]. We do not detail such applications here and refer the interested reader to cited papers. Instead, we want to mention a potential application, different from those previously considered in the literature, which concerns an e-learning platform. In this case, users are students and contents are topics they must study. The types of interactions represent how these topics are delivered; for example, the same topic can be delivered through video, audio, text, images, etc. The history of a user helps understand her learning preferences. In some cases, such preferences are even needs, because there might be reasons why a user might learn content only through audio, or video, or texts, or images, and so forth. In this particular case, our framework allows the corresponding OSP to ensure much greater inclusiveness. As a consequence, the OSP would be able to behave as a Personalized Learning system [7] for each student.

The outline of this paper is as follows: In Section 2, we provide an overview of the related literature. In Section 3, we present the proposed framework, along with the data models underlying it and the approaches for handling and comparing User Histories. In Section 4, we illustrate the experiments we carried out to evaluate our framework. Finally, in Section 6 we draw our conclusions and outline some future developments of our research efforts.

2 Related Literature

Profiling users in OSPs is a much-studied topic in Social Network Analysis. As mentioned in the Introduction, much of the papers on this topic focus on the contents of interest to users, rather than on user actions and behavior. In this section, we illustrate only those papers that belong to this second category as they are the ones most relevant to our framework. In [4], the authors explore the impact of social and information networks on the relationships created between individuals and their online friends. In conducting such a study they focus on ego networks. They also analyze the degree of social support in the network by investigating the gap between the degree of a user and the average strength of her ties. They also propose using content matching, instead of keyword matching, to improve the accuracy and precision of social media recommendations. Their results suggest that, despite the variable ways in which people communicate and maintain social connections through OSPs, the organization of their social relationships remains largely unchanged.

Profiling user behavior in OSPs has been also analyzed to achieve certain purposes [34, 31]. For instance, in [35] the authors use such profiling to define a personalized pricing policy. In carrying out such a task, profiling is used to understand users' willingness to pay for certain services and, if so, to offer personalized pricing. The authors also show that obtaining accurate information about users is a challenging issue because of privacy constraints. The approach they propose to address this issue is based on a dynamic Bayesian game between the user and the seller and analyzes the Perfect Bayesian Equilibrium (PBE) of the game. The results show that users' activity levels on OSPs do not always decrease as profiling technology improves.

In [38], the authors define an approach for predicting the behavior of OSP users by addressing the

challenges associated with the presence of unbalanced data. Indeed, unlike traditional datasets, data collected by OSPs are often unbalanced. The approach proposed by the authors is called ODW-ELM (Overall Distribution Weighted ELM) and aims to enhance the traditional ELM (Extreme Learning Machine) algorithm by including information about the distribution of data. By assigning an appropriate weight to each class based on its size, ODW-ELM improves prediction accuracy in both balanced and unbalanced datasets. Experimental results show the effectiveness of ODW-ELM in various (both binary and multiclass) classification tasks.

In [54], the authors propose TCAM (Temporal Context-Aware Mixture), an approach that includes intrinsic interests and temporal context influences to capture the intensions and preferences driving user behavior. The item-weighting scheme underlying TCAM improves its performance by prioritizing items that align with the interests of users and the topics of temporal context. The study also presents an efficient query processing technique for rapid online recommendation. Experimental evaluations carried out on various OSPs highlight the superior modeling capability of TCAM, which outperforms the state-of-the-art methods in terms of precision in user behavior modeling and effectiveness in making recommendations.

Unlike the framework proposed in this paper, all the approaches described so far do not use multilayer networks to study user behavior in OSPs. Actually, multilayer networks significantly increase the potential of modeling, and next managing, information about user behavior and its evolution over time. They make it possible to model and manage the interdependencies and relationships between nodes belonging to different layers of the network. Consequently, depending on the meaning we give to layers, multilayer networks allow us to model and manage relationships between data from different time instants or those between different concepts associated with the same phenomenon. In turn, all this greatly enhances the accuracy of approaches for profiling the behavior of users, taking into account the multidimensional nature of their histories. In addition, by including multiple layers of user data, multilayer network-based approaches can help detect hidden insights as well as better identify the differences between User Histories and the peculiarities characterizing each of them.

In [40], the authors provide a foundation for a multilayer network-based approach to Social Network Analysis, which integrates insights from various disciplines. Thanks to the usage of multilayer networks, this approach enables a comprehensive understanding of complex social interactions. Its emphasis on multilayer analysis makes it possible to examine how individual behavior and broader network dynamics interconnect and influence one another. This approach, particularly through the use of the Multi-Theoretical Multi-Level (MTML) model, facilitates the analysis of social networks enabling a more dynamic and inclusive understanding of their dynamics and structures. The Multi-Theoretical approach and the Multi-Level analysis allow the examination of how interactions at various levels affect the overall evolution of the network, since they reflect their layered and evolving nature. Based on this, several studies have begun to analyze social networks through multilayer models because these models can facilitate the analysis and representation of multiple aspects of the same social network from different perspectives [41, 22, 24, 43]. Specifically, in [41] the authors propose to model a social network through a multilayered structure, reflecting how real-world communities overlap due to interactions in various contexts. The geographic multilayer weighted social network model proposed by these authors preserves the Granovetterian structure of strong community ties connected to weak links. At the same time, it enhances overlaps between communities through indirect inter-layer cor-

relations influenced by geographic constraints. This approach is similar to ours in that it presents a multilayer view of social networks. However, it differs from ours as it does not handle the temporal component and focuses on links between individuals, without considering interactions with content.

In [43], the authors aim to address the complexity of modern social networks, which often consist of various types of connections. To this end, they propose to model such networks as multilayer graphs in which each layer represents distinct aspects of user interactions or interests. Then, they employ advanced latent variable models to analyze these multilayer networks and demonstrate, through both simulations and real data obtained from the ENRON email dataset, that integrating different layers can enhance the analysis and detect deeper insights into user relationships. Finally, the authors' proposed approach also traces how the various interactions modeled in multilayer networks evolve over time. In this way, it is able to provide a dynamic and comprehensive view of network evolution [23, 6].

In [23], the authors investigate the temporal dynamics of social networks in the social spider Stegodyphus dumicola. To this end, they employ multilayer network analysis to understand how these dynamics influence the behavior of individuals and groups. The authors' analysis shows that social interactions change at a constant rate, but these changes do not significantly affect behaviors like prey attack speed. The authors highlight the importance of dynamic social network modeling in understanding the biological implications of temporal changes in social structures and offer insights potentially applicable to a wide range of social systems. Both the approach in [23] and ours emphasize the temporal dimension as a critical factor in multilayer social network analysis. However, while the former focuses on the dynamics of social networks among spiders, the latter delves into online social networks by exploring how their interactions and connections evolve over time.

An example of the application of multilayer networks for building and handling a user behavior is proposed in [42]. Here, the authors present SocialAU (Social media Authoritative User), an approach to identify influential users in social media. SocialAU models contents posted by a user through a multilayer heterogeneous network and identifies influential users by analyzing not only the network topology but also the opinions expressed by users. They evaluate SocialAU on Twitter and Yelp datasets and compare the results obtained with those returned by other existing approaches. Their experiments show that SocialAU exhibits excellent accuracy and robustness and is able to find influential users whose authoritativeness well agrees with the concept of expertise. Unlike our framework, SocialAU does not consider interactions and temporal aspects. In [36], the authors propose an approach to model user behavior on Twitter. They first use Latent Dirichlet Allocation [10] to identify topics in tweets. Then, they build a directed multilayer network that connects users to the conversations they participated in and the topics they discussed in those conversations. Inter-layer connections link users to the conversations in which they participated. The authors employ the multilayer network to identify influential users and groups of highly connected users; then, they analyze group dynamics and user connectivity. They show that using multilayer network yields more robust results, particularly when the analyzed tweets come from a variety of discussions. The approach in [36] differs substantially from our framework for several reasons. In fact, it is specific to Twitter, it does not consider several types of interactions, and, finally, it is more focused on the topics of interest to the user than on the actions taken by her.

In [47], the authors propose a new recommender system based on users' past history. This approach

employs a multilayer social network to identify target user communities and determines the importance of users exploiting page rank scores. It combines behavioral similarity and social similarity metrics to compute the nearest neighbor similarity of the target user. The experiments conducted show that the precision and recall obtained by this approach are very high. Both this approach and our framework use a multilayer network and reconstruct users' past histories. However, they present many differences in both the objectives and the way of proceeding. For example, the approach of [47] is more focused on the items that affected the user in the past (and thus on contents) while our framework focuses more on user actions. In [30], the authors propose an approach for predicting links between nodes in a network. It is based on a multilayer network that consists of two layers associated with Twitter and Foursquare, respectively. It extracts node-based and meta-path-based features and apply classical classifiers on them to perform link prediction. It also calculates Kendall's correlation coefficient and Hamming distance between layers to determine correlations and, ultimately, dependencies between node activities. The results of their experiments show moderate correlations and high values of the Hamming distance, and thus a low dependence, between layers. The approach in [30] and our framework have some similarities in the use of multilayer network and meta-information. However, their purpose and way of proceeding are totally different.

Finally, in [8] the authors propose a methodology to detect anomalous users in multilayer social networks. They observe that existing anomaly detection techniques in OSPs focus on networks with only one type of interactions between users. Instead, in reality, human interactions are inherently multiplex, with the existence of multiple types of relationships that can be effectively modeled through multilayer networks. The approach they propose is called ADOMS (Anomaly Detection On Multilayer Social networks). It is an unsupervised, parameter-free and network feature-based methodology that automatically detects anomalous users in a multilayer social network and ranks them accordingly to their anomalousness. ADOMS considers two well-known anomalous patterns, namely clique/near-clique and star/near-star anomalies. It ranks users based on the similarity degree of their neighborhoods, distributing them in different layers. Both ADOMS and our framework employ a multilayer network. However, the purpose of ADOMS is anomaly detection and is completely different from the goals of our framework. Moreover, ADOMS does not employ interactions in defining a user's behavior.

As a final remark, we point out that the expressive representation power of multilayer networks has been strongly emphasized in the context of Social Network Analysis, especially in inter-disciplinary scenarios. For example, in [45] the authors present a study of systematic financial risk focusing on the Mexican banking system from 2007 to 2013. To this end, they introduce a multilayer network-based approach to investigate the role of interconnections among banks linked through various types of financial contracts, like derivatives. They show that the previous approaches, which did not use multilayer networks to represent this scenario, significantly underestimated the total systemic risk by as much as 90%. The multilayer approach also provides interesting results in the financial context. For example, in [9] the authors present a multilayer network model with contagion dynamics capable of simulating the information diffusion and transactions of a typical financial market. In this case, the network consists of two layers representing trading decisions and information dynamics, respectively. The analysis performed by the authors compares the behavior of a two-asset market with an isolated one-asset version and assesses their degree of correlation.

Multilayer networks are also indicated as a preferred representation for studying social phenomena.

For example, in [37] the authors introduce a two-layer network model for social coordination in order to investigate decision-making processes based on both social and strategic motivations. This model serves as a system for two populations of agents playing a coordination game between layers. It is used to explain collective social behavior resulting from interactions among individuals. Similarly, in [51] the authors develop a multilayer network-based framework to investigate how pooling or splitting behavior at the level of dyadic relationships affects social properties at both the individual and group levels. The ultimate goal of this framework is to reliably capture multifaceted aspects of sociality through a multilayer representation.

We conclude this section by observing that the application of multilayer networks to Social Network Analysis has led to a new research strand called multilayer network science [3]. As we have seen above, several papers have been proposed in the literature that fall under this line of research. In light of the considerations we have made in this section, and those we will make in the following of this paper, we can say that our work provides an important contribution in the context of multilayer network science, since it shows how the multilayer network model enables the management and comparison of User Histories in X and, potentially, in other Online Social Platforms.

