UNIVERSITÀ POLITECNICA DELLE MARCHE
CORSO DI DOTTORATO DI RICERCA IN INGEGNERIA DELL'INFORMAZIONE
CURRICULUM IN INGEGNERIA INFORMATICA, GESTIONALE E DELL'AUTOMAZIONE

Deep Learning based Decision Support Systems for Quality Control task in Industry 4.0

Ph.D. Dissertation of:
Riccardo Rosati

Advisor:
Prof. Emanuele Frontoni

Coadvisor:
Luca Romeo, PhD

Curriculum Supervisor:
Prof. Franco Chiaraluce

XXI edition - new series

UNIVERSITÀ POLITECNICA DELLE MARCHE
CORSO DI DOTTORATO DI RICERCA IN INGEGNERIA DELL'INFORMAZIONE
CURRICULUM IN INGEGNERIA INFORMATICA, GESTIONALE E DELL'AUTOMAZIONE

Deep Learning based Decision Support Systems for Quality Control task in Industry 4.0

Ph.D. Dissertation of:
Riccardo Rosati

Advisor:
Prof. Emanuele Frontoni

Coadvisor:
Luca Romeo, PhD

Curriculum Supervisor:
Prof. Franco Chiaraluce

XXI edition - new series

UNIVERSITÀ POLITECNICA DELLE MARCHE
CORSO DI DOTTORATO DI RICERCA IN INGEGNERIA DELL'INFORMAZIONE
FACOLTÀ DI INGEGNERIA
Via Brecce Bianche – 60131 Ancona (AN), Italy

A Letizia,
alla mia famiglia,
ai miei nonni

"In the middle of difficulty lies opportunity."
A. Einstein

"Est modus in rebus: sunt certi denique fines,
quos ultra citraque nequit consistere rectum."
Orazio

Abstract

In the context of Industry 4.0, the increasing amount of data in combination with novel disruptive Machine Learning (ML) and Deep Learning (DL) methodologies lied the foundation for helping the human operator to detect production issues as well as classify the quality of the final product. These solutions, embedded into Decision Support Systems (DSSs), offer great opportunities to automatize the overall quality control process.

The objectives and the contributions of this thesis reflect the research activities performed on the topics of: i) Predictive Quality Control (PQC), with the aim to design and implement a DSS for predicting the processing quality and anomaly situations during the machining of a tool; ii) Aesthetic Quality Control (AQC), with the aim to design and implement a DSS for evaluating the aesthetic properties of the material for a manufactured product.

In the first topic, the author presents a DSS comprised of the following cornerstones: data collection, feature extraction, predictive model, cloud storage, and data analysis interface. Differently from the related literature, the proposed approach is based on a feature extraction strategy and a ML model powered by specific topics collected on the lower and upper levels of the production system, allowing the acquisition of high-quality labeled data, which are suitable for supervised ML approach. Compared with respect to other state of the art ML models, the experimental results demonstrated how the proposed approach is the best trade-off between predictive performance, computation effort, and interpretability for the prediction of processing quality.

In the second topic, the author proposes the application of novel ordinal DL methodologies to improve the classification performance in assessing the aesthetic quality of wooden stocks while addressing typical challenges in AQC task, i.e. the bias mitigation, the error minimization between distant classes, the noise in labeling process and the exploitation of ordinal and hierarchical constraints of the categories. The final aim is to support the human operator in the final decision. The proposed approaches are evaluated on a real-world dataset and compared with other state of the art DL methods. For each challenge, the experimental results demonstrate the effectiveness of each proposed approach.

Sommario

Nel contesto dell'Industria 4.0, la crescente quantità di dati in congiunzione con le nuove metodologie di Machine Learning (ML) e Deep Learning (DL) ha posto le basi per aiutare l'operatore umano a rilevare i problemi di produzione e a classificare la qualità del prodotto finale. Queste soluzioni, implementate in sistemi di supporto alle decisioni (DSS), offrono grandi opportunità per automatizzare complessivamente il processo di controllo qualità.

Gli obiettivi e i contributi di questa tesi riflettono le attività di ricerca svolte sui temi: i) Controllo Qualità Predittivo (PQC), con l'obiettivo di progettare e implementare un DSS per la previsione della qualità di lavorazione e delle situazioni di anomalia durante la lavorazione di un utensile; ii) Controllo Qualità Estetico (AQC), con l'obiettivo di progettare e implementare un DSS per la valutazione delle proprietà estetiche del materiale per un prodotto finito.

Nel primo argomento, l'autore presenta un DSS composto dai seguenti punti chiave: raccolta dei dati, estrazione delle features, modello predittivo, archiviazione su cloud e interfaccia di analisi dei dati. Diversamente dalla letteratura, l'approccio proposto si basa su una strategia di estrazione delle features e su un modello di ML alimentato da topic specifici raccolti ai livelli inferiori e superiori del sistema di produzione, consentendo l'acquisizione di dati annotati di alta qualità, adatti ad un approccio di ML supervisionato. Rispetto ad altri modelli di ML allo stato dell'arte, i risultati sperimentali hanno dimostrato come l'approccio proposto rappresenti il miglior compromesso tra prestazioni predittive, costo computazionale e interpretabilità per la previsione della qualità di lavorazione.

Nel secondo argomento, l'autore propone l'applicazione di metodologie di DL per migliorare le prestazioni di classificazione nella valutazione della qualità estetica dei calci di fucile in legno, affrontando le sfide tipiche del compito di AQC, ovvero la mitigazione dei bias, la minimizzazione dell'errore tra classi distanti, il rumore nel processo di annotazione e lo sfruttamento dei vincoli ordinali e gerarchici delle classi del dataset. L'obiettivo finale è quello di supportare l'operatore umano nella decisione finale. Gli approcci proposti sono stati valutati su un set di dati reali e confrontati con altri metodi DL allo stato dell'arte. Per ogni sfida, i risultati sperimentali dimostrano l'efficacia degli approcci proposti.

Acronyms

AI Artificial Intelligence

ANN Artificial Neural Network

AQC Aesthetic Quality Control

ATMs Automated Teller Machines

BCE Binary Cross-Entropy

BN Batch Normalization

CCE Categorical Cross-Entropy

CCR Correct Classification Rate

CLM Cumulative Link Model

CMM Coordinate measuring machine

CNN Convolutional Neural Network

CV Cross Validation

DL Deep Learning

DSS Decision Support System

DT Decision Tree

ECOC Error Correcting Output Codes

FC Fully Connected

FMS Flexible Manufacturing System

GLB Global

HCLM Hierarchical cumulative link model

HMC Multi-label classification

HMCN Hierarchical multi-label classification network

HOBD Hierarchical ordinal binary decomposition

ICT Information and Communication Technologies

KNN K-Nearest Neighbors

KPI Key Performance Indicator

LCPN Local Classifier per Parent Node

LR Linear Regression

LSTM Long Short Term Memory

MAE Mean Absolute Error

ML Machine Learning

MLP Multi-Layer Perceptron

MQTT Message Queuing Telemetry Transport

MS Minimum Sensitivity

MSE Mean Squared Error

MTL Multi-Task Learning

NB Naive Bayes

OBD Ordinal Binary Decomposition

PdM Predictive Maintenance

PQC Predictive Quality Control

QC Quality Control

QWK Quadratic Weighted Kappa

ROI Region Of Interest

RF Random Forest

ROC Receiver Operating Characteristic

RT Regression Tree

RUL Remaining Useful Life

SVM Support Vector Machine

VE Voting Ensemble

XGB XGBoost

ZDM Zero Defect Manufacturing

Contents

List of Figures

List of Tables

Chapter 1.

Introduction

The fourth industrial revolution, also known as *Industry 4.0*, is the result of a rapid manufacturing innovation which is leading to an ever-increasing number of services. The growing influence of Information and Communication Technologies (ICT) in production systems allows the collection of Big Data [1] that can be analyzed and used by companies to create innovative applications, enabling optimization of products and services [2]. This increasing availability of data and related analytic processes, along with new technological advances such as high-computing power and large storage capacity, have changed the paradigm in the fabrication of products and services [3]. The combination of Big Data and Artificial Intelligence (AI) approaches lied the foundation for evolving towards intelligent manufacturing and smart machines [4]. In particular, Machine Learning (ML) and Deep Learning (DL) techniques have become appealing solutions in different industrial areas such as predictive maintenance [5, 6], decision support systems [7, 8] and quality control [9, 10, 11]. Using AI methodologies, the industry scenario has started to reduce machine failure, improve quality control procedure, increase productivity, and substantially lower the production costs. In fact, it is currently possible to predict machine and product failures or make recommendations to human operators for saving costs and time [4, 12].

Among all the applications of AI in Industry 4.0, in recent years the Quality Control (QC) task has undergone a major boost in research and innovation, defining the Quality 4.0 domain. In the real world, many tasks involve the classification of a given manufactured item depending on its quality, according to the paradigm of Zero Defect Manufacturing (ZDM). ZDM aims for the complete elimination of defects, not simply through detection and correction of defective products and process parameters, but also through defect prediction and prevention [13]. In this sense, AI frameworks can aid in two ways: i) for predicting the quality of the products throughout their lifecycles; ii) for assessing the absence of defects on the finished products. In the first case, the quality is referred to the evaluation of non-conformities that could impact the capabilities of the product. This compliance of the products can be analyzed using data analytic via improved predictive methods on manufactured products. In the second case, the task refers to the detection of defects that affect the aesthetic properties of the finished product. According to [14], ZDM can be implemented in two different

way, i.e. *Process-oriented* ZDM, which evaluates the status of each product based on the presence of defects in manufacturing system, and *Product-oriented* ZDM, which focuses on the actual part defects by finding a solution to counteract them (see Fig. 1.1). Inspiring from this framework, the role of Decision Support System (DSS), based on novel AI methodologies, has been highlighted to bridge the gap between smart factory advances and the effective application of these technologies for quality control procedures.

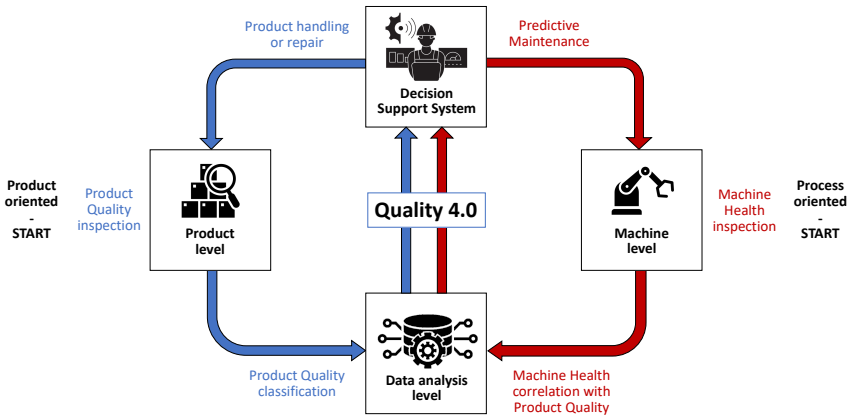


Figure 1.1.: Quality 4.0 framework: the role of Decision Support System and Machine Learning/Deep Learning methods for solving two typical Quality Control applications in Industry 4.0 scenario, i.e. process-oriented and product-oriented Zero Defect Manufacturing.

In this context, the objectives and the contributions of this thesis reflect the research activities performed on the following two main topics:

- Design and implementation of a DSS for **Predictive Quality Control** (PQC) for predicting the processing quality and anomaly situations during the machining of a tool;
- Design and implementation of a DSS for **Aesthetic Quality Control** (AQC) for evaluating the aesthetic properties of a material for manufactured product.

The remainder of this chapter is organized as follows:

- Section 1.1 provides a background and motivation for this thesis topic, motivating the research study and highlights its relevance from the perspectives of AI and DSS in the context of Quality 4.0 and related challenges;

- Section 1.2 presents the main objectives of the thesis, where the problem is formally defined from ML/DL point of view with a list of specific research questions answered in this dissertation;
- Section 1.3 resumes the organization of the thesis;
- Section 1.4 presents the thesis outcomes in terms of scientific publications.

1.1. Background and motivation

In this section, an in-depth analysis of the Quality Control tasks and the corresponding importance of Decision Support Systems in the Industry 4.0 scenario is provided. Then, the challenges present in this context are highlighted.

1.1.1. Quality 4.0 applications

Quality Control (QC) is a growing area in Industry 4.0 and an important step in every production system, representing one of the main differentiating factors in production. In fact, the rapid timing of distribution pushes toward a shortening of time-to-market, which inevitably has an impact on manufacturing processes. But releasing poor quality or defective products to the market can have severe consequences for the company:

- Material consequences: extra costs for disposal of materials, product replacements, higher cost of production;
- Financial consequences: late delivery penalties;
- Intangible consequences: loss of consumer trust, complaints, deterioration of brand image.

For this reason, during manufacturing, quality must be maintained by which Quality 4.0 technologies are available to fulfil major challenges in this field. AI methodologies, Cloud computing, Augmented and Virtual Reality, and Internet of Things (IoT) are all essential tools in this context [3]. In particular, the increasing amount of data in combination with novel disruptive ML/DL frameworks lied the foundation for helping to detect production issues as well as classify the quality of the final product. These solutions offer great opportunities to automatize the overall QC process: the benefit of this approach can be summarized as the saving of time and resources, the minimization of human variability and the increase in production performance.

In the real world, many tasks involve the classification of a given manufactured item depending on its quality. In some cases, this quality is referred to the absence of defects that could impact the capabilities of the product. While these quality control tasks are crucial for some business-to-business industries where any defect can represent a prominent quality problem, there are other scenarios like business-to-consumer

industries, where the aesthetic quality is more important, like in the automotive or weapons industry. In this context, the finished product must guarantee high performance not only at the mechanical level, but also at the aesthetic one according to the expectations of the customer, with the aim to make a manufactured item with excellent perceived quality [15]. This distinction leads to the definition of two different QC applications: Predictive Quality Control (PQC) and Aesthetic Quality Control (AQC), which are described below.

Predictive Quality Control

Defect, anomaly and fault prediction is the main focus for Predictive Quality Control (PQC), which aims to forecast the quality of the product before or during its production via specific models and historical data process [16], [17]. The final goal is to monitor manufacturing machinery and the corresponding produced quality: the Remaining Useful Life (RUL) of the equipment is estimated for scheduling optimal maintenance actions with the aim to avoid production line downtime and interruptions. In literature, this task is defined as Predictive Maintenance (PdM), thus timely performing preventive replacements that allow both to prevent unexpected failures and minimise total maintenance costs [18], [19]. Referring to the framework in Figure 1.1, this QC can be intended as process-oriented approach.

Aesthetic Quality Control

Even before the rise of quality 4.0, in the industrial setting there is a widespread use of vision tools for the automation of QC procedures focusing on quantitative and deterministic analysis of the product, which aim to ensure that it complies with production requirements. These techniques are conceived to verify, for instance, the dimensional or roughness inspection of materials or any other measurable parameter [20], [21],[22]. However, there is a lack of methodologies that allow the modeling and generalization of qualitative analyses. In this case, the aim of Aesthetic Quality Control (AQC) is to determine the quality grade of a product according to non-metric aesthetic canons. Differentiating the aesthetic level of a material or manufactured item is often needed for estimating the commercial value of the final product. In this sense, DL techniques enable to learn objective rules achieving high performances and ensuring the results repeatability, without being influenced by the subjectivity and variability over time as for the human operator. AQC can be considered as a product-oriented approach in the paradigm shown in Figure 1.1.

1.1.2. The role of Decision Support Systems

Starting from the concept of Quality 4.0 and reaching the application in real manufacturing processes, the missing gap is the development of decision analytics in between

[23]. The human perception is not able to deal with industrial systems generating a large amount of data; on the other hand, computers and intelligent systems can process large amounts of data fast, reliably, and accurately, but having no awareness in decision-making processes. Decision Support System (DSS) is an effective tool for bridging this gap, helping decision maker to cope with the identified complex problem processed by machines. At the same time, this allows for the establishment of the human-in-the-loop paradigm: when the human factor is kept in the loop, capabilities become extremely flexible and the existing knowledge in people-centered processes is maintained [13].

Technological advances in Industry 4.0 have recently enabled significant progress in knowledge representation models and learning algorithms enhancing DSS capabilities and their use. For this reason, DSSs are now integrating multiple functions such as analysis, modeling, prediction, optimization and diagnosis. In particular DSSs are used for different purposes in a manufacturing industry, such as process control [8], machine design [24], managers assistance [25], customer satisfaction [26], supply chain [27], etc.

In the industrial scenario, DSSs are suited for knowledge capturing, transferring and distribution between workforce as well as production phases. In this sense, the relation between human processes and quality has been rarely investigated, for clear data privacy issues and modelling challenges [13]. For this reason, according also with the ethic guidelines by the European Commission (Human agency and oversight, [28]), the goal of research should be to develop solutions that optimize human knowledge and capabilities, towards a digitalization process which can enhance the human operator, who must always be at the center of manufacturing production [29].

1.1.3. Challenges

Despite the several advantages, smart factory digitalization brings several challenges that should be taken into account when designing a DSS:

- *Data analysis issues*: in the industrial domain, the amount of data and their heterogeneity is enormous. Data are often acquired in large volumes and high dimensions from heterogeneous sources. Data may be noisy and incomplete, with imbalanced classes and ambiguous labeling. Several ML and DL techniques have been developed during recent years to face these type of problems;

- *Models interpretability*: the human being should be able to understand the reasoning about data through data representations. AI models must aim to increase accuracy rates of predictions while also understanding causality through meaningful models;

- *User-centered DSS*: it is important to design a human-centered system for implementing operator support tools during manual or partially manual processes

as well as product and process inspections.

According to these main issues, the following challenges related to PQC and AQC tasks have been identified.

PQC challenges

- *Difficulty of obtaining quality labeled data* [30]: labeled condition monitoring data samples are required for supervised classification in order to establish mapping relationships between input samples and fault types in training phase. Frequently, obtaining quality labeled data is not possible since: i) component wear is not always easily identifiable, ii) frequent data missing values are present and iii) the annotation is poorly structured.