3 Description of the proposed framework

In this section, we present our framework for building and comparing User Histories in X. Specifically, in Subsection 3.1 we illustrate the data models underlying our framework, while in Subsection 3.2 we describe the approaches it uses to achieve its goals.

3.1 Data models used in our framework

In this section, we illustrate the data models used in our framework. Specifically, in Subsection 3.1.1 we describe a multimodal single-layer network-based model to represent user activities in X, while in Subsection 3.1.2 we see how this Online Social Platform can be represented using a multimodal multilayer network-based model.

3.1.1 A multimodal single-layer network-based model of X

X is a content-based social network for which we are interested in two types of nodes, namely users and contents. Users post contents or interact with them. Contents represent various types of posts on the platform, such as texts, comments, images or videos.

Formally speaking, we denote the multimodal single-layer network as:

$$\mathcal{N} = (N, E) \tag{3.1}$$

N is the set of nodes of $\mathcal{N}$. It can be partitioned into two disjoint subsets: $N = U \cup C$, where U represents the set of nodes associated with users and C denotes the set of nodes associated with contents. A node $u_i \in U$ represents a user who posted or interacted with a content in X. A node $c_j \in C$ denotes a content posted in X by a user. An edge $e_{ij} = (u_i, c_j) \in E$ indicates that u_i posted c_j or interacted with it.

In order to study User Histories, we need a mechanism to represent time within our framework. In particular, we need to examine $\mathcal{N}$ over different time slices to understand how User Histories evolve. Suppose the time interval of interest is equal to $\mathcal{T} = [0, \overline{\tau}]$, where 0 (resp., $\overline{\tau}$) is the first (resp., last) timestamp for which we have data. $\mathcal{T}$ can be divided into n time slices, all of the same length: $ts_1 = [0, \tau_1)$, $ts_2 = [\tau_1, \tau_2)$, $\cdots$, $ts_n = [\tau_{n-1}, \tau_n = \overline{\tau}]$. In the following, we denote by $TS = \{ts_1, ts_2, \cdots, ts_n\}$ the set of all time slices of $\mathcal{T}$.

Finally, we assume that u_i can interact with c_j in various ways. An example of interaction could be a comment to c_j performed by u_i at the time slice ts_h. We denote by R the set of all the possible types of interactions that a user can have with a content in X.

In order to track all the interactions that u_i had with c_j, our model associates the edge $e_{ij} = (u_i, c_j) \in E$ with a list I_{ij} of pairs. The pair (ts_h, r_k) of I_{ij} indicates that u_i performed an interaction of type r_k at the time slice ts_h with c_j. Note that, in principle, at the same time slice ts_h, u_i could perform the same interaction r_k with c_j more times. This means that the list I_{ij} can have duplicates.

An example of a multimodal single-layer network representing the interactions of users with the contents of X is shown in Figure 1.

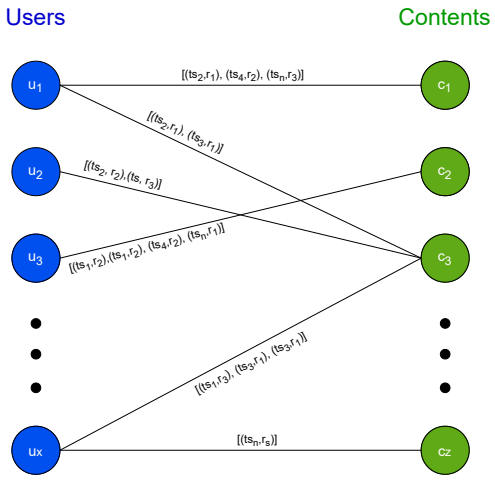

Figure 1: An example of a multimodal single-layer network representing the interactions of users with the contents of X

In principle, this representation allows us to capture the history of a user in X in that it stores all the various interactions she made with the contents present therein. However, it introduces a high level of complexity. In fact, each edge $e_{ij} \in E$ of $\mathcal{N}$ must maintain a list of pairs (ts_h, r_k) that can grow significantly when authors are active or a content is popular (see, for instance, Section 4.1). This not only leads to increased storage requirements but also complicates platform analyses. For example, deriving meaningful structures or metrics handling the temporal evolution of interactions related to a user or a content from this network becomes a challenging task. On the other hand, a User History, whose study is the main focus of this paper, involves an in-depth study of the temporal evolution of user interactions. Moreover, this multimodal network does not provide an intuitive way to visualize temporal evolution; indeed, time slices and interaction types are embedded within edges

and are not directly visible in the network structure. This makes it difficult to get a quick overview of the temporal dynamics of X. For this reason, in the next subsection, we propose a representation of all the information stored in $\mathcal{N}$ by means of a multimodal multilayer network, in which each layer is associated with a different time slice. As we will see, this representation makes for a much easier and more intuitive analysis of the temporal evolution of User Histories and of the interactions users had with contents.

3.1.2 A multimodal multilayer network-based model of X

The goal of our multimodal multilayer network-based model for representing X is to maintain the wealth of information captured by the model introduced in Section 3.1.1 while reducing the complexity of its management and analysis. Our model considers one layer for each time slice. The layer related to the time slice ts_h stores (through its intra-layer edges) all interactions made by users during ts_h with contents created by them or by other users in the same time slice ts_h. The interactions made by the user u_i during ts_h with a content c_j that was posted in a previous time slice ts_v are represented by an inter-layer edge connecting the node u_i in the h-th layer to the node c_j in the v-th layer. In this way, we no longer have to maintain a list I_{ij} of interactions associated with each pair (u_i, c_j). This reduces the complexity of representing interactions over time. In fact, the temporal dynamics are directly visible in the multilayer network structure, where each layer represents a snapshot of the interactions made by users in the corresponding time slice. This makes it easier to track changes in interactions over time, as well as to derive metrics and approaches allowing a User History on X to be built and handled over time.

Our multilayer network-based model can be represented as a tuple:

$$\mathcal{M} = \langle LS, E_x \rangle \tag{3.2}$$

Here, LS is the set of layers, each of which is associated with a time slice. E_x is the set of inter-layer edges. More specifically:

- $LS = \{L_1, L_2, \cdots, L_n\}$ is the set of layers of $\mathcal{M}$. Each layer $L_h \in LS$ corresponds to a time slice $ts_h \in TS$. The layer L_h, $1 \leq h \leq n$, can be modeled as a bipartite network $L_h = \langle U_h \cup C_h, E_h \rangle$, where:
  - $U_h = \{u_{h_1}, u_{h_2}, \cdots, u_{h_p}\}$ is the set of user nodes in L_h. Each node $u_{h_i} \in U_h$ represents a user who posted or interacted with at least one content in the time slice ts_h.
  - $C_h = \{c_{h_1}, c_{h_2}, \cdots, c_{h_q}\}$ is the set of content nodes in L_h. Each node $c_{h_j} \in C_h$ represents a content posted by a user of U_h in the time slice ts_h.
  - $E_h = \{(u_{h_i}, c_{h_j}, r_{h_{ij}}) | u_{h_i} \in U_h, c_{h_j} \in C_h, r_{h_{ij}} \in R\}$ is the set of intra-layer edges in L_h. An edge $e_{h_{ij}} = (u_{h_i}, c_{h_j}, r_{h_{ij}})$ indicates that the user u_{h_i} posted or interacted with the content c_{h_j} in the time slice ts_h; $r_{h_{ij}}$ represents the type of interaction that occurred between u_{h_i} and c_{h_j}. It is worth pointing out that, in the same time slice ts_h, multiple interactions between u_{h_i} and c_{h_j} might occur. In this case, there are multiple edges connecting u_{h_i} and c_{h_j}.

- $E_x = E_x^{uu} \cup E_x^{uc}$ is the set of inter-layer edges in $\mathcal{M}$. Here, E_x^{uu} is the subset of user-user inter-layer edges. It is defined as $E_x^{uu} = \{(u_{h_i}, u_{v_l}) | u_{h_i} \in U_h, u_{v_l} \in U_v, ts_v \neq ts_h \}$. An edge $(u_{h_i}, u_{v_l}) \in E_x^{uu}$ indicates that u_{h_i} and u_{v_l} represent the same person. E_x^{uc} is the subset of user-content inter-layer edges. It is defined as $E_x^{uc} = \{(u_{h_i}, c_{v_j}, r_{h_{ij}}) | u_{h_i} \in U_h, c_{v_j} \in C_v, r_{h_{ij}} \in R, ts_h > ts_v\}$. An edge $(u_{h_i}, c_{v_j}, r_{h_{ij}})$ denotes that u_{h_i} made an interaction of type $r_{h_{ij}}$ with the content c_{v_j} during the time slice ts_h. It is worth pointing out that, in the same time slice ts_h, multiple interactions between u_{h_i} and c_{v_j} might occur. In this case, there are multiple intra-layer edges connecting u_{h_i} and c_{v_j}.

In Figure 2, we report the multilayer network representing the interactions of users with the contents of X already represented by the single-layer network shown in Figure 1.

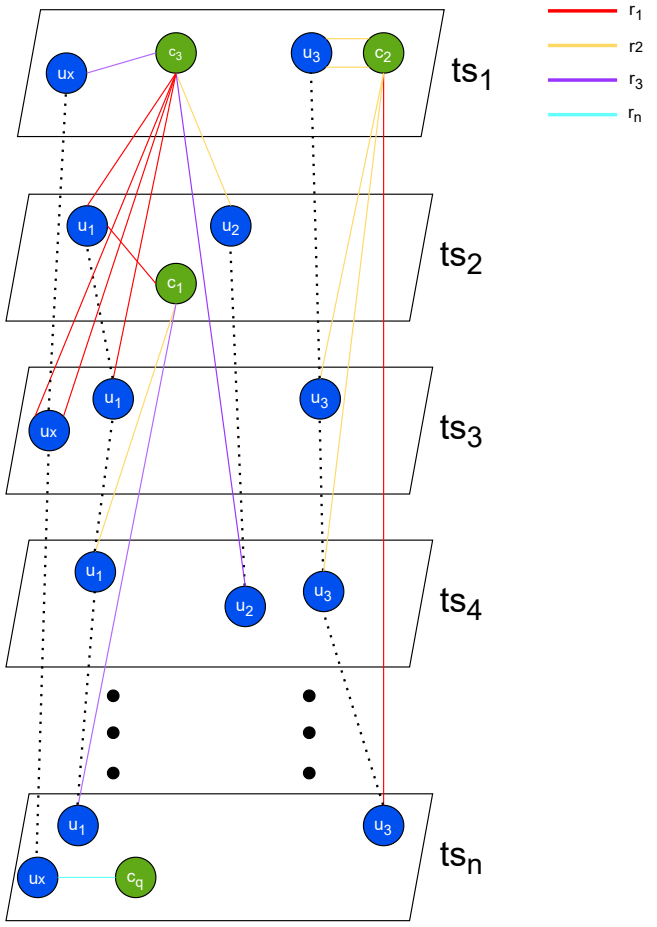

Figure 2: The multilayer network representing the interactions of users with the contents of X already represented by the single-layer network of Figure 1

The multilayer network-based model defined above is very expressive and contains all the information needed to represent and handle User Histories in X. It is certainly easier and more intuitive than the single-layer network-based model in managing time evolution. One might ask whether it is necessary to employ both models in our framework or whether one could adopt only the multilayer

network-based model. We believe that it is more appropriate to use both models. In fact, the single-layer network is more adequate for quickly and compactly storing all the data about the interactions of the X users with contents. For example, adding a new interaction in the single-layer network generally involves adding a new element to the corresponding set I_{ij}, while it involves adding a new inter-layer edge in the multilayer network, and thus changing the network structure. Conversely, the multilayer network-based model is much better suited for the construction and management of User Histories and their time evolution, which require more complex processing in the case of using the single-layer network-based model.