- *Difficulty to ensure model interpretability* [31]: accurate predictive performance and high interpretability are required at the same time for the ML/DL model. Recently many monitoring systems that apply DL methods to historical data have been proposed [32]. Given the complexity of industrial data, these models require a large number of parameters in order to make inferences about system behavior, which is why they are often called "black boxes". As a result of the large number of variables used for parametrizing black-box models, their predictions cannot be traced by humans, not allowing the reconstruction of the decisional path. This contravenes the concept of interpretability required for a DSS.

AQC challenges

- *Limited amount of data in unbalanced setting* [33]: in some production scenarios, certain manufactured pieces are in limited edition and high-end products are made in smaller quantities than more common ones. Accordingly, this implies that samples are not uniformly distributed over all labels. The resulting imbalanced learning problem, due to the presence of underrepresented data or severe class distribution skews, is concerned with the performance of the AI algorithm.

- *Intrinsic Bias* [34]: although the potential of DL methods to learn discriminatory patterns in data is relevant, the inability to detect bias from collected data and the risk to reproduce this bias in model outcome pose a remarkable point in the Industrial 4.0 scenario. Bias may correspond to confound information enclosed in the data or in the model itself. In the DL scenario, the inability to detect bias situations is mainly due to the lack of model interpretability.

- *Minimizing errors between distant classes* [35]: AI models are useful to automatize classification tasks, but very often the error made among problem classes is equally weighted. This is a problem from the real application perspective since

this does not usually happen in human annotation. So the algorithms should minimize misclassification errors among distant classes to effective in a DSS.

- *Task subjectivity and intra-operator variability* [36]: sometimes, AQC is purely dependent on the evaluation and expertise of human eye, resulting in a difficult and subjective task and affecting the repeatability of results. In fact, the task may be affected by intrinsic human operator variability as well as different operating condition. When approaching the problem in a supervised way, these issues are reflected in the ground-truth label, which might not always be reliable due to possible noise or errors in the labeling process.

- *Exploit dataset properties* [37]: most ML and DL algorithms deal with classification problems by considering the target variable as a set of disjointed classes, not exploiting potential structural properties of the data. But several important real-world problems are naturally modeled as a hierarchical structure or disclose an ordinal relation where the distance between classes is not quantifiable a priori.

1.2. Aim of the thesis and research questions

Starting from the research gaps identified in the literature review summarized in Chapter 2, the main objective of this thesis work is to address the challenges above indicated concerning the Predictive Quality Control and Aesthetic Quality Control. The definition of these two Quality 4.0 problems led to the formulation of the following research questions.

1.2.1. Predictive Quality Control problem

The PQC problem is addressed to design and develop a DSS for predicting the processing quality and anomaly situations during the machining of a tool with the purpose of implementing predictive maintenance actions. The research questions are summarized below:

(i) How can supervised ML model be applied to predict the processing quality of a tool starting from a machine sensors raw data?

(ii) How is it possible to obtain and manage reliable data annotation?

(iii) Is it feasible to ensure at the same time high predictive performance and model interpretability?

(iv) Does the proposed ML algorithm outperform standard algorithms widely used in literature?

(v) How can the proposed algorithm be integrated in a DSS to provide suitable feedback for supporting human operator during production stages?

(vi) How should DSS be designed to support real-time data acquisition and model inference?

1.2.2. Aesthetic Quality Control problem

The AQC problem is addressed to design and develop a DSS for evaluating the aesthetic properties of a material for manufactured product with the purpose of supporting the human operator in the final decision. The research questions are summarized below:

(i) How can supervised DL model be applied to perform a classification task based on qualitative aesthetic properties of a material?

(ii) How is it possible to detect and mitigate unwanted bias in data?

(iii) How can errors between distant classes be minimized?

(iv) How is it possible to mitigate noise or errors in labeling process?

(v) Is it feasible to design DL methodology for exploiting ordinal and/or hierarchical properties of the dataset?

(vi) How can the proposed algorithm be integrated in a DSS to provide suitable feedback for supporting human operator during final QC decisions?

1.3. Thesis overview

This thesis aims to answer the aforementioned research questions by proposing DSSs comprising various novel ML and DL methods-based frameworks for Quality 4.0 tasks. These DSSs are designed to face the challenges described in Section 1.1.3. For answering these questions, two real-world industrial use cases were faced.

In particular, the chapters are organized as follows:

- Chapter 2 reviews related literature in two research fields, which are Predictive Quality Control (PQC) (Section 2.1) and Aesthetic Quality Control (AQC) (Section 2.2). The author outlines the strengths and the drawbacks of the existing achievements, focusing on main contributions of this work respect to the state of the art.

- Chapter 3 describes the PQC problem. Firstly, in Section 3.1 a preliminary work is presented about the design of a DSS platform for RUL estimation with interpretable ML model. According to this approach, in Section 3.2 the same

methodology was applied to solve the PQC task related to the prediction of machine processing quality. Section 3.2.1 describes the real use case we aimed to face, Section 3.2.2 presents the proposed DSS framework and Section 3.2.3 the experimental setup and the results obtained.

- Chapter 4 is related to the AQC problem. Section 4.1 and Section 4.2 present the collected dataset and how the task can be addressed by standard DL method, respectively. Then, in Section 4.3 two different strategies are proposed for bias mitigation. Section 4.4 and Section 4.5 describes the proposed ordinal DL methodology in conjunction with a novel exponential loss regularisation. Section 4.6 and Section 4.7 refer to the proposal of hierarchical DL methods for deeply exploiting the structural properties of the dataset. Finally, in Section 4.8 the design of the proposed DSS for AQC is shown.

- Chapter 5 discusses the obtained results and revisits the scientific contributions of this thesis in terms of new methodologies and knowledge created to benefit Quality 4.0 scenario and their validity in real-world industrial applications.

- Chapter 6 concludes the dissertation and provide the directions for the future research work.

1.4. Thesis outcome

The detailed descriptions of the thesis contributions are available in the following publications. For each publication, the main contributions are listed.

Predictive Quality Control

- **Rosati, R.**, Romeo, L., Vargas, V.M., Gutiérrez, P.A., Hervás-Martínez, C., Bianchini, L., Capriotti, A., Capparuccia, R. & Frontoni, E. (2022). Predictive Maintenance of ATM machines by modelling Remaining Useful Life with Machine Learning techniques. In 17th International Conference on Soft Computing Models in Industrial and Environmental Applications.

  The main contributions of the present work (described in Section 3.1) can be summarized as follows:

  (i) the design of a feature engineering stage, performed in collaboration with domain expert maintainers, that ensures to build a representative dataset;

  (ii) the design of an efficient ML strategy to predict the RUL of multiple ATMs;

  (iii) the integration of the algorithm in a scalable cloud-based architecture as the main core of a DSS.

- **Rosati, R.**, Romeo, L., Cecchini, G., Tonetto, F., Viti, P., Mancini, A., & Frontoni, E. (2022). From knowledge-based to big data analytic model: a novel IoT and machine learning based decision support system for predictive maintenance in Industry 4.0. Journal of Intelligent Manufacturing, 1-15.

  The main contributions of our proposed approach (described in Section 3.2) respect to related literature lie in:

  (i) the prediction of processing quality and anomaly situations that are quantitatively annotated in our training dataset by a 3D coordinate measuring machine;

  (ii) the fully automatizing of the feature extraction and prediction step for computing salient Key performance indicators (KPIs), which represent the input of our ML predictive model;

  (iii) the integration of the proposed ML approach in an IoT platform that enables the collection of a huge amount of data and provide actionable decision recommendations for resolving productivity losses and maintenance issue;

  (iv) the combination of a feature extraction technique with a Random Forest (RF) regression model for ensuring the best trade-off between the accuracy prediction, computation effort, and model interpretability.

Aesthetic Quality Control

- Romeo, L., **Rosati, R.**, & Frontoni, E. (2022). Decision Support System Based on Deep Learning for Improving the Quality Control Task of Rifles: A Case Study in Industry 4.0. In Machine Learning and Artificial Intelligence with Industrial Applications (pp. 63-77). Springer, Cham.

  The main contributions of the work, which will be described in Sections 4.1 and 4.2, are summarised below:

  (i) the collection of annotated real dataset specifically tailored to solve the AQC of wooden stocks. Each image is properly annotated by a high specialized technician;

  (ii) the proposing of a DL approach based on VGG-16 and ordinal categorical cross-entropy (CCE) loss in this novel and challenging Quality 4.0 application, i.e. the quality control classification task.

- **Rosati, R.**, Romeo, L., Cecchini, G., Tonetto, F., Perugini, L., Ruggeri, L., Viti, P., & Frontoni, E. (2021). Bias from the Wild Industry 4.0: Are We Really Classifying the Quality or Shotgun Series?. In Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10–15, 2021, Proceedings, Part IV (pp. 637-649). Springer International Publishing.

The main contributions of the work, covered in Section 4.3, are the following:

(i) an in-depth analysis of the DL model to show the risk to reproduce the presence of possible bias in the collected data;

(ii) the proposing of two-stage solution named Hierarchical Unbiased VGG-16 (HUVGG-16) and a voting ensamble methodology for mitigating the detected bias.

- **Rosati, R.**, Romeo, L., Vargas, V. M., Gutiérrez, P. A., Hervás-Martínez, C., & Frontoni, E. (2022). A novel deep ordinal classification approach for aesthetic quality control classification. Neural Computing and Applications, 1-15.

  The main contributions of this work (Sections 4.4) in the field of ordinal classification are:

  (i) the introduction of a DL methodology for ordinal classification specifically tailored for solving an AQC task;

  (ii) the introduction of a DL methodology for ordinal classification based on CLM and CCE, demonstrating how there may be a sort of redundancy between the maximization of an ordinal loss and the modeling of cumulative distribution and imposing the ordinal constraint via the thresholds and slope parameters;

  (iii) the demonstration on how the proposed methodology is able from one side to reduce misclassification errors among distant classes (which is a relevant aspect for the real use-case) and from the other side to reduce the bias factor related to other image features.

- Vargas,V.M., Gutierrez, P.A., **Rosati, R.**, Romeo, L., Frontoni, E. & Hervás-Martínez, C. Exponential loss regularisation for encouraging ordinal constraint to shotgun stocks quality assessment. Applied Soft Computing. (minor revision)

  The main contribution of this work described in Section 4.5 is the proposal to apply the L_p norm to the exponential regularisation that is described in Section 4.5.1 for obtaining soft labels with a more flexible distribution for an ordinal classification problem.

- Vargas, V.M., Gutiérrez, P.A., **Rosati, R.**, Romeo L., Frontoni, E., & Hervás-Martínez, C. (2023). Deep learning based hierarchical classifier for weapon stock aesthetic quality control assessment, Computers in Industry, 144, 103786.

  As described in Section 4.6, the main contributions of the work are the following:

(i) the proposal of a DL hierarchical approach that simplifies, generalises and automatises the AQC task by using multiple ordinal CNN models to predict hierarchically the final label in two steps (one for the macro label and one for the micro);

(ii) the redesign and automation of an AQC task method in order to properly perform in the specific context, i.e. providing classification that could be more suitable for supporting the expert human operator.

- **Rosati, R.**, Romeo, L., Vargas, V. M., Gutiérrez, P. A., Hervás-Martínez, C., & Frontoni, E. Learning Ordinal-Hierarchical Constraints for Deep Learning Classifiers. IEEE Transactions on Neural Networks and Learning Systems. (under review)

  The main contributions of our proposed approach (described in Section 4.7) respect to related literature lie in:

  (i) the proposal of two novel hierarchical DL ordinal methodologies, namely Hierarchical Cumulative Link Model (HCLM) and Hierarchical Ordinal Binary Decomposition (HOBD) that are able to model the ordinal structure of different hierarchical levels of the labels within a single model;

  (ii) the testing of the effectiveness of the proposed methodologies on the AQC task, which disclose a natural hierarchical-ordinal structure of the classes, measuring the performance with respect to state of the art ordinal and hierarchical DL methodologies.

- Cecchini, G., Frontoni, E., Perugini, L., Romeo, L., **Rosati, R.**, Ruggeri, L., Tonetto, F., & Viti, P.. Italian Patent "Sistema di visione per il controllo qualità estetica basato su Intelligenza Artificiale" (pending).

  The DSS described in Section 4.8 has been patented as a technology transfer action between University and company Benelli Armi Spa.

Other publications released during doctoral studies, which are only partially related to the topic of the Doctorate and will not be discussed in the thesis, are listed in Appendix B.

Chapter 2.

State of the art

The state of art related to the Predictive Quality Control is reported in Section 2.1, while the literature review of the Aesthetic Quality Control with the state of the art of employed methodologies are treated in Section 2.2. For each section, the main contributions of this thesis with respect to research gaps are highlighted.

2.1. State of the art: Predictive Quality Control

Predictive Quality Control (PQC) is related to all the analysis that aim to establish baseline performance measures, monitor the performance of smart machines and compare with benchmark standards, and perform different actions accordingly [4]. In Quality 4.0 framework, the goal of PQC is to identify quality-enhancing insights from process and product data by using ML and DL methods in production [32], belonging to the more general field of Predictive Maintenance (PdM).

State of the art in this field poses different challenges related to the downtime and maintenance-related costs and different solutions for improving production efficiency. According to the literature review proposed in [30], the state of the art solutions to deal with this generic problem can be divided into four groups: (i) *knowledge-based*, (ii) *Big Data* analytics, (iii) *Machine Learning* models and (iv) *Ontology and reasoning*. Despite meta-heuristic optimization approaches [38, 39] are being introduced for solving a large scale combinatorial optimization problem related also to PdM tasks [40, 41], IoT sensing technologies together with the designing of a feature extraction stage and the deploying of a ML model ensure to directly learn from data for solving PdM tasks. This peculiarity allows monitoring, annotating, and consequently processing a large amount of heterogeneous data.

Supervised approaches

As predictive quality task involves the detection or prediction of quality and, frequently, machine RUL estimation, it mainly comprises methods of supervised learning. Supervised learning has the goal of estimating a numerical or categorical target variable on the basis of selected input variables (regression respectively classification)

[32]. In this context, one important limitation is the lack of data showing the annotation of anomaly state behavior [30, 42, 43]. In [44], the authors approached this problem by proposing a top-down strategy consisting of first understanding machine operation and then taking action to deal with the problem. In our case, we have built a data-driven model that can learn anomaly situations closely related to productivity losses, thanks also to a robust annotation. The model is able to generalize across different production cycles and operating conditions.

Deep Learning vs Machine Learning methods

For supervised learning, methods of both fields ML and DL can be used. The recent advances of technology and the huge amount of data have laid the foundations to apply DL methodologies for solving PdM task [45, 46]. The application of these solutions includes the implementation of Recurrent Neural Networks (RNNs) and Long Short Term Memory (LSTM) architectures [47, 48] for predicting failure by modeling spatio-temporal relationship across historical data. In [49] and [50], the artificial neural network model was employed for providing the fault prediction and identifying abnormal behaviors. Another work considers the implementation of auto-associative neural networks for finding irregularity in railways [51]. Considering our PQC task, sequential DL approaches (such as RNN and LSTM) may represent affordable predictive models by learning spatio-temporal features. However, the potential of DL approaches may be limited by the interpretability of the model [52], which does not always allow to provide clues on how and why the algorithm achieved the selected prediction. Moreover, knowledge-based approach is not considered for improving methods flexibility. Taking into account these considerations, we proposed a ML-based approach able to provide some insights about model interpretability.

Common ML approaches for solving PdM task include the application of standard classifiers such as K-Nearest Neighbors (KNN), Decision Tree (DT), Naive Bayes (NB), Support Vector Machine (SVM), Random Forest (RF), and XGBoost (XGB) [53]. From ML perspective, most related to our work are the following papers, which proposed the application of ML models for predicting machine errors or RUL and associated cost using sensor and event log data. In particular, the RUL prediction task was solved in [6] using log-based data and XGboost algorithm. An evolution of ensemble learning (i.e. graph based ensemble learning) was presented in [54] for modeling the behaviour of different subsystems using different base learners. Graphical models based on Bayesian network and dynamics Bayesian network were also proposed in [55] and [56] for learning causal relationships among features and across time in terms of conditional probabilities. In the latter, the graphical model is part of a framework designed to predict failures and to measure the impact of such a prediction on the quality of production planning processes and maintenance costs. Another class of ML models includes the application of auto-regressive models (e.g. Auto-regressive Moving Average) for predicting future behavior by using historical data.

In [43], the authors apply the auto-regressive integrated moving average model in a predictive maintenance framework to predict the remaining useful life of components.

The majority of these approaches for PdM use different condition monitoring data (e.g. vibrations, currents, temperature, etc.) and run to failure data for predicting the RUL. However, the annotation of the component wear is not always easily identifiable and traced across different production cycles and operating conditions. Thus, the open issues include the difficulty of obtaining quality labeled data and interpreting it [31]. A severe portion of available data could have no annotations, presents missing values, and is poorly structured. This fact leads to the high demand to have available a huge amount of annotated failure-related datasets. In our case, data are quantitatively annotated by a 3D coordinate measuring machine, resulting in a solid labeling process.

Model interpretability

Generally, state of the art work test different ML models to evaluate which one is more suitable for a given situation. For example in [53], the authors presented a classification model to predict failure using as input vibration limit value and used the accuracy of the models as an evaluation metric. Differently in [42] and [57], the computation cost for training the ML models is considered as an important aspect to take into account. In contrast with respect to the above-mentioned literature, we proposed a ML model that represents the best trade-off between the accuracy prediction, computation effort and model interpretability for optimizing the machining quality. Starting from the fact that manufacturing plants are dynamic environments, both the lack [58, 57] and the excess [59, 44] of data heterogeneity may negatively affect the ML model. Our approach tried to reduce the high dimension of collected data, by introducing a feature extraction stage based on hand-crafted features. Different from the knowledge-based approach, the extraction of salient Key Performance Indicators (KPIs) is fully automatic. KPIs are the set of metrics to reflect the performance of operations in terms of productivity, quality, and maintenance [60]. The KPIs monitoring allows to quantify and identify the aspects of the operational activities [61], proving useful not only in a predictive approach but also in the posterior step for decision-making process [62]. Besides the fact that this step enhances the interpretability and explainability of the data, experimental results demonstrated how it increase the generalization power of the ML model.