It is worth pointing out that, in this paper, we are defining our two multimodal network-based models with reference to X. This will be even more evident when we will illustrate the experiments we conducted to test our models. In fact, the reference dataset is derived from X. However, we would like to observe that our model is sufficiently generic. As a consequence, with a minimum of further abstraction and generalization, it can be used to represent and manage any content-based Online Social Platform for which we are interested in managing users and content.

3.2 Approaches for User History comparison adopted in our framework

In this section, we describe the User History comparison approaches employed in our framework. Specifically, in Subsection 3.2.1 we introduce a formal definition of the concept of User History. In Subsection 3.2.2, we present the naive approach for comparing two User Histories. Finally, in Subsection 3.2.3, we describe the corresponding refined approach.

3.2.1 Definition of User History

The history of a user in an OSP specifies the evolution of her interactions with the various contents in the platform. In order to reconstruct that history, it is necessary to define a representation based on the sequence of interactions made by her over time. Regarding this, we highlight that each OSP allows certain types of interactions with its contents. For example, X allows posting, replying, retweeting, quoting (i.e., retweeting with comments), and liking. In the following, we use the symbol R to indicate the set of allowed types of interactions.

An interaction can be formalized as a tuple:

$$I = (u, c, r, ts, w) \tag{3.3}$$

where u is the user who performs it, c is the content on which it is performed, r is its type, ts is the time slice in which it occurs, and w is the number of times u performed r on c during ts. Regarding the term "interaction" and the way we use it, a clarification is necessary. In fact, in the timestamp ts, u could perform several times, say w times, the same interaction of type r with c. Strictly speaking, we would have w different interactions. However, this representation would greatly burden our notation and related approaches. For this reason, we prefer to use the term "interaction" and say that it is repeated w times, rather than saying that u performed w interactions of the same type with c in ts.

This formalization captures both intra-layer interactions (in which a user interacts with a content of the same layer) and inter-layer ones (in which a user interacts with a content of a previous layer).

A path is a sequence of interactions:

$$P = \langle I_1, I_2, \cdots, I_z \rangle \tag{3.4}$$

such that the time slice of I_y is less than or equal to the time slice of I_{y+1}, $1 \leq y \leq z-1$.

A user path UP_i is a path P in which all interactions refer to the same user u_i. Note that in defining UP_i with respect to P, we have removed a degree of freedom from the interaction formalization, since the user is fixed.

A User History UH_π is a path in which all interactions refer to the same person π. Recall that in our multilayer network-based model $\mathcal{M}$, users belonging to different layers may actually represent the same person. $\mathcal{M}$ is provided with the subset E_x^{uu} of inter-layer edges to indicate that two users u_{h_i} and u_{v_l} represent the same person π. Exploiting such edges it is possible to construct a User History UH_π by appropriately composing various user paths related to different users that actually refer to the same person π. Strictly speaking, we should call "Person History" the User History UH_π. However, we consider it more intuitive to employ the term "User History" and the symbol UH_i to refer to it in the following. From this point of view, the subscript 'i' that we use instead of the subscript 'π' can be seen as the first user in temporal order corresponding to the person π.

The User History UH_i of a user u_i could be represented as a sequence of interactions being it a special case of path. However, we believe that it is more convenient to represent it as an ordered sequence of sets of interactions. Each set corresponds to a specific time slice and contains all the interactions that u_i performed during it.

Formally speaking, the User History UH_i can be represented as:

$$UH_i = \langle S_{i1}, S_{i2}, \cdots, S_{in} \rangle \tag{3.5}$$

Here, S_{ih}, $1 \leq h \leq n$, is the (possibly empty) set of interactions performed by u_i during the time slice ts_h. It can be represented as:

$$S_{ih} = \{I_{ih_1}, I_{ih_2}, \cdots, I_{ih_q}\} \tag{3.6}$$

where I_{ih_g}, $1 \leq g \leq q$, is the g-th interaction performed by the user u_i during the time slice ts_h.

This representation provides a structured and compact way to store and manage the history UH_i of a user u_i. In fact, it allows an easy addition, removal or modification of the interactions carried out by u_i. It is also the basis for the analyses on User Histories that can be performed through our framework and which we describe starting from the next section.

3.2.2 Naive comparison and alignment of User Histories

Having defined what a User History is and how it can be represented, in this section we illustrate a simple approach (which we call naive) to compare two User Histories and, if they are sufficiently similar, to align them. The approach we consider is naive in that it takes into account only the types of interactions performed by users during the various time slices. In fact, considering all the information available for each interaction (i.e., its type, the content with which it is performed and the number of times it is carried out in the corresponding time slice) would be extremely complex

and would require an enormous computational time. Thanks to the naive approach, we are able to identify, with an acceptable computation time, potentially similar pairs of User Histories. Once a pair of User Histories have been recognized as potentially similar and have been aligned, we can perform a much more refined comparison on them, taking into account all available information. As mentioned above, such a comparison is complex and requires much more computation time, but it is performed on a limited number of potentially interesting and aligned pairs of User Histories.

The first step in the naive approach is the extraction of only the types of the interactions forming the user history UH_i. This can be done by applying to UH_i a projection operator $\Pi'(\cdot)$ that returns a simplified version UH_i^σ of UH_i, which stores sets of interaction types instead of sets of interactions.

Formally speaking:

$$UH_i^\sigma = \Pi'(UH_i) = \langle R_{i1}, R_{i2}, \cdots, R_{in} \rangle \tag{3.7}$$

Here, R_{ih}, $1 \leq h \leq n$, is the (possibly empty) set of interaction types performed by u_i during the time slice ts_h. It can be represented as:

$$R_{ih} = \{R_{ih_1}, R_{ih_2}, \cdots, R_{ih_p}\} \tag{3.8}$$

where R_{ih_g}, $1 \leq g \leq p$, is the type of interaction of S_{ih}. Clearly, $p = |R_{ih}| \leq |S_{ih}| = q$.

We are now able to describe our naive approach for calculating the similarity between two simplified user histories UH_1^σ and UH_2^σ. For this purpose, we use a modified version of the classical algorithm to compute the edit distance [53]. The latter is a measure of dissimilarity between two sequences (usually two strings of characters), defined as the minimum number of operations (insertions, deletions and substitutions) required to transform one sequence into another. Our version is inspired by studies on parameterized edit distance [16] and differs from the classical version in that each item in the sequence is a set of elements, rather than a single one. As a result, we will not check the equality between elements (e.g., the equality of two characters in case of strings) but whether the two sets coincide or one is contained in the other. The condition we check is very loose; this is related to the choice of defining a conservative approach that does not filter out two User Histories when in doubt. In fact, we will later proceed with the refined approach that will allow us to preserve or discard pairs of User Histories considered similar by the naive approach.

Having made this premise, the algorithm we use to calculate the similarity between two simplified User Histories is represented in Algorithm 1. It uses dynamic programming to efficiently calculate the edit distance. It receives two simplified User Histories UH_1^σ and UH_2^σ as input and returns their similarity degree calculated by applying the modified version of the edit distance, as specified above. It first initializes to 0 all the elements of a matrix D of dimensions $(n+1) \times (n+1)$, where n is the length of each simplified User History, which coincides with the number of time slices of $\mathcal{T}$. Then, it fills the first row (resp, column) of D with values ranging from 0 to n. In fact, the first row (resp., column) of D represents the cost of inserting or deleting elements to transform an empty User History into UH_1^σ (resp., UH_2^σ). At this point, the algorithm iterates over each element of D and, then, over each element of the two simplified User Histories. If the sets corresponding to the current elements of the two sequences are not empty and, at the same time, are equal or one of them is contained in the other, then the substitution cost is set to 0. Otherwise, it is set to 1. Once this is done, the

Algorithm 1 Computation of the similarity degree between two simplified User Histories UH_1^σ and UH_2^σ

Require: UH_1^σ, UH_2^σ: two simplified User Histories
D: a $(n+1) \times (n+1)$ matrix
h, k, *cost*: an integer
1: **for** $h \leftarrow 0$ to n **do**
2: **for** $k \leftarrow 0$ to n **do**
3: $D[h,k] \leftarrow 0$
4: **end for**
5: **end for**
6: **for** $k \leftarrow 0$ to n **do**
7: $D[0,k] \leftarrow k$
8: **end for**
9: **for** $h \leftarrow 0$ to n **do**
10: $D[h,0] \leftarrow h$
11: **end for**
12: **for** $h \leftarrow 1$ to n **do**
13: **for** $k \leftarrow 1$ to n **do**
14: $R_{1h} \leftarrow UH_1^\sigma[h]$
15: $R_{2k} \leftarrow UH_2^\sigma[k]$
16: **if** $R_{1h} \neq \emptyset$ **and** $R_{2k} \neq \emptyset$ **and** $((R_{1h} \subseteq R_{2k})$ **or** $(R_{2k} \subseteq R_{1h}))$ **then**
17: $cost \leftarrow 0$
18: **else**
19: $cost \leftarrow 1$
20: **end if**
21: $D[h,k] \leftarrow \min(D[h-1,k]+1, D[h,k-1]+1, D[h-1,k-1]+cost)$
22: **end for**
23: **end for**
24:
25: **return** 1 - $\frac{D[n,n]}{n}$

algorithm fills each element of the matrix with the minimum cost required to transform the prefix of the first sequence into the prefix of the second one, considering the cost of making insertions, deletions and substitutions. The minimum edit distance, and thus the dissimilarity degree, between the two sequences corresponds to the value in the bottom right cell of D (i.e., $D[n,n]$). After computing the minimum edit distance, the algorithm returns the similarity degree between UH_1^σ and UH_2^σ as $1 - \frac{D[n,n]}{n}$. Observe that this similarity degree ranges in the real interval $[0,1]$.

Once we have calculated the similarity degree between UH_1^σ and UH_2^σ, we can assess whether or not this similarity is acceptable for the two histories to be considered similar. The easiest way to do this is to employ a threshold th_s. If the similarity degree is higher than or equal to th_s then UH_1^σ and UH_2^σ can be considered similar; otherwise, they are to be considered dissimilar. Obviously, the

lower th_s, the more conservative we are. The threshold can be set according to the need and can be static or dynamic (e.g., it can depend on the maximum or minimum similarity degree between any two User Histories managed in our framework). We will address the role of th_s in more detail in the experiments.

Once two simplified User Histories UH_1^σ and UH_2^σ have been considered similar based on the computations of Algorithm 1 and th_s, it is necessary to proceed with their alignment. The corresponding algorithm we have defined is obtained by modifying the classical edit distance alignment algorithm in such a way as to take the specificity of our reference context into account. In particular, given two sequences to be aligned (in our case, UH_1^σ and UH_2^σ) we have to consider that, at a certain position of each of them, instead of an element there may be a (possibly empty) set of elements. The algorithm we have defined is reported in Algorithm 2. It starts from the bottom right element of the matrix D, storing the value of the minimum edit distance between UH_1^σ and UH_2^σ, and reconstructs the path that led to the minimum edit distance. To this end, at each step, it checks the values stored in the cells lying at the top, left and top left of the current cell. The direction of the path is determined by the operation (insertion, deletion, or substitution) resulting from the value in the current cell. The algorithm continues to reconstruct the path until it reaches the top left element of D. In its body, it uses the function *prepend* which, given a sequence of sets and a new set, adds the latter at the start of the sequence. The algorithm returns two aligned simplified User Histories AUH_1^σ and AUH_2^σ, along with their length. Here, the term "aligned" denotes that the new sequences have been arranged to make sure that equal sets of interaction types are aligned in the same positions of AUH_1^σ and AUH_2^σ. The aligned sequences represent the sequences of operations transforming UH_1^σ into UH_2^σ. Any element of the sequences can be the set containing the symbol '-'. This special set indicates that, in the corresponding position, an insertion or a deletion operation can be made. It is worth pointing out that, owing to the way AUH_1^σ and AUH_2^σ were derived, they may have a dimension δ greater than n. In particular, we have that $n \leq \delta \leq 2n$.

3.2.3 Refined comparison of User Histories

The approaches for naive comparison and alignment of User Histories described in the previous section allowed us to filter out all pairs of User Histories that were not promising, that is, those that were plausibly dissimilar. At the same time, they allowed us to select the most promising pairs and to perform their alignment. In this section, we define our approach for a refined comparison of two User Histories. It allows us to obtain a value of their similarity degree that takes into account not only the interaction types, as was the case with the naive approach, but also the content and weights involved in each interaction. As mentioned above, and as we will see better below, this approach is computationally expensive. Therefore, it makes sense to apply it only on promising pairs. Furthermore, the alignment between promising pairs obtained by applying Algorithm 2 allows the refined comparison approach to verify only sets of interactions performed by the two users in the same time slice, thus greatly reducing its computational cost.

Having made this premise, let us consider two User Histories UH_1 and UH_2, related to two users u_1 and u_2, and assume that the corresponding simplified User Histories UH_1^σ and UH_2^σ were considered promising by the naive comparison approach. Let AUH_1^σ and AUH_2^σ be the two aligned simplified

Algorithm 2 Alignment of two simplified User Histories UH_1^σ and UH_2^σ

Require: D: a $(n+1) \times (n+1)$ matrix
h, k: an integer
$AUH_1^\sigma, AUH_2^\sigma$: an aligned simplified User History

$h \leftarrow n$
$k \leftarrow n$
$AUH_1^\sigma \leftarrow \emptyset$
$AUH_2^\sigma \leftarrow \emptyset$
while $(h > 0)$ **and** $(k > 0)$ **do**
  $R_{1h} \leftarrow UH_1^\sigma[h]$
  $R_{2k} \leftarrow UH_2^\sigma[k]$
  if $R_{1h} \neq \emptyset$ **and** $R_{2k} \neq \emptyset$ **and** $((R_{1h} \subseteq R_{2k})$ **or** $(R_{2k} \subseteq R_{1h}))$ **then**
    Prepend (R_{1h}, AUH_1^σ)
    Prepend (R_{2k}, AUH_2^σ)
    $h \leftarrow h - 1$
    $k \leftarrow k - 1$
  else if $D[h,k] = D[h-1,k] + 1$ **then**
    Prepend (R_{1h}, AUH_1^σ)
    Prepend $(\{\texttt{-}\}, AUH_2^\sigma)$
    $h \leftarrow h - 1$
  else
    Prepend $(\{\texttt{-}\}, AUH_1^\sigma)$
    Prepend (R_{2k}, AUH_2^σ)
    $k \leftarrow k - 1$
  end if
end while
while $h > 0$ **do**
  $R_{1h} \leftarrow UH_1^\sigma[h]$
  Prepend (R_{1h}, AUH_1^σ)
  Prepend $(\{\texttt{-}\}, AUH_2^\sigma)$
  $h \leftarrow h - 1$
end while
while $k > 0$ **do**
  $R_{2k} \leftarrow UH_2^\sigma[k]$
  Prepend $(\{\texttt{-}\}, AUH_1^\sigma)$
  Prepend (R_{2k}, AUH_2^σ)
  $k \leftarrow k - 1$
end while

return $AUH_1^\sigma, AUH_2^\sigma, |AUH_1^\sigma|$

User Histories corresponding to UH_1^σ and UH_2^σ returned by Algorithm 2. Finally, let δ be the length of AUH_1^σ and AUH_2^σ returned by the same algorithm. Consider that: *(i)* the field u of each interaction related to UH_1 (resp., UH_2) is fixed and equal to u_1 (resp., u_2); *(ii)* each component of AUH_1^σ (resp., AUH_2^σ) consists of either a (possibly empty) set of interaction types or a set consisting of only the character '-'. To facilitate the computation of the refined similarity of UH_1 and UH_2 we can construct two aligned User Histories AUH_1 and AUH_2. The length of AUH_1 and AUH_2 is equal to δ. AUH_1 (resp., AUH_2) is obtained from AUH_1^σ (resp., AUH_2^σ) by substituting: *(i)* each (possibly empty) set R_{1y} (resp., R_{2y}), $1 \leq y \leq \delta$, of AUH_1^σ (resp., AUH_2^σ) with the set S_{1y} (resp., S_{2y}) of interactions from which it was derived (see Section 3.2.2); *(ii)* the set consisting only of the character '-' of AUH_1^σ (resp., AUH_2^σ) with the set consisting only of the null interaction I^ϵ. The latter actually represents the absence of any interaction performed by a user with a content in the corresponding time slice.

At this point, we have that AUH_1 and AUH_2 are two aligned sequences of dimension δ. Then, their similarity degree can be obtained by computing the average of the similarity degrees of each pair V_{1y} and V_{2y}, $1 \leq y \leq \delta$, corresponding to the sets of AUH_1 and AUH_2 located at position y. Note that V_{1y} (resp., V_{2y}) can consist of the empty set, the set composed by I^ϵ alone, or S_{1y} (resp., S_{2y}).

If V_{1y} (resp., V_{2y}) is either the empty set or the set consisting of I^ϵ, the similarity degree γ_{12_y} between V_{1y} and V_{2y} is 0. Thus, the only case in which γ_{12_y} can be different from 0 is when V_{1y} and V_{2y} are non-empty sets S_{1y} and S_{2y} of interactions. In the latter case, to compute the similarity degree, it is necessary to employ an appropriate approach that takes into account the similarity degree of the interactions composing S_{1y} and S_{2y} are, as well as the cardinality of S_{1y} and S_{2y} (which can be the same, very close to each other, or completely different).

One way to solve this problem widely used especially in the information systems literature [21] consists in the adoption of an approach based on the computation of a suitable objective function related to the maximum weight matching on a bipartite graph obtained from the interactions composing S_{1y} and S_{2y}.

In particular, let $BG = \langle NSet, ESet \rangle$ be the bipartite graph associated with S_{1y} and S_{2y}. $NSet = NSet_1 \cup NSet_2$ represents the set of nodes of BG. There is a node $n_{1_a} \in NSet_1$ (resp., $n_{2_b} \in NSet_2$) for each interaction of S_{1y} (resp., S_{2y}). $ESet$ is the set of edges of BG. Each edge $E \in ESet$ can be represented as $(n_{1_a}, n_{2_b}, \omega_{ab})$, where n_{1_a} and n_{2_b} are the nodes it connects and ω_{ab} is its weight.

ω_{ab} must take into account the similarity between the interaction $I_{1_a} = (u_1, c_{1_a}, r_{1_a}, ts_{1_a}, w_{1_a})$ associated with the node n_{1_a}, and the interaction $I_{2_b} = (u_2, c_{2_b}, r_{2_b}, ts_{2_b}, w_{2_b})$, associated with the node n_{2_b}. Now, u_1 and u_2 are the users whose histories are to be studied, and $ts_{1_a} = ts_{2_b} = ts_y$, due to the alignment assumption seen above. Therefore ω_{ab} depends only on the other three factors. A possible formulation of it is the following:

$$\omega_{ab} = \varphi_r \cdot sim_r(r_{1_a}, r_{2_b}) + \varphi_c \cdot sim_c(c_{1_a}, c_{2_b}) + \varphi_w \cdot sim_w(w_{1_a}, w_{2_b}) \tag{3.9}$$

In this case $\varphi_r + \varphi_c + \varphi_w = 1$. The function $sim_r(r_{1_a}, r_{2_b})$ returns 1 if $r_{1_a} = r_{2_b}$, while it returns 0 otherwise. The function $sim_c(c_{1_a}, c_{2_b})$ returns a value within the real interval $[0, 1]$ indicating the similarity of the contents c_{1_a} and c_{2_b}. For this purpose, it could apply BERTopic [25] on c_{1_a} and c_{1_b} to determine the topics related to them. Then, it computes the Jaccard coefficient [29] between the sets of topics related to c_{1_a} and c_{1_b}. The function $sim_w(w_{1_a}, w_{2_b})$ is defined as: $sim_w(w_{1_a}, w_{2_b}) =$

$1 - \frac{|w_{1_a} - w_{2_b}|}{max(w_{1_a}, w_{2_b})}$. Again, the value returned by this function is within the real interval $[0, 1]$. As a consequence, the value of ω_{ab} is within the real interval $[0, 1]$; the greater its value and the greater the similarity between I_{1_a} and I_{2_b}.

The maximum weight matching for BG is a set $ESet' \subseteq ESet$ of edges such that, for each node of $NSet$, there is at most one edge of $ESet'$ incident on it and $|ESet'|$ is maximum.

The objective function that returns the similarity γ_{12_y} between S_{1y} and S_{2y} is defined as:

$$\gamma_{12_y} = \frac{2 \cdot \sum_{(n_{1_a}, n_{2_b}, \omega_{ab}) \in ESet'} \omega_{ab}}{|NSet|} \tag{3.10}$$

The value of γ_{12_y} ranges in the real interval $[0, 1]$; the higher the value and the higher the similarity between S_{1y} and S_{2y}.

Once we have determined the similarity between $AUH_1[y]$ and $AUH_2[y]$, $1 \leq y \leq \delta$, we can compute the overall similarity between AUH_1 and AUH_2, and thus also between UH_1 and UH_2, as:

$$\gamma_{12} = \frac{\sum_{y=1}^{\delta} \gamma_{12_y}}{\delta} \tag{3.11}$$

Clearly, also the value of γ_{12} ranges in the real interval $[0, 1]$; in this case, the higher the value and the higher the similarity between UH_1 and UH_2.

The existence of a similarity value ranging in the real interval $[0, 1]$ for each pair of User Histories makes it possible a lot of elaborations on the latter. For example, a variety of clustering techniques can be applied to them in such a way as to group users into clusters, each of which is characterized by homogeneous histories of its members. In a Personalized Learning application, like the one mentioned in the Introduction, this might make it possible, for example, to cluster a class of students into homogeneous groups, each characterized by similar learning histories and exigencies. In this way, it is possible to design a learning plan customized to the needs of each learner group, thus increasing user inclusiveness in learning platforms.

4 Experiments

In this section, we illustrate the experimental campaign we conducted to evaluate our framework. Specifically, in Subsection 4.1 we describe the dataset we used during the experiments. In Subsection 4.2, we illustrate the characteristics of the two support networks obtained from the dataset. In Subsection 4.3, we analyze the User Histories obtained from the multilayer network. In Subsections 4.4 and 4.5, we evaluate the approaches for the naive and refined comparison of User Histories. Finally, in Subsection 4.6, we illustrate the experiments performed to compare these two approaches.

4.1 Dataset

The dataset used in this experimental campaign is derived from [49]. It concerns user interactions on X. Specifically, it includes tweets and corresponding information collected during a time period ranging from February 6, 2020 to February 12, 2020. The dataset was collected in compliance with Terms and Conditions of X and is publicly accessible. The total number of records is 7,179,307.