2.2. State of the art: Aesthetic Quality Control

Aesthetic Quality Control (AQC) refers to all those procedures for assessing the absence of defects that affect the aesthetic properties of the finished product. Current ML and DL for QC have been proven in a variety of industrial domains, including

fabric and textile industry [63, 64], printing industry [65], smart factory proptotype [66], laser-based additive manufacturing [67], welds mass production [68] and automotive industry [69]. In [63], the Fisher criterion-based stacked denoising autoencoders were applied to the problem of patterned fabric defect detection. They divided fabric images into patches of the same size to train the model. Afterward, test patches are classified into defective and defectless categories by also computing the residual between the reconstructed image and defective patch. The authors in [64], firstly employed a regression model to predict the operation parameters by using as inputs the yarn properties and secondly they applied a classification model to predict the quality of textile production by using as inputs the predicted operation parameters. In [69], different ML classifiers based on XGB and RF were employed to predict dimensional defects in a real automotive multistage assembly line. The line encompasses two automated inspection stages with several human-operated assemblies and pre-alignment stages in between.

2.2.1. Deep Learning approaches for Aesthetic Quality Control

Standard DL approaches were employed in [68] to replace costly quality control procedures based on visual inspection during the welds mass production scenario with the aim to improve the defect detection accuracy. They mainly focused on the collection of balanced database and image pre-processing. Deep Neural Network (DNN), Deep Belief Network (DBN) and restricted Boltzmann machine are standard DL architectures that were applied in [65] to perform a visual inspection process in the printing industry 4.0 by using as input a high-resolution optical quality control camera. Similarly, in [66] a standard deep learning strategy fed by images acquired by a camera placed over the assembly line was implemented to predict the quality-control in a smart factor prototype. Differently, the authors in [70] employed as predictors one key-quality index and different process parameters monitored by the control instruments. They applied a DNN consisting of a DBN in the bottom and a regression layer on the top to predict the quality prediction of a complex manufacturing process. Recently, a DL strategy was adopted for detecting geometric inaccuracy of the laser-based additive manufacturing process [67]. They combined the output of a Convolutional Neural Network (CNN) and the output of an Artificial Neural Network (ANN) for analyzing the thermal images and include relevant process/design parameters respectively. The overall network was trained to predict the final pointwise distortion prediction. Also in [11] the authors proposed a CNN solution to automatically extract discriminative features of the images for defect detection and at the same time by ensuring a high processing speed which guarantees real-time detection.

Nevertheless, all of these solutions focus on quantitative and deterministic analyses: the dimensional control, the roughness inspection of the materials, the patterned fabric

defect detection and the test of production parameters are all measurable evaluation procedures. Our work aimed to address a new challenge concerning the modeling of a qualitative analysis that is strictly human dependent, i.e. the aesthetic evaluation of a material.

The main differences with our work respect to related literature lie in the (i) different application of DL methodology we proposed in unexplored and challenging AQC application (i.e. we are interested to classify the aesthetic quality of rifle stocks based on the analysis of wood grains) and (ii) different goal we aimed to solve (i.e. the detection and mitigation of any unwanted bias in this scenario).

2.2.2. Ordinal classification

In the context of AQC task, the classes of the target variable often exhibit a natural ordering. However, the natural ordinal structure of the problem is not usually exploited and modeled in the learning procedure. For these reasons, the state of the art solutions include standard classification and regression models that do not completely solve the ordinal structure of the AQC task. This gap in the scientific literature lies the foundations to introduce a DL-based DSS driven by ordinal constraints for solving an AQC task.

Ordinal problems

Any given classification problem can be defined as the problem of predicting the real class y from an input data $\mathbf{x}$, where $\mathbf{x}$ is a K-dimensional vector $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^K$, and y is chosen from a set of labels $\mathcal{Y} = \{\mathcal{C}_0, \mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_{Q-1}\}$, being Q the total number of categories defined for the problem. Ordinal classification or ordinal regression [71] problems can be defined as a special case of standard classification problems, where an order constraint is included between the categories. In this way, for this kind of problems, the labels satisfy the expression $\mathcal{C}_0 \prec \mathcal{C}_1 \prec \mathcal{C}_2 \prec ... \prec \mathcal{C}_{Q-1}$. The precedence operator ($\prec$) indicates that the categories follow a natural order but, in contrast to a regression problem, they are discrete labels instead of continuous. Moreover, the distance between each category does not necessarily need to be the same. In these terms, we can define the position of each class in the ordinal scale as an integer like $\mathcal{O}(\mathcal{C}_q) = q$.

Thus, in any ordinal problem, the order information described can be accounted to obtain better classification performance, reducing the errors in distant classes while trying to maximise the number of patterns correctly classified. Also, examples that are misclassified in adjacent classes should produce a lower error when evaluating the classification performance of any ordinal model.

Recently ordinal classification (also called ordinal regression) methods have been proven useful in different research areas, including medical research [72, 73, 74], computer vision [75, 76, 77], finance application [78] and environmental management

[79]. An extensive review of ordinal classification approaches was provided in [80]. However, the introduction of these methodologies for solving an AQC task is not still explored in the ML literature. It is worth noting that ordinal classification approaches differentiate from the multipartite ranking problems where learning to rank strategy is applied to automatically construct a ranking model from training data [81, 82]. The multipartite ranking problem represents the state of the art in many information retrieval applications [83]. Although ordinal classification can be potentially scaled for solving a multipartite ranking problem, they are pointwise approaches for classifying data, where a naturalistic order is encoded in the label.

Ordinal classification problems can be easily simplified into other standard problems using the round prediction of a regression model or a cost-sensitive penalty. These are considered standard approaches for solving the ordinal classification task, with the main limitation that they assume a distance between class labels which can influence the performance of the classifier. A specific method based on a cost-sensitive ordinal hyperplanes ranking algorithm has been used for human age estimation using face images [76]. The authors designed the cost of an individual binary classifier so that the misranking cost can be bounded by the total misclassification costs. Other ordinal approaches include ensemble decision tree and random forest models [73, 84] based on a weighted entropy function for selecting the predictors in the tree that reflect the magnitude of potential classification errors. A different approach based on conditional ordinal random field model was proposed for context-sensitive modeling of the facial action unit intensity by answering the context question in terms of temporal correlation between the ordinal outputs [85]. Recent state of the art works move towards two different approaches: ordinal binary decomposition methods and threshold models combining ordinal loss functions.

Ordinal binary decomposition approach

One of the main class of ordinal-based approaches is the Ordinal Binary Decomposition (OBD) strategy. Within this category, the multiple model approaches use several binary classification branches to compute a series of cumulative probabilities. Although this approach introduces a large number of hyperparameters to be tuned, there are some work [74] that try to reduce the effect of this problem, by redesigning the output layer of the conventional deep neural network. Moreover, in the ordinal decomposition approaches, the relationships among different binary classifiers are often neglected. To try to alleviate this issue, it was proposed to learn an ordinal distribution of the problem and to optimize those binary classifiers simultaneously [86]. Similarly, a multiple ordinal regression algorithm to estimate the preferences of humans was proposed [87]. They maximized the sum of the margins between every consecutive class with respect to one or more rankings (e.g., perceived length and weight). An ordinal decomposition approach combined with a fully 3D CNN network was used for assessing the level of neurological damage in Parkinson's disease patients and exploring the

potential classification performance improvement in using ordinal label information [72]. A standard sigmoid function is provided in the output node, rather than using a softmax function for the output nodes. They trained a single convolutional model for solving simultaneously individual binary classification tasks, which were treated as multiple fully connected blocks.

Threshold based models

The most natural strategy to handle the ordinal structure extends the standard regression task by assuming that a latent variable underlies the ordinal classes. In this general approach, called the threshold model, both the latent variable and the thresholds, which act respectively as a mapping function and ordinal constraints, need to be learned from the data. A threshold-based loss function is designed to model the ordinal values among multiple output variables [88]. The authors applied the kernel trick to provide a nonlinear extension of the model. Another work presented a structural distance metric for video-based face recognition [77]. Here the ordinal problem is designed as a non-convex integer program problem that firstly learns stable ordinal filters by projecting video data into a large-marginal ordinal space and then self-corrected the projected data in a structure low-rank strategy. A large margin ordinal regression formulation was also provided as a feature selection strategy for detecting minimum and maximum feature relevance bounds by inducing sparsity in the model [89]. The authors in [90] proposed the introduction of the lp-norm for deriving the ordinal threshold with class centers with the aim to alleviate the influence of outliers (i.e. non-i.i.d. noises). Their approach provided an optimization algorithm and corresponding convergence analysis for computing the lp-centroid. In [91], two neural network threshold ensemble models were proposed for ordinal regression problems. They generated a different formulation of the learned threshold by generating different projections for the parameter updating. Another approach consists in imposing the ordinal constraints on the weights that connect the hidden layer with the output layer [92]. The formulation allows determining the optimum ones analytically according to the closed-form solution of the inequality constrained least-squares problem estimated from the Karush-Kuhn-Tucker conditions. In [93] is proposed a deep CNN model for ordinal regression by considering a family of probabilistic ordinal link functions in the output layer. These ordinal link functions fall within Cumulative Link Model (CLM). They split the ordinal space into the different classes of the problem by using a set of ordered thresholds. The thresholds are learned during the training process by minimizing a loss function that takes into account the distance between the categories, based on the weighted Kappa index.

Limitation of state of the art

Similar to the regression model, the main problem of standard ordinal classification approaches based on regression is the lack of a direct relationship between the prediction error of the regression model and the misclassification error. A different problem arises for the cost-sensitive penalty approach where there is the need to have a priori knowledge of the task in order to properly define the cost matrix. Accordingly, the ordinal binary decomposition approaches are highly influenced by how the overall problem is decomposed and how the results of all decompositions are aggregated into a single final classification. Some recent work in literature tried to overcome these problems by learning a single model for solving simultaneously individual binary classification tasks. However, these methodologies only model a static relationship among the ordinal classes that originate on how the problem is decomposed in binary subtasks. The threshold-based models proposed in literature often require multiple hyperparameters for setting the ordinal probability thresholds. Indeed, most of the state of the art threshold-based approaches require highly demanding optimization procedures, which do not always guarantee optimal convergence and robustness against outliers.

The most related work to our proposal is the paper [93] that introduced the CLMs and quadratic weight kappa for solving an ordinal problem. The main differences with our work lie in the (i) loss function we adopted, (ii) the different hyperparameters (i.e. slope) we learned in the learning process, (iii) a different unexplored task we aim to solve (AQC task) and (iv) the multiple objectives we aim to achieve, i.e. both an increase in generalization performance and also mitigation of unwanted bias related to the geometry. Indeed, we solved the ordinal problem by modeling the cumulative distribution of the AQC classes through the hyperparameters we learn in the CLM. Moreover, in our work, we exploited the standard Categorical Cross-Entropy (CCE) loss for solving the ordinal AQC problem. As we shall see in Section 4.4, our deep ordinal model performs favorably over the CLMs for deep ordinal classification in [93].

2.2.3. Regularization techniques for ordinal learning

Regularization techniques, such as label smoothing, are used for enhancing the robustness of a model in presence of noisy labels: during training procedure, they encourages the classifier to be less confident, giving some probability to the other classes instead of focusing only on the true category. This is useful for ordinal problems, where misclassifying a pattern in an adjacent class is more probable than predicting a distant category. In this context, respect to the state of the art we proposed a more flexible exponential function based on the introduction of the L_p norms. L_p norms have been used in optimisation algorithms in several fields as a generalisation of L_2 and L_1 norms, including binary classification [94], feature selection [95] or generative

adversarial networks [96], among others. Depending on the value of p, the objective varies. In [97], the authors proved that L_2 norm methods tend to expand or bleed out over natural boundaries. Therefore, using a L_p norm where $1 < p < 2$ should provide a more suitable alternative when it is optimised properly. The use of this type of generalised norms has drawn a huge attention in multiple applications, such as 3D medical image super-resolution [98].

More specifically, multiple works have discussed the potential advantages of the alternatives to the L_2 or L_1 norm. In [99], the authors presented an L_p norm alternative to Least Squares Support Vector Machine (LSSVM) [100]. In [101], a new method was proposed achieving robustness by replacing the L_2 norm in conventional linear discriminant analysis by L_p norm in within-class distances and by L_s norm in between-class distances. However, for several of these tasks, the L_p norms have raised a huge attention. Bregman divergences is one of the standard tools for analysing online machine learning algorithms [102], allowing a generalisation of the least mean squared algorithm. In this sense, the loss bounds for these L_p norm algorithms involve others than the standard L_2 or L_1 norms [94].

2.2.4. Hierarchical classification

Even though the methods described in 2.2.2 were proved to improve the performance of ordinal classification problems, none of them included the possibility of having a hierarchical structure implicit in the categories of the problem. When dealing with complex problems that are hierarchically labelled, using a method that exploits these hierarchical structures can lead to better classification performance and lower cost errors.

In the ML literature, hierarchical classification problems are usually addressed using a hierarchical Multi-label classification (HMC) approach. In HMC every instance may belong to multiple classes simultaneously and these classes are arranged in a hierarchical structure, implying that an example associated with a particular subclass belongs at the same time to all the superclasses in the hierarchy. The state-of-the-art approaches include traditional ML models [103] and neural networks architectures [104, 105, 106], in which the hierarchical structure is explored in different ways.

The most naive approach (global classifier) is to employ a single classifier for modeling the entire class hierarchy, where the objective is to predict the classes associated with the leaves of the hierarchy, without considering upper levels. However, a major limitation of this approach is that it completely ignores the class relationships and any hierarchical constraints, while typically predicting only the leaf nodes (flat classification).

Multi-stages approach

A different approach (local classifiers) is to employ a set of classifiers for each node or each parent level, as our first proposal describe in Section 4.6. Thus, each classifier is specialized in solving the classification task associated with the child nodes. In these terms, the Error Correcting Output Codes (ECOC) approach [107] was proposed as a method to decompose a multi-class problem into multiple binary problems that can be solved independently using different models. Then, the predictions of each model are combined like in any ensemble. This method has been used in several previous works [108] in combination with CNN models. Although this approach is not originally hierarchical, a hierarchical approach can be easily derived from it, given that the codes generated for each of the labels can include the hierarchical dependencies of those classes. To do that, the codes can be simply composed of Q_i bits for each of the hierarchical levels, where Q_i is the number of classes in level i. The bits corresponding to the correct label on each of the levels will be active while the others will remain zero. In this way, the hierarchical structure is encoded in the generated codes. However, this approach does not take into account the ordinal information, given that each of the bits is going to be one or zero without taking into account the rest of the bits. Therefore, to address this problem, in [109], the authors proposed a method to generate the ECOC codes in an ordinal way, resulting in better classification performance for ordinal problems. However, in this case, the hierarchical structure of the labels is not represented by the codes. Also, it is worth noting that the ECOC approach usually will spend more time for the training process, given that they decompose the original multi-class problem in multiple binary problems, and each of them is trained using all the training samples. Taking into account the characteristics of the approaches described, in our work, we proposed to combine the described ordinal methods with a new hierarchical classification approach that aims to predict the correct label for an ordinal problem in two separate steps.

One-stage approach

However, the multi-stage approaches require a greater computation effort for learning separated multiple models [110]. More related to our final proposal, described in Section 4.7, the HMC approach in [111], which also leverages a single hierarchical multi-label neural network architecture (HMCN) capable of simultaneously optimizing their local and global loss functions to model the hierarchical structure of the classification task while penalizing hierarchical violations. The benefit of this strategy is to decompose local and global classes that may potentially represent labels of different nature. Different from their definition of global classes (which include all the classes in the hierarchy), our global loss takes into account only the leaf nodes of the proposed hierarchical problem, i.e. the classes associated with the final classification problem we want to solve. Moreover, we consider the possibility of learning ordinal

relationships among labels. Their formulation addresses two main concerns. Firstly, the HMC approach proposed in [111] requires the execution of a post-processing step to ensure that all predictions respect the hierarchical path, penalizing predictions with hierarchical violations during training phase. This is not necessary in our approach, because our method naturally modeled only the admissible paths (i.e. it is not possible to obtain inconsistent labels for the global predictions). Accordingly, our formulation allows to encode ordinal constraints within both global and local losses, assuming that the ordinal dependencies of global loss can be different with respect to local losses. This point takes into account the natural setting of this task. Furthermore, our formulation further leads to model different ordinal relationship among different local losses that can be associated to different hierarchical levels.

Chapter 3.

Predictive Quality Control

In this chapter, ML methodologies, experimental procedure and results about the PQC problem are described. Firstly, in Section 3.1 a preliminary work is presented about the design of a DSS platform for RUL estimation of Automated Teller Machines (ATMs). This work was performed to assess whether a state of the art ML method, i.e. Random Forest (RF) model, could be suitable for addressing the challenges related to a PdM task. According to this approach, in Section 3.2 the RF methodology was applied to solve the PQC task related to the prediction of machine processing quality (see Fig.3.1). Each section describes as first the real use case we aimed to face, then presents the proposed ML-based framework and then the experimental setup and the results obtained.

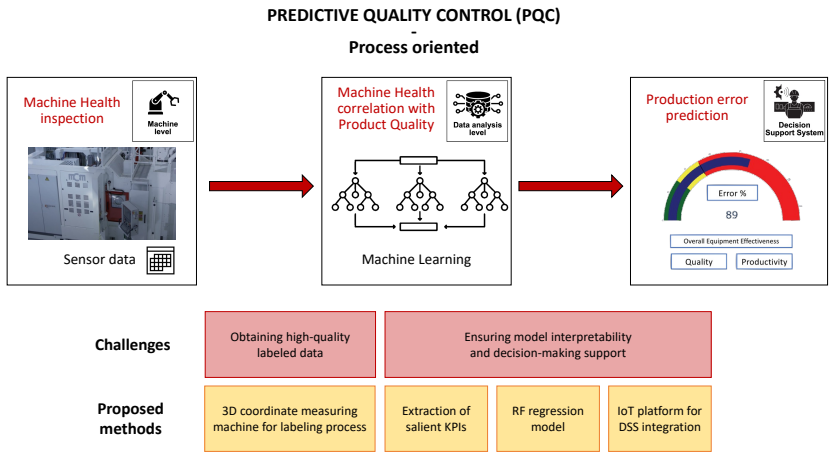


Figure 3.1.: Real-world industrial PQC problem: prediction of production error in MCM machines. PQC is designed according to the Quality 4.0 framework shown in Fig. 1.1. Related challenges and proposed methods for solving them are outlined.

3.1. Interpretable Machine Learning approach for RUL estimation

In recent years, in the context of Industry 4.0 and intelligent manufacturing, there has been increasing emphasis on the Predictive Maintenance (PdM) task [112, 113, 114], as stated in Section 1.1. PdM aims to estimate when a machine might fail in order to schedule corrective maintenance operations before the point of failure. Data-driven algorithms for PdM imitate the normal data behaviour of a machine and use it as a baseline to identify and report deviations in real-time. A machine monitoring system includes input data (time-series) on a range of factors, e.g. from temperature to pressure. The output is the desired target, i.e. a warning of a future failure or the remaining useful life (RUL) of the tool or machinery [115]. The algorithm will then be able to predict when a failure is likely to occur or estimate the life time of the machine. In the literature, the two main machine learning (ML) strategies to solve these tasks are supervised and unsupervised methodologies. The two categories of approaches may be relevant for a different scenario and depend on the availability of sufficient historical training data and the frequency of equipment failure [116].

In this context, the "Smart Manufacturing Machine with Predictive Lifetime Electronic maintenance" (SIMPLE) project was promoted, involving various companies and universities in the Marche region (Italy). The aim of the project is to create innovative products that can be monitored and controlled both locally and remotely, able to implement PdM logics, connected to a new flexible platform. One of the topic of SIMPLE project is related to the prediction of RUL of ATMs manufactured by Sigma company. This represents a key task as these machines are subject to different types of failures, which are difficult to predict by maintenance staff. From a practical perspective, identifying when the next failure might occur is a relevant aspect for significantly reducing maintenance costs and avoiding lack of service delivery for long periods of time. However the main challenges in this field lie in: i) collecting a representative dataset, ii) correctly annotating the observations, iii) handling the imbalanced nature of the dataset and iv) providing a model that simultaneously ensures high performance of accuracy and interpretability.

3.1.1. Data collection

ATM devices generate a very large amount of logs associated to each individual machine. These files can contain all the information about the operation of individual devices and are normally written in verbose mode to allow the analysis of any abnormal behaviour. However, they are generally not required to be structured according to strict rules. In the case of the SIMPLE project, the logging process was rationalised by implementing an original log management solution with aggregation and storage features, secure transfer, automatic parsing and transformation of logs into data. In

addition, the data of the technical interventions was extracted, both as a result of calls for malfunctions and linked to preventive maintenance activities, from Sigma's ticket management system. The information coming from the logs and the maintenance server was then reconciled by means of automatic operations (batches specially developed to correlate the data by machines/dates) and manual operations (for labeling the types of faults found after interpreting the unstructured notes of the technician at the time of the intervention).

Table 3.1.: Characteristics of the ATMs dataset.

	Machines	**# Observations**
Total	89	15254
	Failures x machine	**# Observations x cycle of fault**
Max	8	533
Min	0	1
Mean	2	82
Std	1.8	91

The collected dataset represents a total of 89 different machines, and it is built on raw ATM logs which contain information about several sub-devices. In particular, attention was focused on three different devices installed in the ATM: Cash Recycling Module (CRM), badge reader and receipt printer. The CRM is the most complex device within the ATM as it consists of many mechanical parts that must be driven and maintained in the best possible way to avoid problems when handling money. The dataset resulting from the data extracted through the parsing of the logs is composed of 236 features and a target variable corresponding to the information on the failure event. A failure can be caused by a wide variety of factors, making the prediction task very difficult: it may be the result of a component degradation or a sudden and unpredictable event (such as a jam of a crumpled banknote). Input features are data about opening and closing intervals of the device shutter, number of processed banknotes, number and types of movements in each zone of the device and associated time to complete, number and type of low level error code occurred, information about standard maintenance execution (cleaning of parts or their substitution), labels about abnormal status of single parts observed during maintenance interventions, etc. The high number of features is a consequence of the complexity of the CRM device, which must be able to dispense banknotes, both single and bundled, from the ATMs, and therefore it requires extremely sophisticated mechanics, control/implementation sensors and management/monitoring software. The extracted features are aggregations on a daily basis of this data, while the labels are manually interpreted by the tickets of technical interventions. Statistics about dataset distribution are collected in Table 3.1. The final dataset presents the following challenges:

- the data corresponds to time-series belonging to different machines with an ex-

tremely not homogeneous number of failures;

- there are a large number of variables with unknown correlation within the various types of faults;
- there are a limited number of failure cases;
- there is a possible data leakage issue in dataset preparation, since features are pre-aggregated and some information may be lost on generalization.

3.1.2. Machine Learning pipeline

Task definition

The problem of RUL estimation can be associated both with regression and classification tasks. In a regression approach, RUL is maintained as a continuous value in order to predict the exact remaining time before failure. In the context of ATMs maintenance, estimating the exact moment of failure is a very complex task according to company expertise; for this reason, it was considered more appropriate to treat the problem as a classification task. For the classification approach, the RUL value is converted into a discrete value to predict if the machine will fail within a certain time frame. Therefore two different supervised methodologies were evaluated for solving the task:

(a) Binary classification: we convert the RUL in a binary value (class 1 high-risk of failure, class 0 low-risk of failure). Label is considered as class 1 according to a time window of 6 days before failure;

(b) Multi-class classification: RUL is converted into 5 classes to predict machine failures in different time windows. In our experiments, we consider the following label encoding:
    - very high risk (class 5) : number of days before failure between 1 and 3;
    - high risk (class 4) : number of days before failure between 4 and 6;
    - medium risk (class 3) : number of days before failure between 7 and 9;
    - low risk (class 2) : number of days before failure between 10 and 14;
    - very low risk (class 1) : number of days before failure between 15 and 19.

One of the main concerns approaching the problem as a classification task is related to the huge imbalance distribution of classes. For example, as regards the binary approach, considering the fault class as only the day 0 of RUL implies a small number of failures (and, from a practical perspective, no time to model and to prevent failures); on the other side, considering the fault class as several n days before the day 0 of RUL implies a high risk of training ML model on data which are not enough discriminative of failures (ie. data that are too far from failures).

Pre-processing, feature extraction and selection

The domain experts suggested the most discriminative features that can be monitored in order to discriminate the RUL. These features represent daily statistical measures (i.e. max, min, std, mean, median, 1-st quartile, 3-rd quartile) derived from the original raw data. To delete redundant features, a standard correlation analysis was applied using a threshold of Pearson correlation value equals to 0.5 (with p-value < 0.05), which removed 135 features (Figure 3.2a). Most of the removed features regard the type of error counters and some of the quartiles about the dispense preparation time and deposit time of banknotes . Then, feature importance by embedded methods highlighted the most relevant features, as shown in Figure 3.2b.

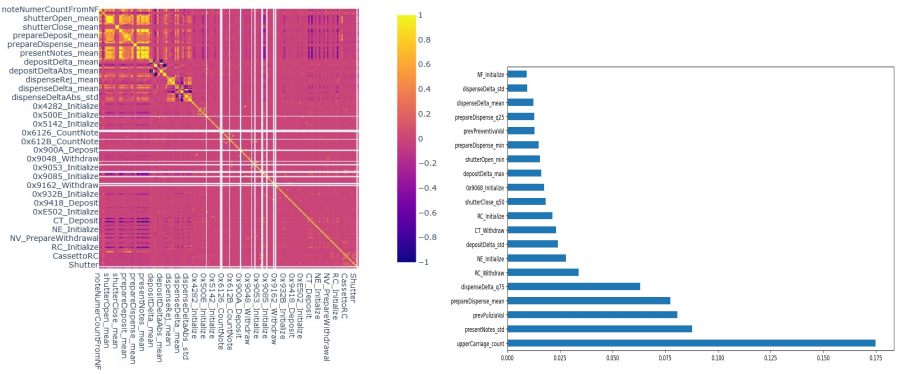


Figure 3.2.: Feature extraction and selection procedure: a) Pearson correlation analysis; b) a subset of the most important features.

Pre-processing techniques such as normalization and outlier removal were applied together with oversampling/undersampling strategies to face the imbalance issue in the classification task [117]. In particular, the following strategies were selected:

- SMOTE [118] to increase the number of faults. The number of faults samples generated was the one needed to match the number of non-fault samples;
- Random undersampling [119] to decrease the number of non-fault samples (random selection of 5000 samples);
- SMOTE + Random undersampling [120]: for increasing and decreasing both classes to the same number of samples.

Predictive model

The selected ML algorithm to solve both binary and multi-class classification tasks is the Random Forest (RF) model. RF represents a variant of bagging proposed by [121]

and consists of an ensemble of decision trees (DTs) generated by independent identically distributed random vectors. RF is modeled by sampling from the observations, from the features (i.e., n° of features to be selected) and by changing tree-parameters. The idea behind this sampling is to maximize the diversity among trees, by sampling from the features set and from the data set as well. In particular, the random feature selection is carried out at each node of the tree, by choosing the best feature to split within a subset of the original feature set. In classification problems, an ensemble of DTs is built, which aims to split the data into subsets that contain instances with similar values (homogeneous). For each subset, a random feature selection is carried out at each node of the tree. In addition to predictive performance, we have also to take into account another important factor for solving a PdM task, which is model interpretability. This requirement is satisfied by the RF algorithm, which allows providing a direct interpretation of the most discriminative features.

3.1.3. Performance evaluation and Results

Experimental procedure

The RF-based PdM approach was compared with respect to other state-of-the-art ML approaches employed for solving PdM tasks. In particular, the following models have been considered for comparing both binary and multi-class tasks:

- Linear Regression (LR) [122];
- K-Nearest Neighbors (KNN) [123];
- Decision Tree (DT) classifier;
- Support Vector Machine (SVM) with Gaussian and linear kernel [124];
- Support Vector Machine (SVM) with Elastic Net penalty [124];
- Gaussian Naive Bayes (NB);
- XGBoost (XGB) [6].

All the ML models have been tested only with the preprocessed dataset. A 10-fold Cross Validation (CV) procedure stratified over machines and class values was performed in order to evaluate the performance of the RF model. Although this experimental procedure is computationally demanding, it ensures measuring the ability of the proposed algorithm to predict RUL across unseen machines. For each algorithm, the related hyperparameters were optimized by implementing a grid search in a nested 5-CV. Table 3.2 shows the different hyperparameters for the proposed ML models and all competitors' ML approaches, as well as the grid-search set. For LR, linear and gaussian SVM the λ penalty controls the 2-norm regularization. For the SVM Elastic Net $\alpha = \lambda_1 + \lambda_2$ and l1_ratio=$\frac{\lambda_1}{\lambda_1+\lambda_2}$ where λ_1 and λ_2 control separately the 1-norm

Table 3.2.: Range of hyperparameters (Hyp) for the proposed RF model and the other state-of-the-art ML approaches.

Model	Hyp	Range
RF	n° of classification trees	$\{100, 200\}$
	n° of features to select	$\{2, 5, 10\}$
	max depth	$\{25, 50, 100\}$
LR	λ	$\{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}\}$
KNN	n° of neighbors	$\{3, 5, 7, 9\}$
	weight function	{'uniform', 'distance'}
	distance metric	{'euclidean', 'manhattan'}
DT	split criterion	{'gini impurity', 'entropy'}
	max depth	$\{50, 100\}$
	min n° of leaf size	$\{1, 2, 3, 4, 5\}$
SVM_gaussian	λ	$\{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}\}$
SVM_elasticnet	α	$\{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}\}$
	l1_ratio	$\{0, 0.25, 0.5, 0.75, 1\}$
SVM_linear	λ	$\{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}\}$
NB	variance smoothing	$\{10^{-9}, 10^{-8}, 10^{-7}\}$
XGBoost	learning rate	$\{10^{-3}, 10^{-2}, 10^{-1}\}$
	max n° of estimators	$\{50, 100, 200\}$
	max depth	$\{50, 100\}$
	n° of features to select	$\{1, 2, 4\}$

and 2-norm regularizations. The main metrics selected for tuning model hyperparameters and evaluating the final models are the F1 score and the balanced accuracy (defined as the average of recall obtained on each class) [125], since we consider both True Positives and False Positives decisive factors for our task.

Evaluation metrics

The assessment of the ML classification tasks described in Section 3.1.2 was performed according to the following standard metrics:

- Correct Classification Rate (CCR), also known as *accuracy*, which indicates the percentage of correctly classified samples

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}; \tag{3.1}$$

- Precision, also called *positive predictive value*

$$Precision = \frac{TP}{TP + FP}; \tag{3.2}$$

- Recall, also known as *sensitivity*

$$Recall = \frac{TP}{TP + FN}; \tag{3.3}$$

- F1_Score

$$F1\_Score = 2 * \frac{Precision * Recall}{Precision + Recall}; \tag{3.4}$$

- Balanced accuracy [125]

$$Bal\_Accuracy = \frac{1}{C} \sum_{i=1}^{C} Recall_i; \tag{3.5}$$

- Receiver Operating Characteristic (ROC), which summarizes the trade-off between the true positive rates (TPR) and the false-positive rates (FPR) for a predictive binary classifier.

where TP is True Positive, FP is False Positive, TN is True Negative, FN is False Negative and C is the number of classes.

Classification performances

The predictive performance of the RF model is shown in Table 3.3 for binary task and Table 3.4 for multi-class approach on the ATMs dataset. The values shown in tables are the mean across the respective metric resulting from the single CV folds. For the binary classification, it can be noted that the best prediction results were obtained by our RF in all metrics except recall. This trend is also confirmed by the best Area Under Area under the ROC Curve (AUC) value achieved by RF (see Figure 3.3). Low recall values denote that the classifier presents a high number of false negatives, which may be due to the imbalanced distribution of the dataset. This is confirmed also by the F1-score, which is below 0.5. As regards the multi-class classification, it is possible to note how the model's performance significantly decreases. However, RF achieved the best results for F1 score and balanced accuracy, which are the most relevant metrics for our PdM task.

3.1.4. SIMPLE DSS for Predictive Maintenance tasks

The designed platform is composed of two server groups, each of which is composed of three servers and a Storage Area Network (SAN). The solution includes a hardware (HW) infrastructure consisting of two distinct server and storage groups located in two different data centers, for ensuring the level of reliability required at the infrastructure level. The software technical specifications of the SIMPLE project led to the design of

Table 3.3.: Predictive performance of ML approaches for the binary approach.

Model	Accuracy	Precision	Recall	F1	BalAccuracy	MAE
LR	0.933	0.652	0.211	0.319	0.601	0.067
KNN	0.934	0.679	0.208	0.317	0.600	0.066
DT	0.825	0.183	**0.381**	0.244	0.618	0.175
SVM_gaussian	0.933	0.749	0.158	0.259	0.577	0.067
SVM_elasticnet	0.925	0.659	0.179	0.279	0.562	0.069
SVM_linear	0.927	0.535	0.216	0.305	0.601	0.073
NB	0.928	0.126	0.380	0.185	0.543	0.702
XGB	0.932	0.639	0.256	0.363	0.622	0.068
RF	**0.938**	**0.766**	0.243	**0.368**	**0.621**	**0.062**

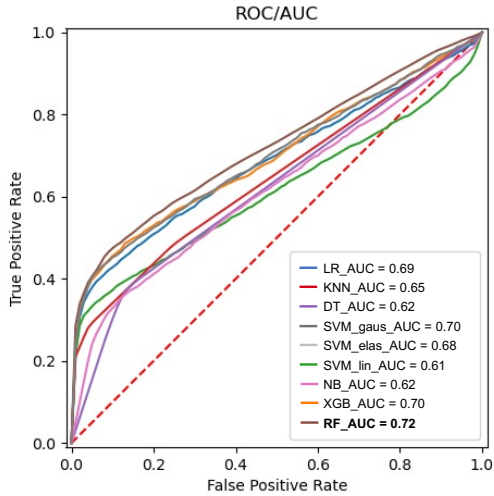


Figure 3.3.: ROC curves of ML methods for the binary approach.

the system architectural scheme illustrated in Figure 3.4. The platforms assure scalability and interactions with different PdM tasks and different companies enrolled in the SIMPLE project. Each task has their own constraints, characteristics and requirements, which have guided the technical choices on the communication protocol with the devices towards a solution that should be as generic as possible, involving the support of at least two protocols (MQTT and REST). A container-based deployment technology was the basic architectural solution employed, in order to quickly manage the scalability and the resources of containerised applications, check the integrity status of applications and manage corrections with automatic placement, restart, replication

Table 3.4.: Predictive performance of ML approaches for the multi-class approach. In this case, binary metrics (precision, recall, F1) are averaged across classes.