Each instance of the dataset represents all the interactions a user made with a tweet. It is characterized by the following fields:

- `tweet_id`: it specifies the tweet identifier.
- `engagee_id`: it denotes the identifier of the user who posted the tweet.
- `engager_id`: it represents the identifier of the engager, i.e., the user who interacted with the tweet.
- `reply_timestamp`: it specifies the timestamp at which the engager replied to the tweet; if she did not perform any reply, the value of this field is null.
- `retweet_timestamp`: it indicates the timestamp at which the engager retweeted the tweet; if she did not retweet at all, the value of this field is null.
- `retweet_with_comment_timestamp`: it denotes the timestamp at which the engager quoted the tweet; if she made no quote, the value of this field is null.
- `like_timestamp`: it represents the timestamp at which the engager liked the tweet; if she did not post any like, the value of this field is null.

It is worth pointing out that an engager may make multiple interactions with the same tweet. For example, she may like a tweet and then retweet it. In this case, the instance corresponding to this pair (engager-tweet) has non-null values for the `retweet_timestamp` and `like_timestamp` fields, while it has null values for the `reply_timestamp` and `retweet_with_comment_timestamp` fields.

The tweets in the dataset cover a wide range of topics reflecting the heterogeneous and dynamic nature of content shared in X. This diversity includes tweets related to current events, personal opinions, social interactions, and various other subjects that can typically be found on X. Indeed, our goal was to have a dataset capable of capturing the heterogeneity of the content present in X, ensuring a comprehensive representation of the X ecosystem during the period to which it refers.

Regarding user interactions, the dataset captures various forms of user engagements, including likes, retweets, replies and quotes. This variety provides a holistic view of how users interact with content on X and can allow the extraction of patterns and preferences in users' behavior during their engagement. These interactions are crucial to understanding the dynamics of information dissemination on this Online Social Platform.

The users in the dataset are anonymized; however, the anonymization process was performed in such a way that a sufficient detail is retained to ensure engagement pattern analysis. In fact, despite anonymization, the dataset retains essential information about users involved, such as the number of followers, the number of users they follow and the activity level. Such information helps to evaluate the influence and reach of users within X.

The random sampling technique adopted to build our dataset ensures that the tweets included in it are representative of the original dataset. This way of proceeding ensures a balance between comprehensiveness and manageability, allowing for an efficient, and yet detailed, analysis of the interactions and patterns within the X community.

Finally, we would like to note that a hash function was applied to all identifiers to ensure user privacy. Thanks to this expedient, the dataset maintains the confidentiality of user information and, at the same time, allows for meaningful analyses. In Table 1, we report some basic statistics of our dataset.

Property	*Value*
Number of tweets	5,227,456
Number of users	7,066,710
Number of replies	379,873
Number of retweets	1,581,922
Number of retweets with comments	108,216
Number of likes	6,183,928

Table 1: Basic statistics of our dataset

4.2 Support Networks

Starting from the dataset described in the previous section, we constructed both the single-layer network and the multilayer one to be employed for our experiments. Table 2 reports some basic statistics of the single-layer network.

Property	*Value*
Number of user nodes	7,066,710
Number of content nodes	5,227,456
Number of edges	13,481,395
Average number of pairs (timestamp, interaction type) in an edge	380
Maximum number of pairs (timestamp, interaction type) in an edge	17,568
Minimum number of pairs (timestamp, interaction type) in an edge	2
Median number of pairs (timestamp, interaction type) in an edge	249

Table 2: Basic statistics of our support single-layer network

Table 2 shows that the number of pairs (timestamp, interaction type) that can be associated with an edge can also be very large. This is the reason why it is very difficult to make elaborations on User Histories and give an intuitive view of them using the single-layer network. In fact, we would have to scroll through the various pairs (timestamp, interaction type) associated with the various edges several times and continuously.

The multilayer network-based model associated with our framework involves the adoption of one layer per time slice (see Section 3.1.2). Since our dataset includes a period of seven days, we decided to set the duration of each time slice to one day. This choice was motivated by the desire to have a fine enough level of granularity, while not creating an overly complex multilayer network, with an excessive number of layers. In our opinion, segmenting the analysis of posts and comments on X into periods shorter than one day does not add any significant value. In fact, such phenomenon is not extremely fast. Moreover, with a finer granularity analysis of one day, we would have added

unnecessary complexity to the multilayer network, which would have had an excessive number of layers without any significant gain in the extracted information. Clearly, in presence of a much larger dataset (e.g., lasting several months or a year), a different time slice (e.g., lasting a week) could have been chosen. In this case, we would have foregone fine-grained analysis of the phenomenon but would not have complicated the multilayer network with the presence of dozens or hundreds of layers, which would have made it extremely time-consuming and complex for our framework to extract information. The intra-layer edges represent the interactions with contents performed by users during the day associated with the corresponding layer. The inter-layer edges are associated with interactions with contents made by users on different days over time. The number of nodes associated with users and contents, as well as the number of intra-layer edges, vary across layers reflecting the dynamic nature of user-content interactions. Table 3 provides some basic statistics of the multilayer network.

Property	*Layer 1*	*Layer 2*	*Layer 3*	*Layer 4*	*Layer5*	*Layer 6*	*Layer 7*	*Interlayer*
Number of edges	1,532,069	1,950,173	2,110,253	1,974,957	2,010,450	1,954,729	1,953,774	1,947,671
Number of nodes	1,353,520	1,718,606	1,847,443	1,735,818	1,762,390	1,713,951	1,715,067	
Number of user nodes	738,737	958,299	1,044,080	972,331	986,033	958,115	961,756	
Number of content nodes	614,783	760,307	803,363	763,487	776,357	755,836	753,311	

Table 3: Basic statistics of our support multilayer network

4.3 Analysis of User Histories

The first investigation on User Histories concerned the analysis of the distribution of the different types of interactions (i.e., replies, retweets, quotes, likes and posts) over time. This distribution is shown in Figure 3. In it, the x-axis represents the date while the y-axis denotes the number of interactions.

From the analysis of this figure we can detect some interesting information. First, we can see that likes represent the most common type of interactions in the reference interval. This highlights that users prefer to express their interest in a content by liking it, rather than replying, retweeting or retweeting with comments. This is likely due to the ease and simplicity of the liking task, compared to the replying or retweeting ones. This figure also shows that the number of posts is relatively stable over the time interval into consideration and suggests a consistent level of content generation by users. In addition, retweets are more common than replies and quotes; this suggests that users prefer to share content without adding their own comments on it. This could be related to the peculiarities of X, whose main goal is to share information quickly and widely. Finally, quote is the least common type of interaction. This could be due to the additional effort required to add a comment to the retweet or it could suggest that users prefer to share content without editing it or making additions to it.

Table 4 presents the distribution of users against the number of interaction types they made in the time interval of interest. From the analysis of this table we can see that a very large proportion of users engage in only one type of interactions. There is also a rather small number of users who engage in two or three types of interactions, while only a negligible number of them engage in more than three types of interactions.

Figure 4 shows the distribution of User Histories against their length in log-log scale. Specifically, the y-axis represents the number of interactions composing a User History while the x-axis represents

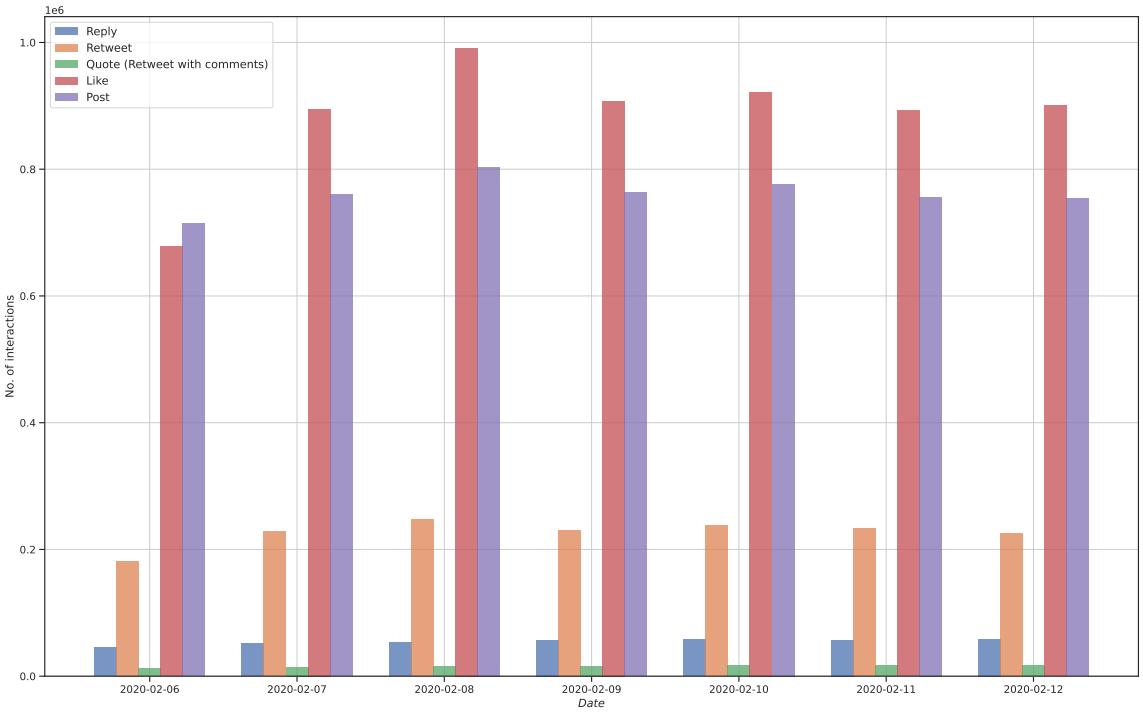

Figure 3: Distribution of the interactions over time

No. of interaction types	*No. of users*
1	4,564,738
2	1,258,448
3	453,347
4	241,810
5	118,649

Table 4: Distribution of users against the number of interaction types they engaged

the number of users. The distribution is strongly skewed toward the bottom, indicating that a large number of users have histories composed by a relatively small number of interactions. Specifically, more than 1.3 million users have recorded only one interaction. Upon initial examination, it appears that there could be issues with our dataset. Actually, this feature of our dataset simply reflects the typical power-law distribution that characterizes the number of interactions of users (as well as many other properties related to them) in an Online Social Platform like X. In other words, it is precisely the presence of this user distribution that ensures robustness and realism to our model. In fact, by having this user distribution, our dataset reflects as accurately as possible the real distribution of users' activities on X, enhancing the applicability and validity of the dataset and the results obtained from applying our framework on it. In addition, the inclusion of the most passive users helps to analyze the impact of passive content consumption, which is a significant aspect of the dynamics of Online Social Platforms. Regarding this, we point out that understanding why a large proportion of

users interact little can lead us to derive insights into the behavior and preferences of this category of users. Such insights can prove useful in several applications, including targeted content delivery and user retention strategies.

Summarizing, while it might seem counter-intuitive to include users with only one interaction in a study focused on user engagement, actually their presence is critical to building a comprehensive and realistic model of the interactions in X. Indeed, these users offer valuable insights into the broader spectrum of user behavior and contribute significantly to the breadth and depth of the analysis.

As the number of interactions grows, the number of users quickly decreases. This trend continues until about 30 interactions, after which the decrease becomes more gradual. This suggests that only a small proportion of users are very active in the platform, presenting a User History with a very high number of interactions. Finally, there are few users with an exceptionally high number of interactions, with the maximum number of interactions present in a single User History being 2,127. Various hypotheses could be made about these users; for example, they could be power users, influencers or bots.

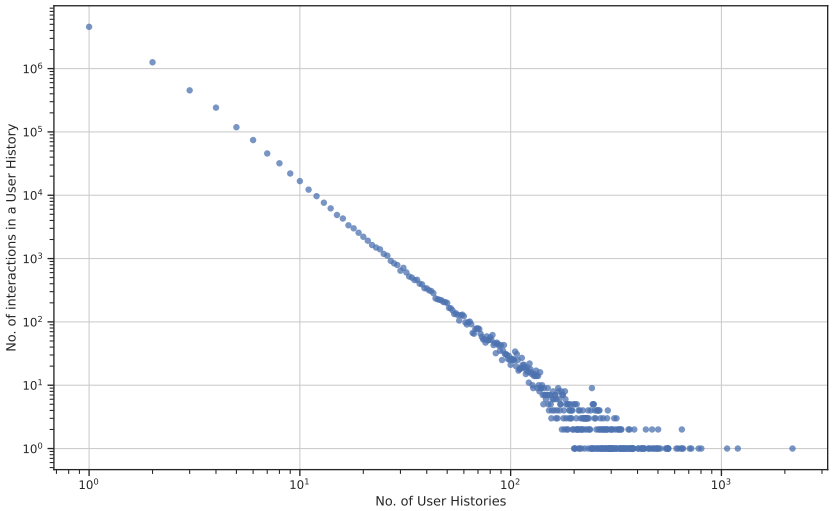

Figure 4: Distribution of User Histories against their length (log-log scale)

4.4 Experiments on the approach for the naive comparison of User Histories

The goal of these experiments is to evaluate our approach for the naive comparison of pairs of User Histories described in Section 3.2.2. This evaluation will allow us to test the filtering level of this approach as the threshold th_s varies. In turn, this will give us important insights into: *(i)* the filtering capability of the naive comparison approach, which should presumably be high, otherwise it would not make much sense to apply this approach before the corresponding refined comparison one; *(ii)* the tuning of the threshold th_s depending on the characteristics of the scenario in which one is operating and the goals one wants to address.

The first observation concerns the number of pairs of User Histories to be considered. In fact, recall that our dataset has 7,066,710 users and as many User Histories. Consequently, the number of pairs of

User Histories that we would potentially have to consider would be enormous, and the computational time to apply the naive comparison approach to all these pairs would be prohibitively long. However, User Histories composed of a very small number of interactions are insignificant. Therefore, we can think of filtering them and try to find a trade-off between the number of interactions of the User Histories and the number of pairs of User Histories to be examined.

In Table 5, we report the number of User Histories and pairs of User Histories against the minimum number of interactions allowed in them. From the analysis of this table we can observe that a good trade-off between the two requirements highlighted above is obtained by selecting all User Histories that contain at least 10 interactions within them. In fact, in this case, the number of pairs to be examined is just over 4.5 billion. For all these pairs, we applied Algorithm 1 to calculate the edit distance and, then, the naive similarity degree.

Minimum No. of interactions	*No. of User Histories*	*No. of pairs of User Histories*
1	6,907,738	23,858,418,684,453
2	2,343,000	2,744,823,328,500
3	1,084,552	588,125,978,076
4	631,205	199,209,560,410
5	389,395	75,814,038,315
6	270,746	36,651,562,885
7	196,414	19,289,131,491
8	150,722	11,358,485,281
9	118,763	7,052,265,703
10	96,714	4,676,750,541

Table 5: Number of User Histories and pairs of User Histories against the minimum number of interactions allowed in them

In Figure 5 and Table 6, we show the number of pairs or User Histories considered similar by the naive approach against the threshold th_s. As can be seen from this figure and this table, our naive comparison approach proves to be very effective in strongly reducing the number of pairs that can be considered similar. In fact, this number is drastically reduced even at values of the threshold th_s not particularly high. This attests to the importance of the naive approach, which allows for an initial filtering of pairs in such a way that the refined approach is applied to only a small fraction of them, considered promising by the naive approach. Without the latter and the filtering it allows, the number of pairs that would have to be examined by the refined approach would be enormous. As a result, taking into account that this approach involves much more complex calculations than the naive one, the computation time for obtaining refined similarities between pairs of User Histories would be prohibitive. This would not be justified taking into account that most of these pairs have very low similarities, as highlighted in Figure 5 and Table 6, and thus would be useless for the next applications.

As for the next experiments, having to choose a threshold value to identify the pairs of User Histories on which to apply the refined approach, we chose to set th_s to 0.50. The first reason for such a choice is that this value allows a sufficiently strong, but not drastic, reduction of the pairs of User Histories. In this way, we are able to apply the refined approach in a reasonable time and,

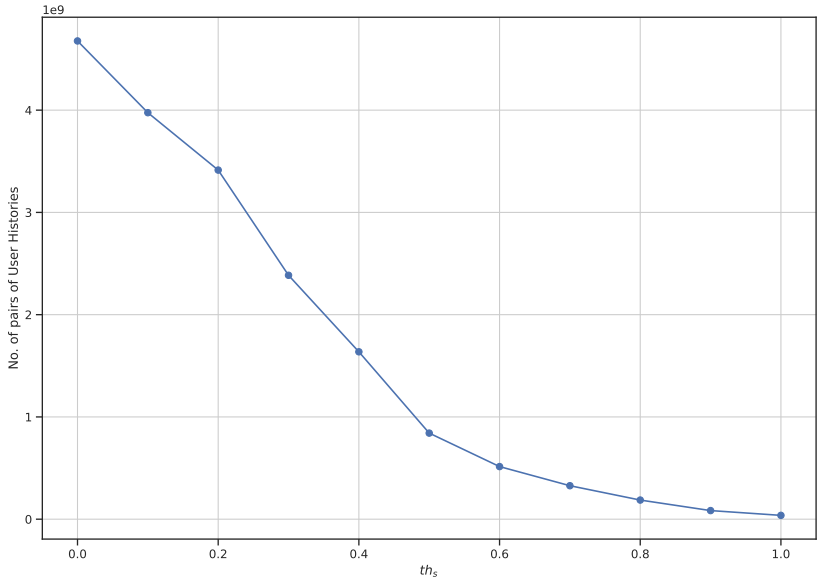

Figure 5: Number of pairs of User Histories considered similar by the naive comparison approach against the values of th_s

Values of th_s	*No. of pairs of User Histories considered similar by the naive comparison approach*
0	4,676,750,541
0.1	3,975,237,959
0.2	3,414,027,877
0.3	2,385,142,741
0.4	1,636,862,689
0.5	**841,815,080**
0.6	514,442,525
0.7	327,372,512
0.8	187,070,013
0.9	84,181,499
1	37,413,985

Table 6: Number of pairs of User Histories considered similar by the naive comparison approach against the values of th_s

simultaneously, avoid discarding potentially similar pairs. The second reason lies in the fact that, at $th_s = 0.50$, the curve in Figure 5 has an elbow; consequently, the choice to set th_s to 0.50 is also in line with the elbow method [26].

4.5 Experiments on the approach for the refined comparison of User Histories

In Section 3.2.3, we have seen that the refined comparison of User Histories takes into consideration three factors, namely the types of interactions made by users, the contents with which users made these interactions, and the number of times users made them (see Equation 3.9). This makes the refined comparison approach much more comprehensive than the naive one, which only takes the types of interactions into account. Underlying the refined approach there is a maximum weight matching on a weighted bipartite graph whose weights are computed by a weighted mean taking into account the three types of similarities mentioned above. Clearly, the refined similarity between two User Histories returned by our approach depends on the three weights φ_r, φ_c and φ_w of Equation 3.9. Since we are interested in the histories of users, their behavior must play a key role. This behavior is determined by the interactions they performed. Therefore, we argue that the greatest value should be assigned to φ_r. However, along with the types of interactions, contents play an extremely important role in determining the similarity between two User Histories; so, we argue that the value to be assigned to φ_c should be sufficiently high. Clearly, in this scenario, the lowest weight will be assumed by φ_w. Based on this reasoning, we set the following weights as the baseline: $\varphi_r = 0.50$, $\varphi_c = 0.35$, $\varphi_w = 0.15$.

Since we have chosen this baseline on the basis of a theoretical reasoning, we consider it important to check whether it is "stable" or, in other words, whether our refined similarity computation approach is resilient against changes in the values of the parameters φ_r, φ_c and φ_w, as long as these changes are reasonable, i.e., they do not upset the hierarchies of weights defined on the basis of the previous reasoning.

To carry out this check, we decided to perform a grid search on the three parameters φ_r, φ_c and φ_w considering many combinations of their values provided that they respect two constraints, namely: *(i)* the sum of the three values must be equal to 1; *(ii)* the hierarchy of the weights must not be skewed, that is, φ_r must be greater than or equal to φ_c, which, in turn, must be greater than or equal to φ_w.

During the grid search, for each pair (UH_1, UH_2) of User Histories such that $u_1, u_2 \in U$:

- We computed the value of the refined similarity γ_{12}^b corresponding to the baseline values of φ_r, φ_c and φ_w.
- For each parameter combination of interest for the grid search:
  - We computed the value of the refined similarity γ_{12} corresponding to this combination.
  - We computed the value of the parameter $\Delta\gamma_{12} = \frac{|\gamma_{12} - \gamma_{12}^b|}{max(\gamma_{12}, \gamma_{12}^b)}$, which is an indicator of the deviation between the refined similarity value obtained with that combination and the one obtained with the baseline.

Finally, for each parameter combination considered above, we computed the average $\Delta\gamma$ of the values $\Delta\gamma_{12}$ corresponding to all the pairs (UH_1, UH_2) of User Histories such that $u_1, u_2 \in U$. The values of $\Delta\gamma$ for the baseline and the other parameter combinations are reported in Table 7.

From the analysis of this table we can see that our refined similarity computation approach is very stable and extremely resilient against variations of the values of the parameters φ_r, φ_c and φ_w as long as such variations are reasonable. In fact, the variations of the mean values of the similarities

φ_r	φ_c	φ_w	$\Delta\gamma$	φ_r	φ_c	φ_w	$\Delta\gamma$
0.50	**0.35**	**0.15**	**baseline**	0.50	0.40	0.10	0.03
0.50	0.42	0.08	0.04	0.50	0.44	0.06	0.06
0.50	0.46	0.04	0.08	0.48	0.32	0.20	0.04
0.46	0.32	0.22	0.06	0.46	0.28	0.26	0.09
0.44	0.30	0.26	0.09	0.44	0.32	0.24	0.08
0.42	0.34	0.24	0.06	0.42	0.32	0.26	0.09
0.42	0.36	0.22	0.07	0.40	0.40	0.20	0.07
0.40	0.35	0.25	0.08	0.52	0.38	0.10	0.03
0.52	0.40	0.08	0.04	0.54	0.38	0.08	0.04
0.54	0.36	0.10	0.03	0.56	0.34	0.10	0.04
0.56	0.36	0.08	0.04	0.58	0.32	0.10	0.07
0.58	0.30	0.12	0.07	0.60	0.30	0.10	0.07
0.60	0.28	0.12	0.08	0.60	0.25	0.15	0.08
0.60	0.32	0.08	0.09	0.60	0.35	0.05	0.08

Table 7: Values of $\Delta\gamma$ for the baseline and other combinations of the values of φ_r, φ_c, φ_w

between the pairs of User Histories (i.e., the values of $\Delta\gamma$) are very limited and range within the real interval $[0.03, 0.09]$ (recall that the possible range of values of this parameter is the real interval $[0, 1]$). Consequently, the baseline we have defined is very reasonable. However, should we have made errors in setting the values of this baseline (as long as they are reasonable), the errors in the final results returned by the refined comparison approach would be very limited being them at most on the order of 10%.