Model	Accuracy	Precision	Recall	F1	BalAccuracy	MAE
LR	0.298	0.462	0.262	0.222	0.262	1.272
KNN	0.267	0.281	0.254	**0.256**	0.254	1.394
DT	0.297	**0.518**	0.258	0.173	0.258	1.151
SVM_gaussian	**0.304**	0.379	0.156	0.174	0.265	**1.132**
SVM_elasticnet	0.262	0.346	0.257	0.215	0.257	1.431
SVM_linear	0.302	0.317	**0.272**	0.247	0.262	1.294
NB	0.214	0.310	0.255	0.159	0.255	1.183
XGB	0.273	0.266	0.261	0.255	0.261	1.402
RF	0.291	0.294	0.266	**0.256**	**0.266**	1.343

and scaling. For the orchestration and monitoring of each functional module identified (i.e. Nginx, VerneMQ, Cassandra...), the installation and configuration of the OKD (Kubernetes Container) platform on the identified HW infrastructure has been tackled. From this basis, the software solutions considered to be the most promising has been refined also according to the ML computing capacity required.

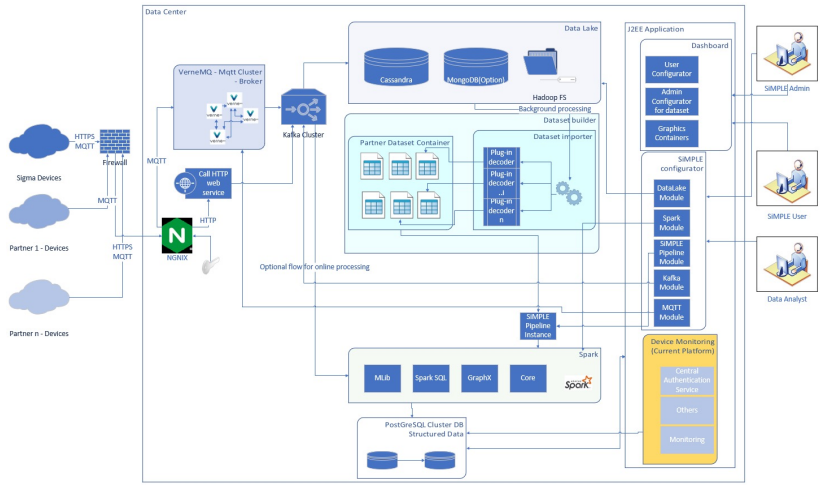


Figure 3.4.: System architecture diagram of predictive maintenance platform within SIMPLE project.

3.2. DSS for the prediction of processing quality

The proposed Predictive Quality Control (PQC) task originated from a specific company demand: the prediction of processing quality and anomaly situations during the machining of a tool. According to the prediction of non-compliant processing during production phase, the aim is to take preventive actions so that errors do not propagate along the production chain. For this reason, this task falls within the PdM paradigm.

The company Benelli Armi Spa is an Italian firearms manufacturer specialized in the production of semiautomatic sport rifles, producing more than $200k$ weapons per year and exporting to 78 countries around the world. Thanks to the latest manufacturing technologies, Benelli produces weapons that combine excellent ballistic performance and superb functional qualities. In recent years, the company has embraced the typical Industry 4.0 framework that includes a strong industrial structure as well as advanced technology to ensure high standards of production in each component of the final product. Among all the various manufacturing processes of the factory, the QC phase is a fundamental step in the production of a rifle as the finished product must guarantee high performances both at mechanical and aesthetic level.

In particular, the defined PQC task, represented in Fig.3.1, aims to predict production errors during the machining of MCM systems. These machines perform various manufacturing processes (e.g. drilling and sanding) for the mechanical properties of the products. Predicting the processing quality i) helps to prevent defective parts from entering the next stage of assembly and slowing down the production cycle, ii) allows to implement PdM actions to replace exhausted tools.

3.2.1. Data collection

The input parameters were represented by topic at level 0 (L0), i.e. processing parameters (e.g. acceleration, speed, position) collected from two different machine centers (see Figure 3.5a). The condition monitoring data, that are the annotation used for the supervised model training, were represented by topic at level 2 (L2) acquired by a robotic part loading system for coordinate measuring machine (see Figure 3.5b). The overall flexible integrated manufacturing system includes two operator loading/unloading stations, two robot loading/unloading stations, one automated vertical parts store and one parts washing unit.

The L0 processing parameters were acquired by two different machining centers (mc) (i.e. MCM[1] clock 5-axis machining centers). On a 5-axis mc, the cutting tool moves across the X, Y and Z linear axes and rotates on the A and B axes to approach the workpiece from any direction. The mc can be configured for multitasking operations, such as milling, turning, grinding, boring, etc. Moreover, all mc can be configured with a single pallet, pallet exchanger, multi-pallet systems or integrated in

[1] https://www.mcmspa.it

(a) Machine center (MCM clock 5-axis machining center)

(b) Hexagon 3D coordinate measuring machine

Figure 3.5.: Experimental setup in the real industrial use case: advanced processing (a) and measuring (b) machines.

a Flexible Manufacturing System (FMS). The level of automation can be changed or increased during the service life of the plant, providing considerable flexibility.

The L2 condition monitoring data were acquired by a robotic part loading system for Coordinate measuring machine (CMM) (i.e. Hexagon[2] Manufacturing Intelligence Robotic CMM part loading). The CMM system automatically identifies in real-time parts that are out of tolerance and triggers alarm situations. The CMM can be easily used by operators with minimal training as a cost-effective automated part loading system that increases the throughput of CMM and maximizes operational capacity.

[2]https://www.hexagonmi.com

3.2.2. Proposed DSS Framework

The proposed ML-based strategy is conceived to solve the above-mentioned PQC task by allowing the continuous collection of an annotated dataset and the provision of a data analytic interface for supporting the maintainer/operator. Figure 3.6 describes the overall architecture of the proposed DSS, which is comprised of five IoT and ML cornerstones:

- Data collection: the IoT sensing technology is based on the Message Queuing Telemetry Transport (MQTT) broker [3] [126]. The central concept in MQTT dispatcher is topics that collect processing parameters (e.g. acceleration, speed, position) of the machining centers at the lower level L0. Accordingly, the status and conditioning data collected by the Hexagon machine represents the topic at the higher level L2. All data are synchronized and are collected in a SQL database and then in Azure Blob cloud storage.

- Feature extraction: the Trapezoidal Numerical Integration (TNI) is performed to compute a Key Performance Indicator (KPI) for each processing parameter during each MCM production cycle.

- Predictive model: a RF regression model is applied in order to estimate the status and conditioning data using the collected processing parameters as predictors.

- Cloud architecture: the ML model is deployed as a docker container with an API endpoint for testing unseen acquired data. Azure ML service is used for providing a cloud-based environment for deploying and updating ML model. Azure Blob storage is adopted to store the ML model weights and the ML model outcomes (predicted error %).

- Data analytics: the ML model outcomes are displayed in a GUI-based data analytic interface for supporting the maintainer/operator.

Data collection

The central sensing and communication point is the MQTT broker (Mosquito), which is in charge of dispatching all messages between the senders and the rightful receivers. The implementation of MQTT protocol in JFMX was performed by using the Paho Java library. Each client that published a message to the broker, includes a topic into the message, that represents the routing information for the broker. Each client that

[3] MQTT is a lightweight telemetry protocol, coming from the world of M2M and now widely applied in IoT. The central communication point is the MQTT broker, which is in charge of dispatching all messages between the senders and the rightful receivers.

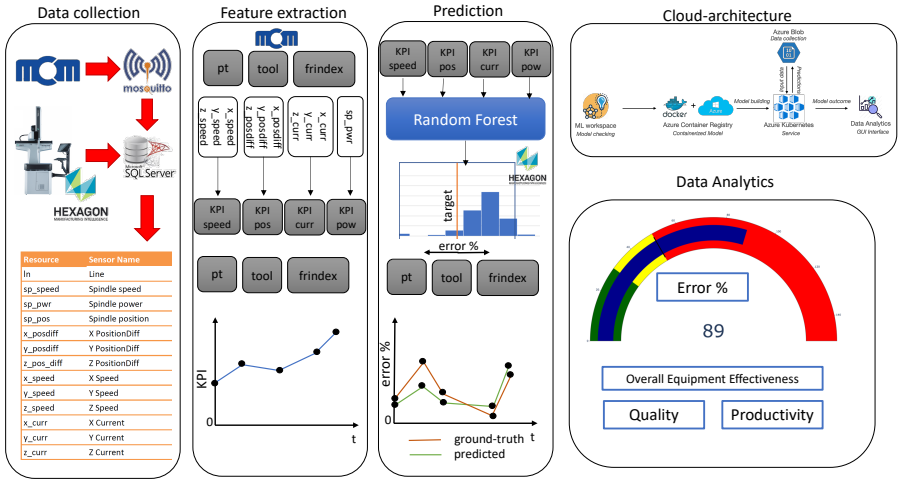


Figure 3.6.: Flow-chart of the proposed approach: data collection, feature extraction, prediction phase based on Random Forest predictive model, cloud storage and data analytics.

wants to receive messages subscribes to a certain topic and the broker delivers all messages with the matching topic to the client. Therefore the clients don't have to know each other, they only communicate over the topic. This architecture enables highly scalable solutions without dependencies between the data producers and the data consumers. The payload of messages are just a sequence of bytes, up to 256Mb, with no requirements placed on their format, and with MQTT protocol usually adding a fixed two bytes header to most messages. Other clients can subscribe to these messages and get updated by the broker when new messages arrive.

The central concept in MQTT to dispatch messages are topics. A topic is a simple string that can have more hierarchy levels. For example, a topic for sending status data of mc2 of an FMS is the following: JFMX/L1/fms/UNIT/mc1/STATUS. On one hand, the client can subscribe to the exact topic or on the other hand use a wildcard. The wildcard (+) allows arbitrary values for one hierarchy while the multilevel wildcard (#) allows to subscribe to more than one level (e.g. the entire subtree). The MQTT topic organization allows to guarantee the Quality of Service (QoS). The MQTT protocol handles retransmission and ensures the delivery of the message, regardless how unreliable the underlying transport is. In addition, the client is able to choose the QoS level depending on its network reliability and application logic:

- **QoS 0 – at most once:** it guarantees a best effort delivery. A message won't be acknowledged by the receiver or stored and redelivered by the sender. This is often called "fire and forget" and provides the same guarantee as the underlying TCP protocol.

- **QoS 1 – at least once:** it is guaranteed that a message will be delivered at least once to the receiver. The sender will store the message until it gets an acknowledgement in form of a PUBACK command message from the receiver.
- **QoS 2 – exactly once:** it guarantees that each message is received only once by the counterpart. It is the safest and also the slowest quality of service level. The guarantee is provided by two flows there and back between sender and receiver.

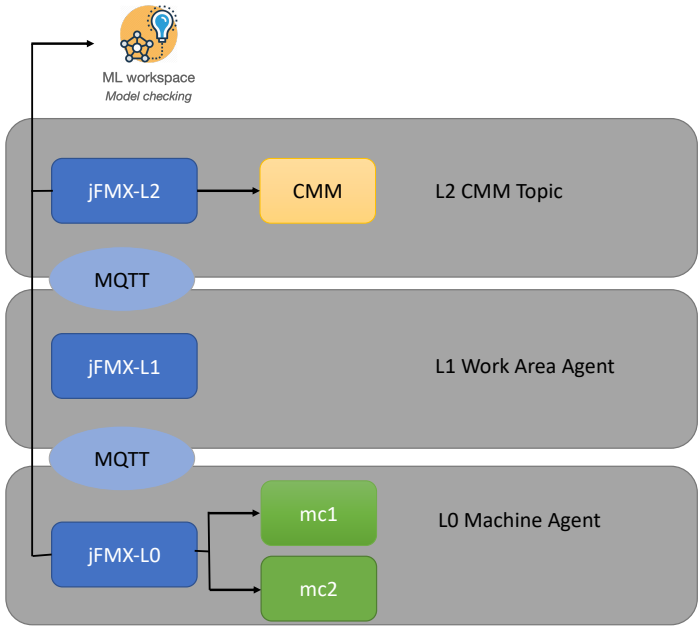


Figure 3.7.: Flow chart of the data layer: jFMX MQTT Namespace.

The MQTT topic namespace was defined to manage interactions with the IoT application running on the jFMX gateway hierarchy. Figure 3.7 shows the hierarchy of the jFMX MQTT Namespace. Based on this criterion, our IoT application running on an IoT gateway may be viewed in terms of the resources it owns and manages as well as the unsolicited events it reports:

- **account_name**: Identifies a group of devices and users. It can be seen as partition of the MQTT topic namespace. For example, access control lists can be defined so that users are only given access to the child topics of a given account_name.
- **client_id**: Identifies a single gateway device within an account (typically the MAC address of a gateway's primary network interface). The client_id maps to the Client Identifier (Client ID) as defined in the MQTT specifications.

- **app_id**: Unique string identifier for application (e.g., “L0” for mc topics, “L2” for CMM topics).
- **resource_id**: Identifies a resource(s) that is owned and managed by a particular application. Management of resources (e.g., sensors, actuators, local files, or configuration options) includes listing them, reading the latest value, or updating them to a new value. A resource_id is a hierarchical topic, where, for example, “fms/mc1/spindle/temp” may identify a temperature sensor and “fms/sh/Y/pos” a position sensor.

L0 Machine Center Topic

The jFMX MQTT publisher is executed by the L0-Gateway (Flight Recorder) and delivers message related to different application included inside the L0-MachineAgent. All the topics published by the L0 MachineAgent is related to the processing parameters of mc and have an app_id defined as: JFMX/L0/workAreaName/unitName where the workAreaName is the absolute unique name of the workarea and the unitName is the name of the unit inside the workarea. For our PdM task we refer to the topics related to the Working Step Analyzer application embedded in the L0-Machine Agent related to the two different mc (see Table 3.5) The topics related to the dataObj field represent the considered processing parameters (see Table 3.6) acquired by mc (i.e. computer numerical control [CNC] and accelerometer). The CNC and accelerometer data were acquired by a sampling frequency of 24 and 100 Hz respectively.

L2 CMM Topic

The L2 CMM topics represents the condition monitoring data that were acquired by a robotic part loading system for coordinate measuring machine (see Table 3.7). The condition monitoring data reflect the quality of processing in terms of the measured deviation (deviation) with respect to the optimal condition. The optimal condition highlighted no deviation compared to the planning processing. The alarm situation is triggered once the measured deviation overcomes the admitted tolerance.

All the L2 CMM Topic and L0 Machine Center Topic were synchronized by considering the physical tool (**tl**, identifier of the tool formatted as <tooltype>/<tool serial number>), the part machined (**pt**) and the type of processing (**frindex**). Although the system could consider all the type of processing we took into account the drilling procedure (i.e. FRINDEX=10,20,30). This procedure has the intrinsic advantage of being standard, i.e. independent of the tl. The synchronized L2 CMM Topic together with the L0 Machine Center Topic were saved in a SQL database.

Feature extraction

All the computed KPIs (speed, pow, pos, curr) represent the predictors of the ML model for each observation/physical tool (i.e. specific triplet tl, pt and frindex). The

Table 3.5.: Topics related to the working step analyzer application embedded in the L0-Machine Agent.

Field	Type	Description
ts	Date	start date of the episode according to the clock of the mc
l2mc	String	mc code
l2wa	String	workArea code
ordNo	String	order number
ptType	String	identifier of the part type
opNo	String	operation number
dType	String	identifier of the workingstep message type (”wsEv”: in case of sensor coming from mc, ”accTrace”: in case of sensor coming from Accelerometer)
slot	Int	sub section as indicated by the part program
pt	String	identifier of the part machined
tl	String	identifier of the tool formatted as ¡tooltype¿/¡tool serial number¿
life	Int	life of the tool at the beginning of the step
sensor	String	name of the sensor
unit	String	measure unit for the specifc sensor
sampling	Int	milliseconds sampling interval
dataObj	Object	complex object containing two elements: n[int] segment of the acquisition, data: Array[Double] data acquired
mc	String	machine name
wa	String	workArea name

output of the ML model was represented by the percentage measurement error (error %).

Notation

We let a candidate univariate time series of a specific sensor $\boldsymbol{S}$ collected from a CNC sensor as $\boldsymbol{X} = \{\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_T\}$ where T denotes the number of observations. Notice how this time series is relative to the signal of a specific sensor and relative to a specific triplet comprised of physical tool (tl), the part machined (pt) and the type of processing (frindex). We denote the $error\%$ as a direct quantitative measure about the machine quality and the $deviation$ and $tolerance$ the measured deviation and tolerance reported in Table 3.7.

Our feature extraction strategy is based on a geometric area analysis (GAA) and trapezoidal area estimation (TAE) procedure that is widely used for solving novelty and anomaly detection task [127, 128]. The relative KPI is computed by temporally

Table 3.6.: Topics related to the dataObj field.

Sensor (CNC)	**unit**	**Description**
SP_SPEED	rpm	spindle rotation speed
SP_POW	W	spindle power consumption
X_AXIS_CURR	A	x-axis current consumption
Y_AXIS_CURR	A	y-axis current consumption
Z_AXIS_CURR	A	z-axis current consumption
B_AXIS_CURR	A	B-axis current consumption
A_AXIS_CURR	A	A-axis current consumption
X_AXIS_POS	μm	X-axis position
Y_AXIS_POS	μm	Y-axis position
Z_AXIS_POS	μm	Z-axis position
B_AXIS_POS	μm	B-axis position
A_AXIS_POS	μm	A-axis position
FRINDEX	None	type of processing
Sensor (accelerometer)	**unit**	**Description**
velMOD_RMS	mm/sec	3 directions vibratory speed module

Table 3.7.: L2 CMM topics.

Field	**Type**	**Description**
ts	Date	start date of the episode according to the clock of the mc
pt	String	identifier of the part machined
tl	String	identifier of the tool formatted as ¡tooltype¿/¡tool serial number¿
FRINDEX	Int	type of processing
measured	Double	measured value
deviation	Double	measured deviation
tolerance	Double	admitted tolerance

normalizing the TAE as follows:

$$\begin{aligned} KPI &= \frac{1}{T}\int_1^T x_t dt \\ &= \frac{1}{T}\sum_{t=1}^{T}\int_{t-1}^{t} x_t dt \\ &\approx \frac{1}{2}\frac{1}{T}\sum_{t=1}^{T}(t-(t-1))[x_t - x_{t-1}] \end{aligned} \tag{3.6}$$

All the computed KPI for each specific sensor $\boldsymbol{S}$ is depicted in Table 3.8. For the position and current a global KPI was extracted by computing the euclidean distance of X,Y,Z axis.