4.6 Comparison of the results provided by the naive and refined approaches

Both in the Introduction and in Sections 3.2.2 and 3.2.3 we pointed out that our framework performs the comparison of User Histories in two steps. During the first step it adopts a naive approach to greatly reduce the number of User Histories to be compared. User Histories judged potentially similar by the naive approach are then examined by the refined one for the final computation of their similarity degree. In this section, we want to see if there is a rough agreement between the results provided by the naive and refined approaches. Indeed, if this were not the case, the filtering performed by the naive approach would risk eliminating many potentially interesting pairs of User Histories. On the other hand, if the naive and refined approaches gave exactly the same results, then the latter would be useless, because it would be much more computationally expensive.

As we mentioned in Section 4.4, we generally decided to keep the pairs of User Histories whose naive similarity degree is greater than a threshold th_s, which we generally set to 0.50. In this experiment, however, in order to avoid the risk of filtering out, through the naive approach, a large number of pairs that would be considered similar by the refined one, we decided to take all pairs of User Histories with a naive similarity degree greater than or equal to 0.40. This seemed to us a good trade-off between the need mentioned above and that of not taking a number of pairs of User Histories that would become computationally difficult to handle. And, indeed, if we set $th_s = 0.4$, instead of $th_s = 0.5$, the number of pairs of User Histories we must consider in this experiment increases from 841,815,080 to

1,636,862,689.

For each of these pairs we calculated its similarity degree using the naive and refined approaches. Then we constructed a diagram in which we reported the naive similarity degree on the x-axis and the refined similarity degree on the y-axis. This diagram is shown in Figure 6.

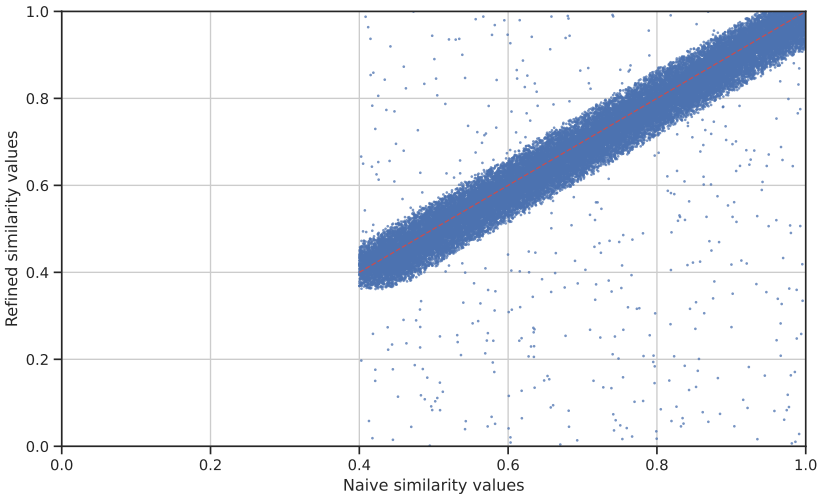

Figure 6: Refined similarity values against naive similarity values for pairs of User Histories

From the analysis of this figure we can observe that there is a marked concordance between the naive and refined similarity values. Generally, given a certain naive value, the corresponding refined one is in a reasonable range around the naive value. This is a confirmation that the pre-filtering performed by the naive approach does not distort results. On the other hand, it can be observed that there are some cases, albeit not very numerous, in which the refined similarity value is much higher or much lower than the naive one. This is a confirmation that the refined approach, although more expensive, can still provide a more accurate result and, therefore, is worth adopting.

As a further analysis on this issue, we calculated the distribution of the differences between the similarity values returned by the refined and the naive approaches. This distribution is shown in Figure 7. The analysis of this figure fully confirms all the considerations we have drawn from the examination of Figure 6.

5 Discussion

As mentioned in Section 3.1.2, our framework is designed to work primarily with content-based OSPs. It is therefore optimized for this type of social media. As evidence of this, the two fundamental concepts stored in the nodes of the multimodal multilayer network-based model adopted by our framework are those of user and content, which are modeled through nodes. Interactions between users and content are modeled by intra-layer and inter-layer edges (see Section 3.1.2).

Actually, our framework provides a specific modeling to represent interactions. In fact, a given

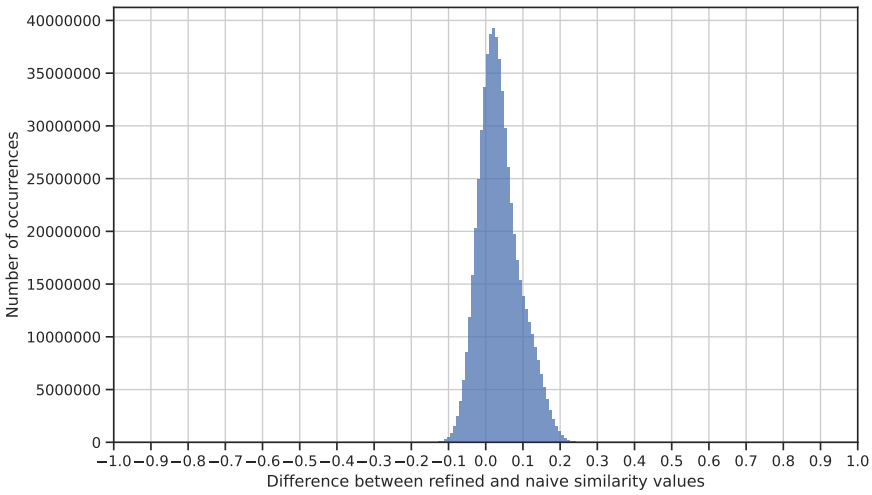

Figure 7: Distribution of the differences between the refined and the naive similarity values for pairs of User Histories

interaction I is represented as $I = (u, c, r, ts, w)$, where u is the user who performs it, c is the content on which it is performed, r is its type, ts is the time slice, and w is the number of times u has performed r on c during ts (see Equation 3.3).

Therefore, our framework currently provides partial handling of multiple interactions. In particular, it can handle the presence of interactions of different types performed by a user u on a given content c in a given time slice ts, as well as the presence of interactions of the same type performed by two different users u_1 and u_2 on the same content c, or by a user u on two different contents c_1 and c_2. The only case of multiple interactions that our framework approximates is when a user u performs multiple interactions of the same type r on the same content c in a given time slice ts. In this case, in fact, only the number of performed interactions is stored and not their list (and thus their timestamps). This is because it is designed for content-based OSPs, not interaction-based OSPs. In other words, it is more suitable for networks that are not very dynamic (as content-based OSPs are) than for very dynamic networks (as interaction-based OSPs are).

However, our framework and associated model could be easily modified to fully handle multiple interactions and thus an interaction-based OSP. In particular, it would be sufficient to represent an interaction I as follows:

$$I = (u, c, r, ts, \mathcal{T}) \tag{5.1}$$

where $\mathcal{T} = \{\tau_1, \tau_2, \cdots, \tau_v\}$ represents the timestamps contained in the time slice ts in which the user u performed the interaction r on the content c.

Although this model is not very different from the previous one, it is much more suitable for handling an interaction-based OSP. The following two examples give an idea of this fact.

- Suppose two users u_1 and u_2 had 10 interactions of type r on c in the time slice ts. Now suppose that u_1 made (almost) all of the interactions at the beginning of ts, while u_2 made (almost) all of the interactions at the end of ts. Although the total number of interactions made in ts is the same (and thus the value of w in Equation 3.3 is 10 in both cases), the time evolution of the interactions (specified by the sequence $\mathcal{T}$ in Equation 5.1) tells us that, at the end of ts, u_1 tends to stop interacting with c, while u_2 is still very active in making interactions on c.

- Consider two users u_1 and u_2 who, within a given time slice ts, engage in a series of interactions to accuse and provoke each other. The current interaction model (Equation 3.3) would only be able to quantify how many provocative interactions each of the two users performed in the time slice ts. In contrast, if the model of Equation 5.1 were adopted, we would be able to know the time sequence of provocative interactions and thus be able to tell whether the main person responsible for such a discussion is u_1 (e.g., because she/he has made several provocative interactions with u_2, who has not responded for some time) or u_2 (because she/he has sent a series of provocative responses and comments to u_1 in response to a simple remark by the latter), or whether both are responsible at the same level (because they exchange continuous and compulsive provocative interactions).

As a further interesting observation, we note that Equations 3.5 and 3.6, which define the User History of a user u_i, continue to be valid even if the definition of I in Equation 3.3 is replaced by the definition of I in Equation 5.1. In this case, however, the level of detail with which a User History is represented and the descriptive power of the two equations is greatly increased because we are able to reconstruct the time sequence of all (possibly even multiple) interactions of all users in all time slices that form the time period of interest.

One might ask why interaction modeling based on Equation 5.1 is not always used, since it appears to be more powerful than that based on Equation 3.3. The reason lies in the resources it requires. Storing the entire list of interactions of all users with their corresponding timestamps requires a much larger amount of space; moreover, performing the various processing on such lists, rather than on the corresponding cardinalities, takes much more time. In interaction-based contexts, where interactions play a key role, the use of more resources is compensated by the results on interactions, which are much more accurate. In content-based contexts, interactions are much less important, and results on interactions may be more approximate because the content component is of much greater interest. In these cases, the resource cost of managing the interaction list is not justified by the results obtained, which, as far as the content component of the OSP is concerned, would be identical to those obtained using Equation 3.3 instead of Equation 5.1.

A final consideration concerns the duration of the time slice. This is clearly related to the level of dynamicity of the OSP. In content-based and more static OSPs, the duration of the time slice can be longer (e.g., for X we chose a duration equal to one day). Conversely, in interaction-based OSPs, which are more dynamic, the duration of the time slice must be shorter (e.g., 12 hours or even less in the case of very high or continuous interaction flow).

6 Conclusion

In this paper, we have proposed a framework that uses a multimodal multilayer network to build, manage and compare User Histories operating on X. We first have clarified the difference between a User History and a User Profile by specifying that the latter focuses mainly on the contents of interest to the user, while the former considers not only the contents but also her interactions and, more generally, her behavior. Next, we have proposed two data models to represent all the information of interest to the framework. The first is based on a single-layer multimodal network; it is very compact but is difficult to use by the approaches implemented in our framework. The second is based on a multimodal multilayer network that represents all available and necessary information in a more immediate and easier to process way. After defining the two supporting models, we have illustrated the approaches for calculating the similarity between two User Histories. In particular, we have first proposed a naive approach, which considers only interaction types. It is computationally simple but returns approximate results. Next, we have illustrated a refined approach that considers not only the types of interactions but also the contents they refer to and their number. It provides much more accurate results but is computationally expensive. To perform a trade-off between accuracy and computational cost, our framework first applies the naive approach to filter out pairs of users whose histories are presumably dissimilar. Then, on the remaining pairs, it applies the refined approach to determine their similarity degree more rigorously. In the paper, alongside already existing applications benefiting from the knowledge of User Histories in an OSP, we have mentioned a new one, which can employ User Histories to increase the level of inclusiveness of an e-learning system. Finally, we have illustrated several experiments we conducted to evaluate the effectiveness and the efficiency of the proposed framework.

Our framework should not be considered an ending point, but rather a starting point. Indeed, it could be improved in several directions. The first of them is to define new instances of our multilayer network-based model customized to other very common Online Social Platforms, such as Facebook, Reddit and LinkedIn. This should allow us to check whether the model proposed in this paper is sufficiently general, or needs an additional level of generalization and abstraction to operate indifferently on any Online Social Platform. After that, we might consider enriching User Histories in such a way as to store additional user-related information (e.g., geographic information). This would lead to an enrichment of models and to a redesign of the approach for the refined comparison of User Histories. On the other hand, the new information would pave the way to further applications benefiting from it. A further development could be in the context of Industry 4.0. In this case, we could use our framework to study the interactions between machines in a production line. In this way, we could define "Machine Histories", analogous to User Histories, which could be employed in various applications, for example in identifying a machine similar or compatible to one that crashed or must be replaced because it is obsolete. An additional context of interest could be the financial one. In this case, we could use our framework to study interactions between financial operators or between single investors. In this way, we could reconstruct their histories and, consequently, their tendency to make risky, balanced or conservative investments, etc. In this case, the similarity between User Histories could help to cluster financial operators or single investors into groups with homogeneous behaviors, or to find an investor who wants to take over the investment of another one with a similar history,

but who needs to exit a certain investment for some supervening reason.

References

[1] A.S.M. Alharbi and E. de Doncker. Twitter sentiment analysis with a deep neural network: An enhanced approach using user behavioral information. *Cognitive Systems Research*, 54:50–61, 2019. Elsevier.

[2] J. Andreu-Perez, C.C.Y. Poon, R.D. Merrifield, S.T.C. Wong, and G.Z. Yang. Big data for health. *IEEE Journal of Biomedical and Health Informatics*, 19(4):1193–1208, 2015. IEEE.

[3] O. Artime, B. Benigni, G. Bertagnolli, V. d'Andrea, R. Gallotti, A. Ghavasieh, S. Raimondo, and M. De Domenico. Multilayer Network Science: From Cells to Societies. *Elements in Structure and Dynamics of Complex Networks*, 2022. Cambridge University Press.

[4] K. Balasubramaniam, S. Vidhya, N. Jayapandian, K. Ramya, M. Poongodi, M. Hamdi, and G.B. Tunze. Social network user profiling with multilayer semantic modeling using ego network. *International Journal of Information Technology and Web Engineering*, 17(1):1–14, 2022. IGI Global.

[5] F. Baumann, P. Lorenz-Spreen, I.M. Sokolov, and M. Starnini. Modeling echo chambers and polarization dynamics in social networks. *Physical Review Letters*, 124(4):048301, 2020. APS.

[6] M. Bazzi, M. A. Porter, S. Williams, M. McDonald, D. J. Fenn, and S. D. Howison. Community detection in temporal multilayer networks, with an application to correlation networks. *Multiscale Modeling & Simulation*, 14(1):1–41, 2016. SIAM.

[7] M.L. Bernacki, M.J. Greene, and N.G. Lobczowski. A systematic review of research on personalized learning: Personalized by whom, to what, how, and for what purpose (s)? *Educational Psychology Review*, 33(4):1675–1715, 2021. Springer.

[8] P.V. Bindu, P.S. Thilagam, and D. Ahuja. Discovering suspicious behavior in multilayer social networks. *Computers in Human Behavior*, 73:568–582, 2017. Elsevier.

[9] A. Biondo, A. Pluchino, and A. Rapisarda. Informative contagion dynamics in a multilayer network model of financial markets. *Italian Economic Journal*, 3:343–366, 2017. Springer.

[10] D.M. Blei, A.Y. Ng, and M.I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003. Microtone Publishing.

[11] K. Bok, G. Ko, J. Lim, and J. Yoo. Personalized content recommendation scheme based on trust in online social networks. *Concurrency and Computation: Practice and Experience*, 32(18):e5572, 2020. Wiley Online Library.

[12] G. Bonifazi, F. Cauteruccio, E. Corradini, M. Marchetti, G. Terracina, D. Ursino, and L. Virgili. Representation, detection and usage of the content semantics of comments in a social platform. *Journal of Information Science*, 2022. SAGE.

[13] 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.

[14] 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.

[15] 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*, 48:783–810, 2022. SAGE.

[16] F. Cauteruccio, G. Terracina, and D. Ursino. Generalizing identity-based string comparison metrics: Framework and Techniques. *Knowledge-Based Systems*, 187:104820, 2020. Elsevier.

[17] E.B. Choi, J. Kim, D. Jeong, E. Park, and A. P. del Pobil. Detecting agro: Korean trolling and clickbaiting behaviour in online environments. *Journal of Information Science*, 2023. SAGE.

[18] K. Cohen, F. Johansson, L. Kaati, and J.C. Mork. Detecting linguistic markers for radical violence in social media. *Terrorism and Political Violence*, 26(1):246–256, 2014. Taylor & Francis.

[19] A. Culotta. Towards detecting influenza epidemics by analyzing twitter messages. In *Proc. of the First Workshop on Social Media Analytics (SOMA'10)*, pages 115–122, Washington D.C., USA, 2010. ACM.

[20] N. Dakiche, F.B. Tayeb, Y. Slimani, and K. Benatchba. Tracking community evolution in social networks: A survey. *Information Processing & Management*, 56(3):1084–1102, 2019. Elsevier.

[21] 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.

[22] M. E. Dickison, M. Magnani, and L. Rossi. *Multilayer social networks*. 2016. Cambridge University Press.

[23] D.N. Fisher and N. Pinter-Wollman. Using multilayer network analysis to explore the temporal dynamics of collective behavior. *Current zoology*, 67(1):71–80, 2021. Oxford University Press.

[24] Y. Ge, L. Liu, X. Qiu, H. Song, Y. Wang, and K. Huang. A framework of multilayer social networks for communication behavior with agent-based modeling. *Simulation*, 89(7):810–828, 2013. Sage Publications Sage UK: London, England.

[25] M. Grootendorst. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. *arXiv preprint arXiv:2203.05794*, 2022. arXiv.

[26] J. Han, M. Kamber, and J. Pei. *Data Mining: Concepts and Techniques - Third Edition*. 2011. Morgan Kaufmann notes.

[27] S. Hudson, L. Huang, M.S. Roth, and T.J. Madden. The influence of social media interactions on consumer–brand relationships: A three-country study of brand perceptions and marketing behaviors. *International Journal of Research in Marketing*, 33(1):27–41, 2016. Elsevier.

[28] M.R. Islam, S. Muthiah, and N. Ramakrishnan. NActSeer: Predicting User Actions in Social Network Using Graph Augmented Neural Network. In *Proc. of the ACM International Conference on Information and Knowledge Management (CIKM'19)*, pages 1793–1802, Beijing, China, 2019. ACM.

[29] P. Jaccard. The distribution of the flora in the alpine zone. *New phytologist*, 11(2):37–50, 1912. Wiley Online Library.

[30] M. Jalili, Y. Orouskhani, M. Asgari, N. Alipourfard, and M. Perc. Link prediction in multiplex online social networks. *Royal Society open science*, 4(2):160863, 2017. The Royal Society Publishing.

[31] 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.

[32] K.Z. Khanam, G. Srivastava, and V. Mago. The homophily principle in social network analysis: A survey. *Multimedia Tools and Applications*, 82(6):8811–8854, 2023. Springer.

[33] W. Li, Y. Li, W. Liu, and C. Wang. An influence maximization method based on crowd emotion under an emotion-based attribute social network. *Information Processing & Management*, 59(2):102818, 2022. Elsevier.

[34] H. Liang. Drprofiling: deep reinforcement user profiling for recommendations in heterogenous information networks. *IEEE Transactions on Knowledge and Data Engineering*, 34(4):1723–1734, 2020. IEEE.

[35] Q. Lin, L. Duan, and J. Huang. Personalized Pricing through User Profiling in Social Networks. In *Proc. of the International Symposium on Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks (WiOpt'19)*, pages 1–8, Virtual Conference, 2021. IEEE.

[36] A.P. Logan, P.M. LaCasse, and B.J. Lunday. Social network analysis of Twitter interactions: a directed multilayer network approach. *Social Network Analysis and Mining*, 13(1):65, 2023. Springer.

[37] H. Lugo and M.S. Miguel. Learning and coordinating in a multilayer network. *Scientific reports*, 5(1):7776, 2015. Nature.

[38] X. Luo, C. Jiang, W. Wang, Y. Xu, J.H. Wang, and W. Zhao. User behavior prediction in social networks using weighted extreme learning machine with distribution optimization. *Future Generation Computer Systems*, 93:1023–1035, 2019. Elsevier.

[39] Y. Luo, J. Ma, and C. K. Yeo. Identification of rumour stances by considering network topology and social media comments. *Journal of Information Science*, 48(1):118–130, 2022. SAGE.

[40] P. R. Monge and N. S. Contractor. *Theories of communication networks*. 2003. Oxford University Press, USA.

[41] Y. Murase, J. Török, H. H. Jo, K. Kaski, and J. Kertész. Multilayer weighted social network model. *Physical Review E*, 90(5):052810, 2014. APS.

[42] E. Oro, C. Pizzuti, N. Procopio, and M. Ruffolo. Detecting topic authoritative social media users: a multilayer network approach. *IEEE Transactions on Multimedia*, 20(5):1195–1208, 2017. IEEE.

[43] B. Oselio, A. Kulesza, and A. O. Hero. Multi-layer graph analysis for dynamic social networks. *IEEE Journal of Selected Topics in Signal Processing*, 8(4):514–523, 2014. IEEE.

[44] F. Pierri, C. Piccardi, and S. Ceri. A multi-layer approach to disinformation detection in US and Italian news spreading on Twitter. *EPJ Data Science*, 9(1):35, 2020. Springer.

[45] S. Poledna, J. L. Molina-Borboa, S. Martínez-Jaramillo, M. Van Der Leij, and S. Thurner. The multi-layer network nature of systemic risk and its implications for the costs of financial crises. *Journal of Financial Stability*, 20:70–81, 2015. Elsevier.

[46] D. Preoţiuc-Pietro and T. Cohn. Mining user behaviours: a study of check-in patterns in location based social networks. In *Proc. of the ACM Web Science Conference (WebSci'13)*, pages 306–315, Paris, France, 2013. ACM.

[47] M. Razghandi and S.A.H. Golpaygani. A context-aware and user behavior-based recommender system with regarding social network analysis. In *Proc. of the International Conference on e-Business Engineering (ICEBE'17)*, pages 208–213, Shanghai, China, 2017. IEEE.

[48] J.E. Rowley and B.J. Keegan. An overview of systematic literature reviews in social media marketing. *Journal of Information Science*, 46(6):725–738, 2020. SAGE.

[49] C. Toraman, F. Şahinuç, E.H. Yilmaz, and I.B. Akkaya. Understanding social engagements: A comparative analysis of user and text features in Twitter. *Social Network Analysis and Mining*, 12(1):1–16, 2022. Springer.

[50] M. Tsvetovat and A. Kouznetsov. *Social Network Analysis for Startups: Finding connections on the social web*. Sebastopol, CA, USA, 2011. O'Reilly Media, Inc.

[51] A. van der Marel, S. Prasher, C. Carminito, C. L. O'connell, A. Phillips, B. M. Kluever, and E. A. Hobson. A framework to evaluate whether to pool or separate behaviors in a multilayer network. *Current Zoology*, 67(1):101–111, 2021. Oxford University Press.

[52] L. Vassio, M. Garetto, C. Chiasserini, and E. Leonardi. Temporal dynamics of posts and user engagement of influencers on facebook and instagram. In *Proc. of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM'21)*, pages 129–133, Virtual event, 2021. ACM.

[53] R.A. Wagner and M.J. Fischer. The string-to-string correction problem. *Journal of the ACM*, 21(1):168–173, 1974. ACM.

[54] H. Yin, B. Cui, L. Chen, Z. Hu, and Z. Huang. A temporal context-aware model for user behavior modeling in social media systems. In *Proc. of the International Conference on Management of Data (SIGMOD/PODS'14)*, pages 1543–1554, Snowbird, UT, USA, 2014. ACM.