Table 3.8.: Extracted KPIs for specific sensor. KPIs are related to rotation speed (speed), power consumption (pow), position (pos) and current consumption (curr).

Sensor (CNC)	**KPI**
SP_SPEED	speed
SP_POW	pow
X,Y,Z_AXIS_POS	pos
X,Y,Z_AXIS_CURR	curr

For each observation, the error % was computed by considering the measured deviation and the associated tolerance as follows:

$$error\% = \frac{deviation}{tolerance} * 100 \tag{3.7}$$

The error % measurement reflects a direct and quantitative measure about the machining quality. In particular, an error % greater than 100 reflects an out of tolerance machining while an error % lower than 100 correspond to a machining that does not exceed the tolerance limits.

The final dataset consist of 438 observations/physical tools collected by two different mc from the 1st October 2019 to the 31st May 2020.

Predictive model

For solving the regression task we have taken into account predictive performance, interpretability, and predictive accuracy. These factors represent also the three fundamental requirements defined by the company for solving the PdM task. For this reason, the Random Forest (RF) model was selected for solving the regression task. In this case, it consists of an ensemble of regression trees (RTs) (i.e., n° of RT) generated by independent identically distributed random vectors. Since we aim to solve a regression problem, the best splitting features for each node was computed according to the sum of squared error.

Although RF allows learning a non-linear decision boundary, the RF originated as ensemble tree based model ensures an intuitive notion of interpretability: it allows providing a direct interpretation of the most discriminative KPIs. However, the degree of interpretability depends on the model size (i.e., number of weak learners/regression tree and depth of the tree) [129]. Hence, the interpretability of RF was encouraged by constraining the number of weak learners and the depth of the tree in the validation set. This lead also to control the computation effort for the training phase.

The importance of a specific KPI in the RF model to identify the percentage measurement error (error %) was measured according to a permutation of out-of-bag feature observation [130]. A KPI is considered relevant to identify the error %, if permuting its values should affect the model error. On the other hand, if a KPI was not relevant, then permuting its values should not affect significantly the model error. The permutation importance of each feature is computed as: $1 - error$ (after permuting the feature values). Compared to the standard impurity-based importance the permutation approach is unbiased towards high cardinality features and measure directly the ability of feature to be useful to make prediction [131].

Cloud architecture

A container logic was adopted for packaging the ML application and all its dependencies, so the application runs reliably from one computing environment to another. A docker image is essentially a snapshot of a container. Microsoft Azure IoT portal was adopted for providing a cloud-based environment based on virtualized containers. This environment can ensure hardware and software isolation, flexibility, and inter-dependencies between the IoT devices and data collection, features extraction, and prediction phases. These properties are suitable for our industrial use case since the proposed DSS is currently designed to work with four operating machines and it provides the capability to be scaled up to collect a huge amount of data from different interconnected machines. This advantage also lies the foundations to continuously update the model, once a new machine is connected to the system.

The proposed architecture is depicted in Figure 3.8 Cloud Architecture. We used a Python ML library for training and testing our feature extraction stage and RF model with respect to other state-of-the-art ML approaches. Afterward the feature extraction procedure and the containerized RF model was pushed to Azure Container Registry. During this step, we included the *azureml-monitoring* and *azureml-defaults* for enabling respectively the data collection feature and the deployment to Kubernetes. Consequently, the ML model was deployed to Azure Kubernetes Service (AKS). For our purpose we configured the AKS with 3 agent nodes of type Standard_D3_v2 (4 vCores), thus leading to a total of 12 vCores. Additionally, in our AKS configuration, we explicitly enable data collection (input data and predictions outcome). The L2 CMM Topic together with the L0 Machine Center Topic were exported from SQL database to azure blob storage, by allowing a continuous testing and update/retraining of the ML model once, for instance, a drift situation was detected. All the prediction results together with the model weights (i.e. decision rule of the ensemble trees) were stored in the Azure Blob storage.

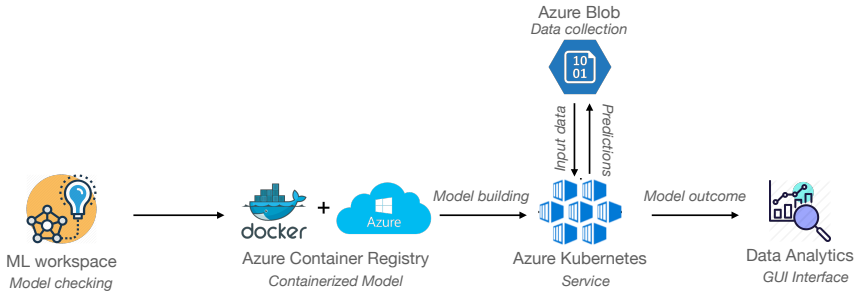


Figure 3.8.: Cloud architecture: ML workspace, Azure Container Registry, Azure Kubernetes and Azure Blob

Data analytics

A GUI interface was created to display the predicted error % over different tl, pt, frindex. In particular, the GUI was finalized to provide a timely indication to the machine operator when the error % exceeds a certain tolerance threshold that may be different for each tl, pt, frindex. Additionally the Azure Application Insights instance was enabled for providing a high level overview of the deployed API in terms of featuring failed requests, response time, number of requests and availability. The log analytic feature of the Application insights allows to view and inspect the logs provided from our containerized model in terms of *stdout* and *stderr*.

3.2.3. Performance evaluation and Results

In this Section the experimental results for the proposed DSS specifically tailored for solving PdM task are shown. The prediction of the processing quality represents the main task we aim to solve. All the results related to the predictive performance and computation effort of the proposed approach with respect to the state-of-the-art approaches are depicted whit the results related to the model interpretability, i.e. the most relevant KPIs. In addition, more details on the implemented GUI for the proposed DSS are reported.

Experimental procedure

The RF-based PdM approach was compared with respect to other state-of-the-art ML approaches employed for solving PdM tasks. In particular, the following models have been considered:

- LR with ridge penalty (LR ridge) [132];
- LR with elastic net penalty (LR elastic) [132];
- Regression Tree (RT) [54, 24];

- XGB [6];
- SVM with Gaussian Kernel [124]
- Multi-Layer Perceptron (MLP) [45, 46]
- LSTM [47, 48]

A 10-fold CV procedure was performed in order to evaluate the performance of the RF model. The hyperparameters were optimized by implementing a grid search in a nested 5-CV. Hence, each split of the outer CV loop was trained with the optimal hyperparameters (in terms of mean squared error) tuned in the inner CV loop. Despite this model checking procedure is expensive in terms of computation effort it allows to obtain an unbiased and robust performance evaluation [133].

Table 3.9 shows the different hyperparameters for the proposed ML models and all competitors' ML approaches, as well as the grid-search set. For the LR ridge the λ penalty controls the 2-norm regularization. For the LR Elastic $\alpha = \lambda_1 + \lambda_2$ and l1_ratio=$\frac{\lambda_1}{\lambda_1+\lambda_2}$ where λ_1 and λ_2 control separately the 1-norm and 2-norm regularizations.

Table 3.9.: Range of Hyperparameters (Hyp) for the proposed ML models and all competitors' ML approaches.

Model	**Hyp**	**Range**
RF	n° of regression trees	$\{5, 10, 15, 20, 25\}$
	n° of features to select	$\{1, 2, 4\}$
LR ridge	λ	$\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$
LR elastic	α	$\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$
	l1_ratio	$\{10^{-4}, 10^{-3}, 10^{-2}, 0.1, 0.2, 0.3, 0.4, 0.5\}$
RT	max depth	$\{5, 10, 15, 20, 25\}$
	min n° of leaf size	$\{5, 10, 20, 50, 100\}$
XGBoost	learning rate	$\{0.001, 0.01, 0.10\}$
	max n° of estimators	$\{5, 10, 15, 20, 25\}$
	max depth	$\{5, 10, 15, 20, 25, 50, 75\}$
	n° of features to select	$\{1, 2, 4\}$
SVM Gaussian	Box Constraint	$\{1, 5, 10, 50, 100, 500, 10^3, 5 \cdot 10^3, 5 \cdot 10^4, 10^4\}$
	Kernel Scale	$\{10^{-2}, 0.1, 1, 10, 10^2, 10^3, 10^4\}$
MLP	learning rate	$\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$
	n° of hidden layers	$\{1, 2, 4\}$
	n° of units	$\{4, 8, 16, 32\}$
LSTM	learning rate	$\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$
	n° of hidden layers	$\{1, 2, 4\}$
	n° of units	$\{4, 8, 16\}$

Evaluation metrics

The following metrics were considered to evaluate the predictive performance of the regression task described in Section 3.2.2:

- Mean Absolute Error (MAE), which measures the absolute difference between the predicted and the ground truth error %

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\|y_i - \hat{y}_i\|; \tag{3.8}$$

- Mean Squared Error (MSE), which measures the squared difference between the predicted and the ground truth error %

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2; \tag{3.9}$$

- R^2 score (coefficient of determination), which is a proportion between the variability of the data and the correctness of the model used. It varies in range: $[-\infty; 1]$[134]

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{3.10}$$

where n is the number of data points, $\hat{y}_i$ is the predicted value of y_i and $\bar{y}$ is the mean value of y. The statistical significance of the R^2 score was evaluated at the 5% significance level with respect to the zero value. The R^2 score distribution over each CV fold was found to follow a normality distribution according to the Anderson-Darling test (A = 0.436, $p = 0.246$). Hence, we used the parametric paired t-test ($\alpha = 0.05$) to compare the performance of the proposed approach with respect to state-of-the-art work.

Predictive performance

The predictive performance of the RF regression model is shown in Table 3.10. It can be noted that the best prediction results were obtained for our RF and XGboost model (R^2 score 0.868 and 0.877 respectively), while the LR model achieved the lowest predictive performance (R^2 score 0.591). Accordingly the RF and XGboost models show similar and competitive performance in terms of MAE (0.089 and 0.088 respectively) and MSE (0.018 and 0.017 respectively). R^2 score distribution of RF is significantly higher ($p < .05$) than ML based regression model (i.e. LR ridge, LR elastic net, RT, SVM Gaussian) and DL based regression model MLP and LSTM. In particular, the performance of sequential DL approaches might be limited by the low

presence of a huge amount of annotated sequential data in this PdM scenario, in order to learn spatio-temporal dependencies. Figure 3.9 shows the comparison between the predicted error % from RF and its real values obtained from the L2 CMM Topic. We focused on a subset of 42 testing samples (one fold of CV-10 procedure).

Table 3.10.: Predictive performance and computation effort of ML approaches. * indicate whether R^2 distribution over the 10-fold is significantly higher than 0.

Model	MAE	MSE	R^2	Train + Val (sec)	Testing (sec)
LR ridge	0.166	0.054	0.591*	0.521(0.007)	$< 10^-3(0)$
LR elastic	0.166	0.054	0.591*	0.531(0.058)	$< 10^-3(0)$
RT	0.100	0.023	0.835*	0.310(0.022)	$< 10^-3(0)$
XGBoost	0.088	0.017	0.877*	80.740(10.057)	$< 10^-3(0)$
SVM Gaussian	0.151	0.047	0.648*	1.223(0.101)	$< 10^-3(0)$
MLP	0.161	0.051	0.618*	39.454(0.651)	0.062(0.002)
LSTM	0.156	0.044	0.667*	89.549(2.618)	0.570(0.023)
RF	**0.089**	**0.018**	**0.868***	2.353(0.328)	$< 10^-3(0)$

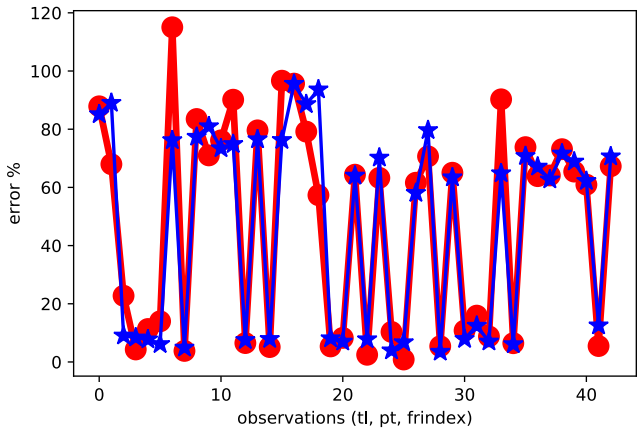


Figure 3.9.: Predictive performance of RF for the estimation of the error % over a subset of observations (tl, pt, frindex): red line ground truth, blue line prediction.

Computation effort

Taking into account the high performance achieved by the proposed approach, we decided to test the computation effort with respect to other state-of-the-art ML ap-

proaches. Table 3.10 compares the time effort (test and training+validation stage) of the proposed RF algorithm with respect to other state-of-the-art algorithms. All the experimental comparisons were performed on Intel Core i7-4790 CPU 3.60 GHz with 16 GB of RAM and NVIDIA GeForce GTX 970. Although the predictive performance of RF is similar to XGboost, the training and validation of RF model are significantly ($p < .05$) faster than XGboost with a gain of 34x. This peculiarity ensures the possibility to retrain the ML model in the cloud with an average latency of 2.353 sec for learning from around 400 new samples. At the same time the RF prediction latency may be neglected $< 10^{-3}$ sec and the RF model can give a timely and consistent prediction.

Interpretability

The interpretability of the proposed RF model was measured according to the feature/permutation importance (Figure 3.10). The speed KPI achieved, on average, the highest permutation importance score, thus highlighting the most discriminative power of this KPI with respect to the other features. The feature importance together with the predicted error % values are the salient ML outcomes of the proposed DSS for supporting the maintainer/operator during the machining quality task. For instance, taking into account the predicted error %, the operator may exploit preventive action in order to avoid future errors during the machine processing. At the same time, the localization of the most discriminative KPI may address the human operator to detect the source of the error, while optimizing the overall equipment effectiveness, productivity, and quality of production.

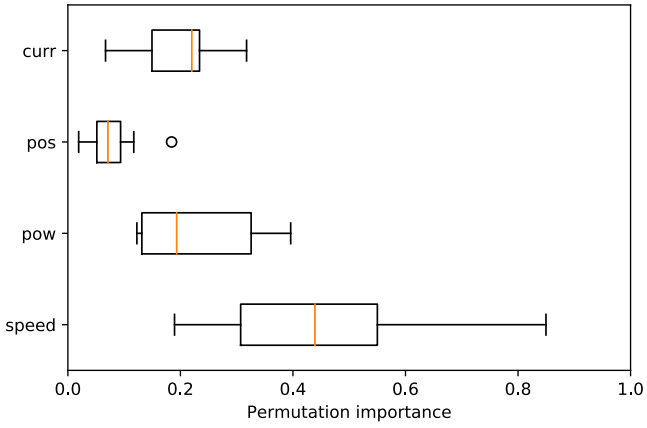
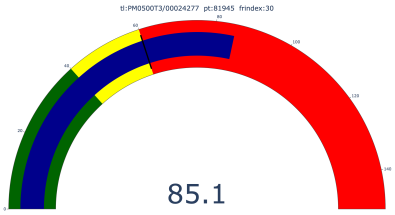
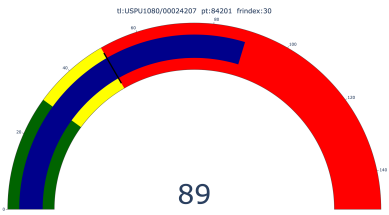


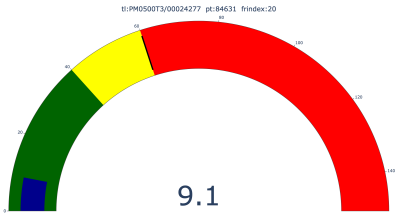
Figure 3.10.: Feature/permutation importance of RF model.



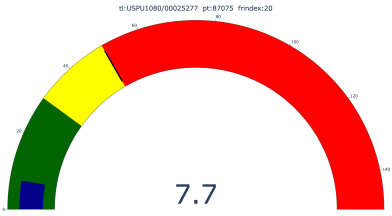
(a) GUI interface observation tl:PM0500T3/00024277 pt:81945 frindex:30



(b) GUI interface observation tl:USPU1080/00024207 pt:84201 frindex:30



(c) GUI interface observation tl:PM0500T3/00024277 pt:84631 frindex:20



(d) GUI interface observation tl:USPU1080/00025277 pt:87075 frindex:20

Figure 3.11.: Examples of GUI interface of the proposed DSS for four different observations (tl, pt, frindex): the predicted error % is represented by blue line and reported below the gauge chart, the black line represents the admissible tolerance threshold that can change across different triplets. Red bar: alarm event; yellow bar: potential risk situation; green bar: within tolerance limit.

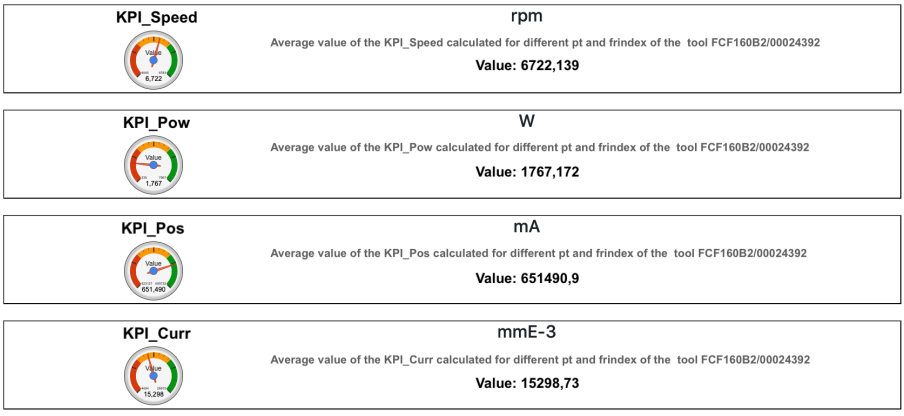


Figure 3.12.: Example of the proposed GUI interface displaying the average value of the KPI predictors for a specific tl across different pt and frindex.

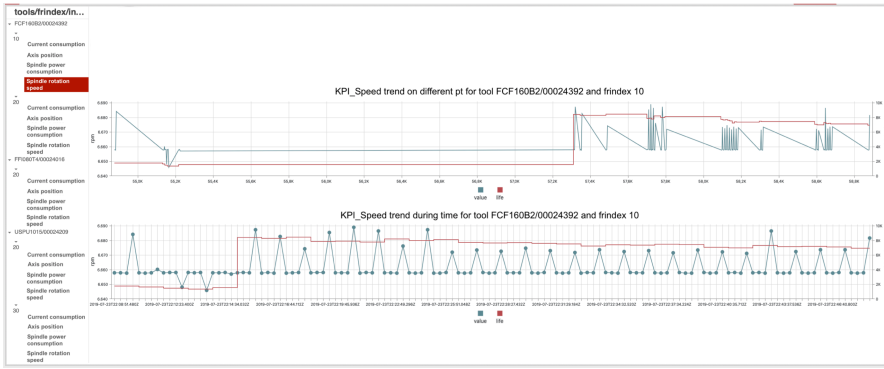


Figure 3.13.: Example of the proposed GUI interface displaying the trend of KPI predictors (blue line) together with the life parameter (red line) (i.e. life of the tool at the beginning of the step) for a specific tl and frindex across pt (top graph) and across time (bottom graph).

Data analytics: GUI interfaces

Figure 3.11 shows an examples of GUI interfaces of the proposed DSS for four different observations (tl, pt, frindex). We represent the predicted error % and the tolerance limits, which can be different for each observation. In particular, a predicted error % greater than the admissible threshold (red bar) represents a significant machining error (i.e. alarm event), while an error % that falls within the yellow bar correspond to a potential risk situation (i.e. machining error below but close to tolerance limits). The green bar reflect how the machining is properly executed within the tolerance limit.

Taking into account the high predictive results and the interpretability of the proposed approach, our GUI interface is not limited to show only the predicted error % (i.e. Figure 3.11). In fact, we go further by supporting the operator by showing the average value of the KPI predictors (see Figure 3.12) for a specific tl across different pt and frindex. Additionally, the trend of KPI predictors together with the life parameter (i.e. life of the tool at the beginning of the step) is displayed for a specific tl and frindex across pt and across time in a separated dashboard, as shown in Figure 3.13.

Chapter 4.

Aesthetic Quality Control

The Aesthetic Quality Control (AQC) task is an unexplored and challenging QC application related to the aesthetic evaluation of a material, where the aesthetic aspect of the product is not measurable and is based on expert observation (see Section 1.1). This task is usually done by a technician that classifies each of the items one by one merely using its expert knowledge and focusing on qualitative and subjective analyses. In particular, in this chapter we will discuss about the QC task of wooden stocks, which refers to a real industrial case study of Benelli Armi Spa: it is related to assign a certain grade to each item according to the aesthetic properties of wood (see Fig.4.1). In order to prevent the sale of a product that does not meet the expectations of the customer, the company conducts QC on wooden part made by external suppliers defining whether it complies with the quality requirements.

Wooden stocks are characterized by their uniqueness: each one is different from all the others. This difference between all of them leads to a classification problem of the samples according to the aesthetic aspect. It is possible to pass from woods with minimum variations of color to others with remarkable contrasts between light and dark veins that fit together in a twisted and variegated way. Grade increases as the grain in wood increases, and therefore the item will have a higher value from an aesthetic point of view and, consequently, also from an economic one. Generally, the commercial classification of wooden stocks is defined in five major categories ranging from grade 1 up to 5, where grade 1 indicates almost veinless wood and grade 5 a very twisted and variegated grain pattern. Each different type of rifle model manufactured by the company is equipped with a stock belonging to a specific grade class and this coupling is at the total discretion of the company according to its market decisions.

Today the aesthetic QC of wooden stocks is only based on the evaluation of the human eye. At first, the operator has to verify the conformity and integrity of the material, then decide whether the item has the right characteristics to be part of the grade class given by its manufacturer. If there is no agreement, the stock can be sent back or reclassified in another grade. As there is no the support of an objective method, the entire process is solely entrusted to the ability and experience of the technical staff, and this implies several main limitations and drawbacks. First of all, the results are affected by a high subjectivity that could give different responses depending on the

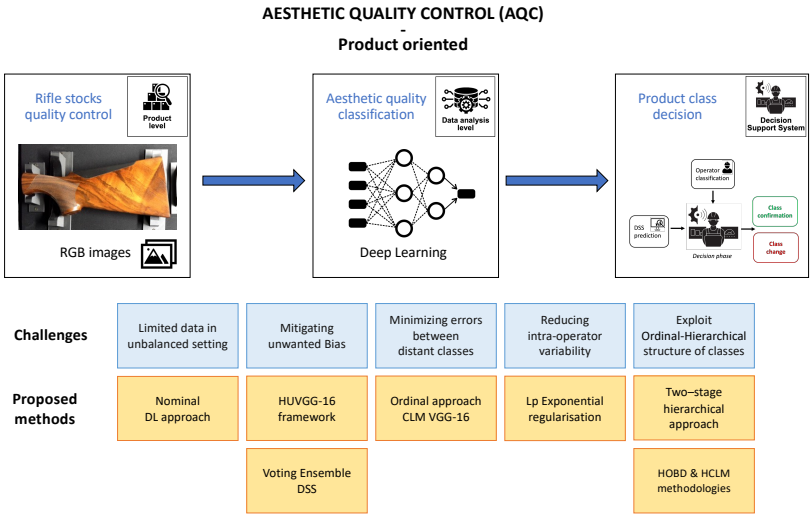


Figure 4.1.: Real-world industrial AQC problem: support the human operator in classifying the aesthetic quality of wooden stocks. AQC is designed according to the Quality 4.0 framework shown in Fig. 1.1. Related challenges and proposed methods for solving them are outlined.

time when the classification takes place and by which operator. Like all aesthetic evaluations, the personal judgement leads to intrinsic variations in results and therefore alterations are routine, especially on high quality woods. Moreover, a training period is necessary for each operator to acquire the required expertise, and this implies a significant investment in time and resources. Another issue to consider is that the number of samples to be checked depends on the lot size, but the QC is a time-consuming task: for this reason, the operator inspection consists in examining only a minimum part of the stocks delivered by suppliers, following that many pieces are considered right without being evaluated. The application of ML and DL techniques offers great opportunities to solve these issues and automatize the overall AQC process, saving time and resources and maximizing the performances reducing the high intrinsic variability across different human operators.

In this chapter, Section 4.1 and Section 4.2 present the collected wooden stocks dataset and how our AQC task can be addressed by standard DL method, respectively. Then, in Section 4.3 two different strategies are proposed for bias mitigation: HUVGG-16 framework (Section 4.3.1) and Voting ensemble approach (Section 4.3.2). Section 4.4 describes the proposed ordinal DL methodology based on CLM VGG-16 architecture. Section 4.5 introduces a novel exponential loss regularisation based on L_p norm. Section 4.6 refers to the proposal of a hierarchical DL framework for exploiting also the hierarchical property of the dataset. Section 4.7 proposed the

HCLM and HOBD methodologies for simultaneously learning hierarchical-ordinal constraints with a single network. Finally, in Section 4.8 the design of the proposed DSS for AQC is shown. For each section, the proposed methodologies, experimental procedure and results are described.

4.1. Data collection

The first version of the collected dataset is composed of both left and right side images belonging to 951 different rifles, for a total of 1902 images with a size of 1000×500 pixels. According to the aesthetic quality of wood, the stocks have been classified into 4 main grades (1, 2, 3, 4) and their relative minor grades ($2^-, 2^+, 3^-, 3^+, 4^-, 4^+$), resulting into 10 different classes as reported in Table 4.1. All the grades are reported together with the number of stocks. The majority class is the grade 4 (270 stocks) while the minority class is the grade 4^+ (106 stocks). Note that the class 1 has not been divided into minor labels because the company produces model series with higher quality classes. Figure 4.2 shows an example of stock for each of the 10 classes. The images were acquired with a high-definition RGB camera placed in the top-view configuration. During the annotation procedure, a highly specialized technician accurately inspects the item and assigns the labels of the stock using a custom data annotation platform (see Figures 4.3 4.4). Table 4.2 shows the classes distribution for each rifle series. With the aim of deeply investigating the relationship between the most confounding classes (i.e. 3 and 3^+), additional images were acquired leading to the final dataset configuration shown in the Table 4.1 comprising of 2120 images. The distinction between the above classes is important also from a business point of view: most of the rifles series produced by the company mount these wood grades.

The detention and conservation of the acquired dataset are regulated by an agreement between Benelli Armi Spa and Università Politecnica delle Marche.

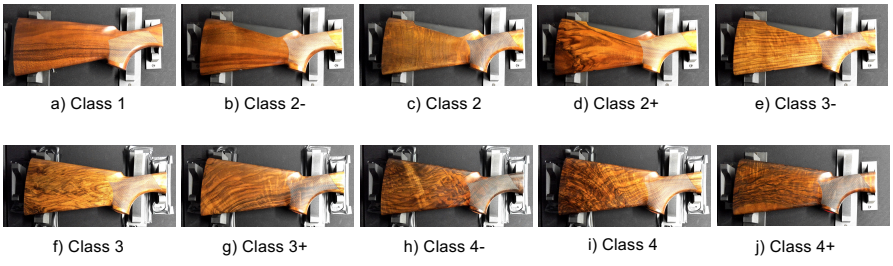


Figure 4.2.: Example of different stocks for each aesthetic quality class belonging to the collected dataset.

Table 4.1.: Aesthetic quality classes distribution for both versions of the rifles stocks dataset.

Label	1	2^-	2	2^+	3^-	3	3^+	4^-	4	4^+
First version	165	148	212	177	168	250	198	208	270	106
Final version	165	148	212	177	179	306	344	208	275	106

Table 4.2.: Aesthetic quality classes distribution for each rifle series for the first dataset version.

Rifle series (ID code)	1	2^-	2	2^+	3^-	3	3^+	4^-	4	4^+
ACCADEMIA GR3+ (2)	0	0	0	0	0	24	85	9	2	0
RAFFAELLO 2013 GR3 (3)	0	0	0	11	130	9	0	0	0	0
RAFFAELLO 2013 GR2 (4)	0	32	149	26	14	3	0	0	0	0
ANNIVERSARY 5O° GR4 (6)	0	0	0	0	0	1	19	101	252	61
828 NIKEL GR3+ (8)	1	0	0	0	11	176	14	0	0	0
828 CAL20 GR3+ (9)	0	0	1	0	2	35	70	12	2	0
MONTFELTRO EUROPE GR1 (10)	151	19	18	4	7	1	0	0	0	0
MONTEFELTRO CLARO GR2 (11)	3	79	27	105	4	0	0	0	0	0
ARGO E GR4 (12)	0	0	0	0	0	0	0	8	2	18
FRANCHI EUROPE GR2 (13)	10	18	17	31	0	0	0	0	0	0
ANNIVER. 50° CAL.20 GR4 (14)	0	0	0	0	0	1	3	35	1	14
RAFFAELLO CAL.12 GR4 (15)	0	0	0	0	0	0	7	43	11	13

Acquisition bench

The bench is composed of an industrial lamp and a high-definition RGB camera installed at the top of a photographic box, as shown in Figure 4.3. The box shields from external lights: this configuration guarantees the acquisition of images at high resolution with a uniform brightness, avoiding the reflection produced by curvature and polishing of stocks. Once the final configuration had been chosen, the internal staff of Benelli prepared the templates to facilitate the insertion of the items inside the box and to guarantee the homogeneity of the acquisitions.

Annotation Software

A custom software product has been developed to allow operators to acquire images, to note the quality class of the item and to store all in a dedicated database. The user must indicate for both sides of each stock-rod pair: a) the article typology; b) the overall quality class between stock and rod; c) the quality class of stock and rod individually. The interface of the annotation software is shown in Figure 4.4. In a second phase, in the same software the predictive models have been integrated so that, when the operator acquires the image, the classification obtained by the models is returned.

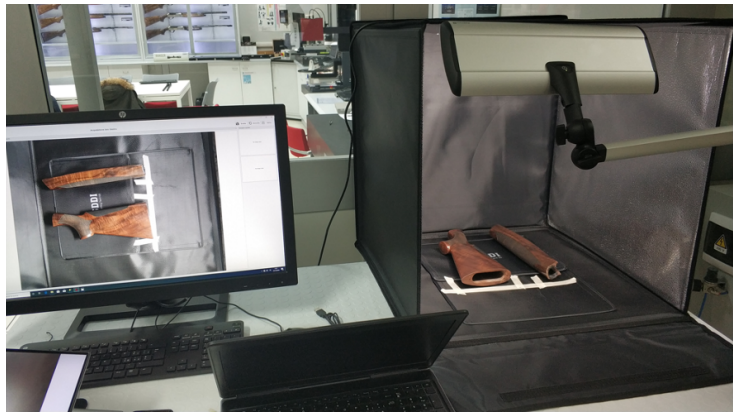


Figure 4.3.: Custom data annotation platform. The stock is placed in the box where the RGB camera and an industrial lamp are mounted. The annotation software allows the operator to capture the image and to record the grade.

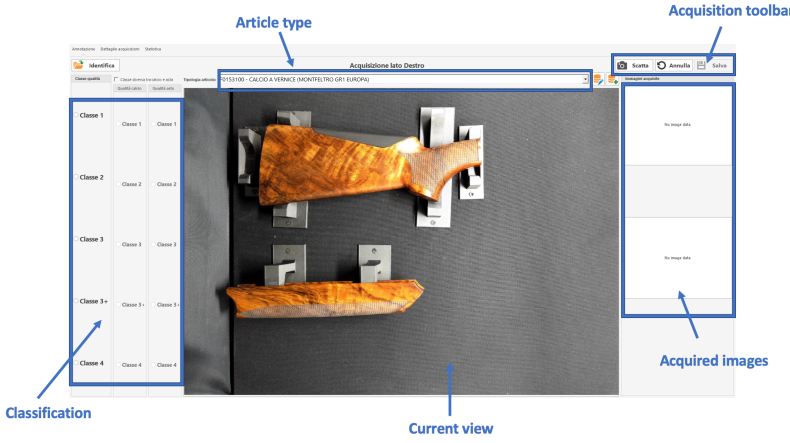


Figure 4.4.: The custom data annotation software presents several panels: “Classification”, where the user selects the overall quality class between stock and rod and the quality class of each item individually; “Article type” for the item typology and the rifle series it belongs to; “Current view” represents the field of view of the camera; “Acquisition toolbar” by which the user can take, cancel or save the image; “Acquired images” which allows a temporary display of the acquisitions.

4.2. Nominal Deep Learning approach

4.2.1. Task definition

For solving the AQC task corresponding to the classification of rifles stocks wood grade, the problem was firstly subdivided into different classification tasks, sorted according to the level of difficulty established by the company itself:

(a) prediction of *middle* quality classes: $Y = \{1, 2, 3, 4\}$;

(b) prediction of *meta* quality classes: $Y = \{1^*, 2^*, 3^*, 4^*\}$, where $1^* = \{1\}$, $2^* = \{2^-, 2, 2^+\}$, $3^* = \{3^-, 3, 3^+\}$, $4^* = \{4^-, 4, 4^+\}$;

(c) prediction of *all* quality classes: $Y = \{1, 2^-, 2, 2^+, 3^-, 3, 3^+, 4^-, 4, 4^+\}$;

where X and Y are respectively the input and the output space. We are interested to learn an agnostic model that classifies the quality classes without being given information about the specific rifle series. This is because, in the real-industrial case situation, the technician is engaged in the QC procedure where the rifle series is not known a priori.

4.2.2. Classification models

The proposed classification tasks are based on the fine-tuning strategy of state-of-the-art CNNs for image classification, i.e. AlexNet [135], VGG-16 [136] and ResNet50 [137]. A transfer learning approach was used to fine-tune the networks on ImageNet [138] pre-trained weights. These architectures were chosen for two main reasons: i) they achieved competitive performances on ImageNet challenge, ii) they are relatively simple (i.e. not too deep), allowing to obtain low-level features for fine-tuning. For all the networks, the last fully-connected layer was modified from 1000 to K neurons, where K is the dimension of output space for each task as defined in Section 4.2.1 - Task definition.

AlexNet

The architecture consists of 5 convolutional layers followed by 3 fully-connected ones. Rectified linear unit (ReLU) activation function is applied after every convolutional and fully-connected layer and Dropout is inserted before the first and the second fully-connected layers. The output of the last fully-connected layer is fed to softmax which produces a distribution over the class labels.

VGG-16

This network is composed of 13 convolutional layers that extract image features. Each convolutional block has filters with a 3×3 pixels receptive field and is followed

by a ReLU activation function. For CNN-parameter dimensionality reduction, max-pooling layers are used after 2 or 3 convolutional blocks. Finally, 3 fully-connected layers followed by a softmax layer are used to predict a probabilistic label map. Dropout regularization layers were inserted after the first and second fully-connected layers with a rate of 0.3. As regards the fine-tuning, the first 4 convolutional blocks were frozen. The VGG-16 architecture is depicted in Figure 4.5.

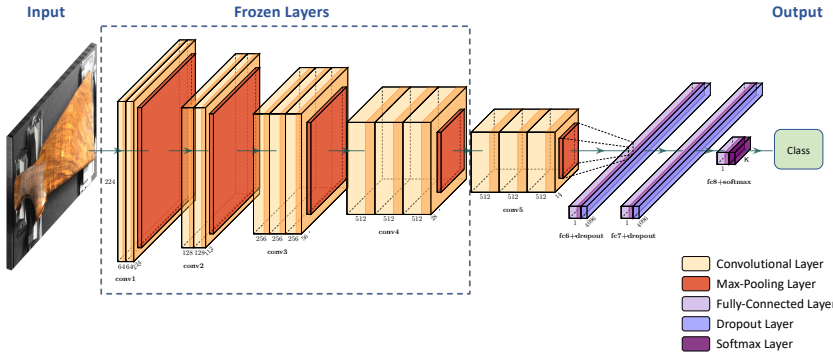


Figure 4.5.: The VGG-16 architecture.

ResNet50

It is a Residual Network having 50 layers. Compared to VGG-16, ResNet50 has an additional identity mapping capability that allows to bypass a CNN weight layer if the current layer is not necessary. This shortcut reduces the vanishing gradient problem and it prevents to avoid overfitting. In ResNet50, each ResNet block is 3 layers deep. In our case, all the modules were fine-tuned.

Loss functions

As regards loss functions, each multi-class classification task a), b) and c) was independently solved by considering respectively i) a standard Categorical Cross-Entropy (CCE) and ii) an ordinal categorical cross-entropy (OCCE). The standard CCE loss function is defined as follows:

$$\mathcal{L}_c = \sum_{i=1}^{K} (t_i log p_i + (1 - t_i) log(1 - p_i)) \tag{4.1}$$

where K is the number of quality classes $|Y|$, t_i is the target class vector and p_i is the posterior probability vector. The target class vector is computed by encoding the ground-truth category $y = k$ with $t = (0, \ldots, 0, 1, 0 \ldots, 0)$ where only the element

t_k is set to 1. In addition, a custom ordinal categorical cross-entropy (OCCE) was defined to encourage an ordinal structure, which penalized the ranking as follows:

$$\mathcal{L}_o = 1 + \Delta \cdot \mathcal{L}_c \tag{4.2}$$

where

$$\Delta = \text{argmax}(t - \hat{t}) \tag{4.3}$$

The main idea behind the loss function defined in Equation 4.2 is to penalize the error between the target class vector (t) and the predicted target class vector ($\hat{t}$) according to the ranking (i.e. by penalizing more non-consecutive misclassified samples).

4.2.3. Performance evaluation and Results

Experimental procedure

All the networks considered were fed with stock images resized to 224x224 pixels in order to match the ImageNet input dimension. The first version of the dataset was employed. The mean value was removed from each image. A mini-batch stochastic gradient descend (SGD) was adopted as optimizer. The best batch size, the initial learning rate and the momentum in the range $\{32,\ 64,\ 128\}$, $\{1 \cdot 10^{-4},\ 1 \cdot 10^{-3},\ 1 \cdot 10^{-2}\}$, $\{0.8, 0.9\}$ were explored respectively. For each task, these hyperparameters have been validated in a separate validation set using a grid-search approach. The number of epochs was set to 30.

The dataset was split by a startified holdout procedure, i.e. using 60% of images as training, 20% as validation and 20% as test. Images belonging to the same rifle ID were maintained in the same set. This checking was performed to ensure that the algorithm may be able to generalize across different unseen rifle stocks. Due to the small dimension of the dataset, data augmentation was performed on-the-fly on the training set, applying horizontal flip, rotation and zoom to original images. To cope with the slight unbalance of the dataset, class weights were computed for weighting the loss function.

All the experiments were performed using TensorFlow 2.0 and Keras 2.3.1 frameworks on Intel Core i7-4790 CPU 3.60GHz with 16GB of RAM and NVIDIA GeForce GTX 970.

Evaluation metrics

The classification performance was evaluated according to CCR, Precision, Recall and F1_score, defined as in Section 3.1.3.

Classification performance

The results for solving the task a), b) and c) described in Section 4.2.1 are reported. Figure 4.6 shows the training and validation accuracy of VGG-16, ResNet50 and AlexNet across each epoch for solving task a) on the first version of the dataset. The validation accuracy of VGG-16 overcomes both that of the ResNet50 and AlexNet.

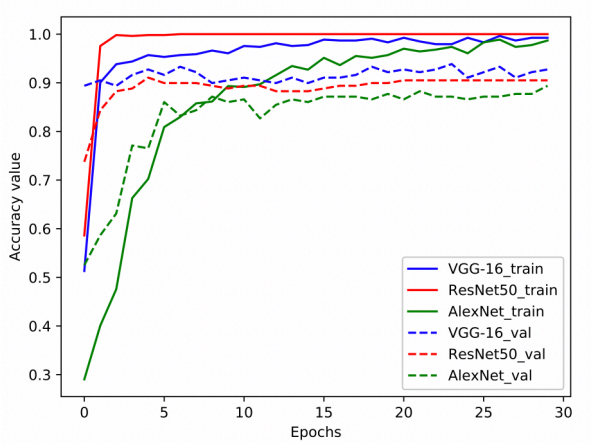


Figure 4.6.: Accuracy curves across each epoch during training and validation phases for VGG-16, ResNet50 and AlexNet using O-CCE loss for solving task a).

Table 4.3 shows the classification performance of VGG-16, ResNet50 and AlexNet for solving task a), b) and c) using the standard CCE loss on the test set.

For each task, the VGG-16 overcomes the ResNet50 and AlexNet in terms of CCR, Recall, Precision and F1. Hence, we show in Table 4.4 the performance of VGG-16 using the CCE and O-CCE as loss functions for solving tasks a), b) and c).

The O-CCE loss is more reliable for solving all tasks. The performance of the model decrease according to an increase in the difficulty of the task. Figure 4.7 shows the confusion matrices of the best performing model (VGG-16 O-CCE) for solving task a), task b) and c).

4.2.4. Bias detection

Starting from these results, we go further by analyzing any possible unwanted bias that may influence the classification performance. In particular, the bias may be unknown and embedded in the dataset/images. In this scenario, following the company's suggestion, we have pointed out different possible bias factors: rifle series, the instant of time (hour of the day) where the QC is carried out, stock sale id, production time (minutes). Considering the Cramer's correlation [139], we have analyzed how these possible bias factors are correlated with respect to the quality classes

Table 4.3.: Classification performance of VGG-16, ResNet50 and AlexNet for solving task a), b) and c) using the standard CCE loss on the test set. The best performing model in terms of F1 is reported in bold for each task.

Model	CCR	Precision	Recall	F1
Task a				
VGG-16	**0.961**	**0.954**	**0.962**	**0.952**
ResNet50	0.923	0.922	0.933	0.922
AlexNet	0.954	0.947	0.956	0.945
Task b				
VGG-16	**0.912**	**0.873**	**0.908**	**0.884**
ResNet50	0.882	0.862	0.874	0.845
AlexNet	0.896	0.843	0.874	0.853
Task c				
VGG-16	**0.632**	**0.612**	**0.623**	**0.601**
ResNet50	0.585	0.562	0.583	0.558
AlexNet	0.605	0.586	0.588	0.563

Table 4.4.: Classification performance of VGG-16 for solving task a), b) and c) using the CCE and O-CCE loss on the test set. The best performing model in terms of F1 is reported in bold for each task.

Loss	CCR	Precision	Recall	F1
Task a				
CCE	0.961	0.954	0.962	0.952
O-CCE	**0.961**	**0.962**	**0.962**	**0.964**
Task b				
CCE	0.912	0.873	0.908	0.884
O-CCE	**0.925**	**0.884**	**0.923**	**0.901**
Task c				
CCE	0.632	0.612	0.623	0.601
O-CCE	**0.657**	**0.643**	**0.658**	**0.633**

$Y = \{1, 2^-, 2, 2^+, 3^-, 3, 3^+, 4^-, 4, 4^+\}$. Table 4.2 shows that each specific rifle series is bounded to a specific quality class. Different rifle series have different exclusive characteristics, i.e. size, shape, color, polishing, plastic insert and other specific treatments (see some examples in Figure 4.2). The Cramer's analysis found that the rifle series is significantly correlated (0.67, $p < .05$) with respect to the ground-truth quality classes Y. As a consequence, the rifle series represents a bias in the VGG-16 model, due to the inherent production requirements. The detected bias is also demonstrated by the high significant Cramer's correlation (0.70, $p < .05$) found between the VGG-16 prediction of task c) and the rifle series. This fact is also confirmed by

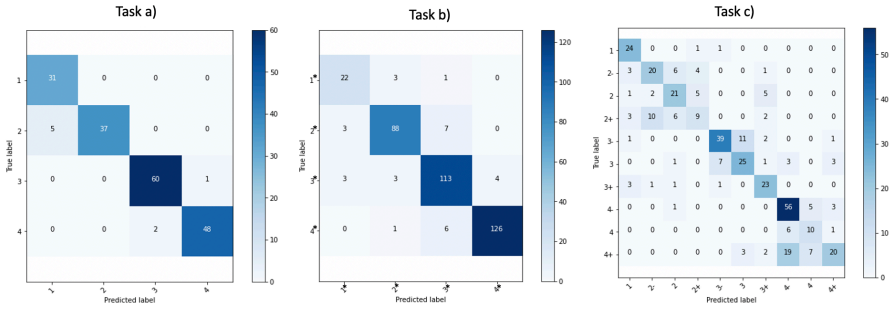


Figure 4.7.: Confusion matrices of the best performing VGG-16 models with O-CCE loss for solving task a), b), c) respectively.

exploring the saliency map of the VGG-16 (see Figure 4.8, left side) according to the approach proposed by [140]. The most discriminative pattern of the network is placed on the rifle edge, thus reflecting a more focus on the geometry (shape of the stock head) respect to the evaluation of wood grain.

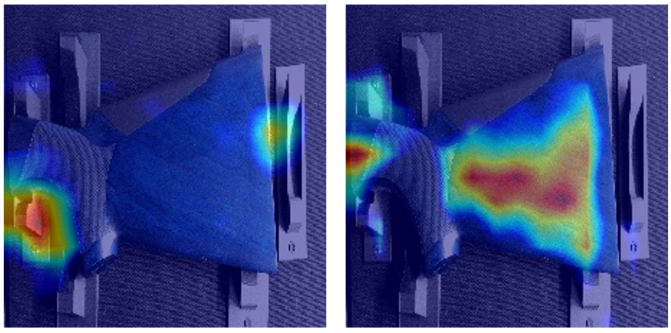


Figure 4.8.: Saliency maps of a grade 2 stocks belonging to *Raffaello* series. Left: VGG-16 model trained for task c); right: VGG-16 sub-network trained for solving the quality task on *Raffaello* series.

4.3. Bias mitigation

Starting from the assumption that the objective is to classify the quality classes without being given information about the specific rifle series, the model should be able to classify the quality grades independently from the geometry and the specific rifle characteristics. To achieve this purpose, first a hierarchical networks approach, named HUVGG-16, was proposed as described in Section 4.3.1. Then, a voting ensemble approach that better suited the AQC task from a practical perspective was designed,

reported in Section 4.3.2 .

4.3.1. HUVGG-16 framework

HUVGG-16 framework is designed to learn separately two-stage hierarchical networks that are able to predict respectively the rifle series (*model task*) and the quality classes (*quality task*). The single network of the first stage predicts all the rifle series; based on the rifle series predicted, we assign a specific second stage sub-network for classifying the quality classes. Each sub-network is conceived to predict only the quality classes associated with respect to the rifle macro-series. Each macro-series is defined according to the company's knowledge by aggregating rifle series which have the same geometrical characteristic. According to this, the first stage network was trained with all images of first version dataset; each second stage sub-network was trained only with specific classes associated with that series by production requirements (e.g. for *Montefeltro* series only classes $1, 2^-, 2$ and 2^+ have been considered). Figure 4.9 shows in detail the workflow of the proposed approach.

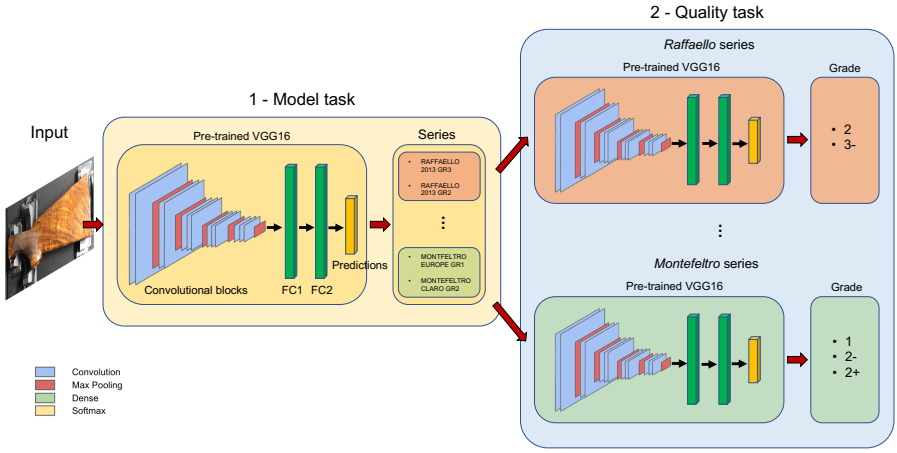


Figure 4.9.: Workflow of the proposed bias mitigation approach HUVGG-16. The first VGG-16 is conceived to learn the rifle series (*model task*) while each sub-networks is specialized to classify the quality classes (*quality task*) for each rifle macro-series. Each macro-series is defined according to the company's knowledge by aggregating rifle series which have the same geometrical characteristic.

Performance evaluation and Results

The same experimental procedure and preprocessing described in Section 4.2.3 was performed. The classification performance was evaluated according to CCR, Preci-

sion, Recall and F1_score, defined as in Section 3.1.3, and the results are reported in Table 4.5. Figure 4.10 shows the confusion matrix of HUVGG-16 for solving the model task and Figure 4.11 the ones for solving the quality task on *Raffaello* and *Montefeltro* series. The model task appears to be very easy for the CNN to solve, demonstrating that rifle characteristics largely affect the classification. Despite the similar classes considered for each quality task, the results are quite promising. Accordingly, the extracted saliency maps are constrained to focus on wood grains rather than the geometrical edges (see Figure 4.8 right side). Thus, this strategy allows to alleviate the bias by separating the two task and providing the prediction of quality classes for each rifle macro-series model.

Table 4.5.: Classification performance of HUVGG-16 for solving Model and Quality tasks and O-CCE loss on the test set.

Task	**CCR**	**Precision**	**Recall**	**F1**
Model task	0.973	0.982	0.973	0.972
Quality task				
Raffaello series	0.952	0.948	0.962	0.954
Montefeltro series	0.831	0.852	0.842	0.803

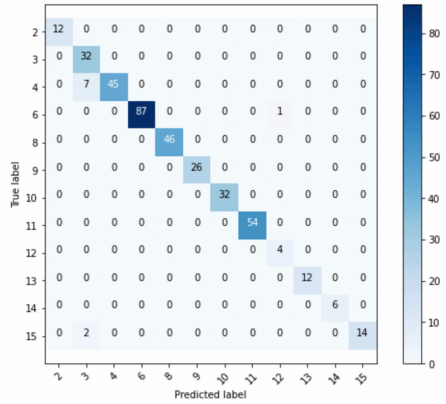


Figure 4.10.: Confusion matrix of model classification task of HUVGG-16. Labels represent the codes of rifle series as defined in Table 4.2.

Limitations

Despite the effectiveness of this approach, the proposed HUVGG-16 framework presents two main limitations:

- training one model for each rifle series is not convenient. Each time the com-

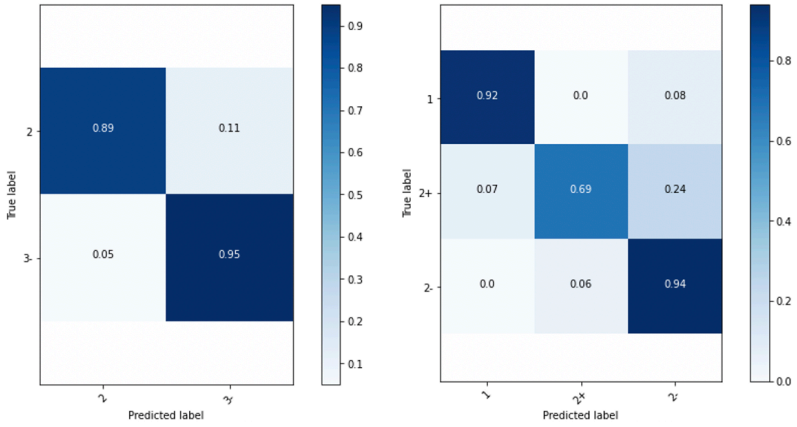


Figure 4.11.: Confusion matrices of quality classification task of HUVGG-16: Left: *Raffaello* series [codes 3,4]; right: *Montefeltro* series [codes 10,11].

pany defines a new rifle, there is the need to retrain the first stage network and to train a new sub-network related to the new product;

- second stage models predictions are "constrained" on the classes on which the models are trained for the *quality task*, but this does not meet the practical needs of the company for the QC task that should be solved.

To overcome these issues, a voting ensemble approach was designed as described in next Section 4.3.2.

4.3.2. Voting ensemble strategy

With the aim of reducing the bias due to the characteristics of the rifle model (shape, geometry, plastic inserts, size, knurling, etc.) and focusing only on the evaluation of wood grain, two key manipulations were applied to the dataset images (Figure 4.12): i) the extraction of a Region Of Interest (ROI) as focused as possible on the wood part and ii) the application of a ridge filter to enhance the grain pattern. These images were used as input for a voting ensemble approach, comprising of three different ML/DL models, according to the scheme reported in Figure 4.13.

Preprocessing and feature extraction

This step is useful to extract in a fully automated way a ROI where the percentage of wooden stock in the image is maximise in order to encourage the model to focus only on the aesthetic evaluation of the material. To obtain the ROI, the following steps were defined:

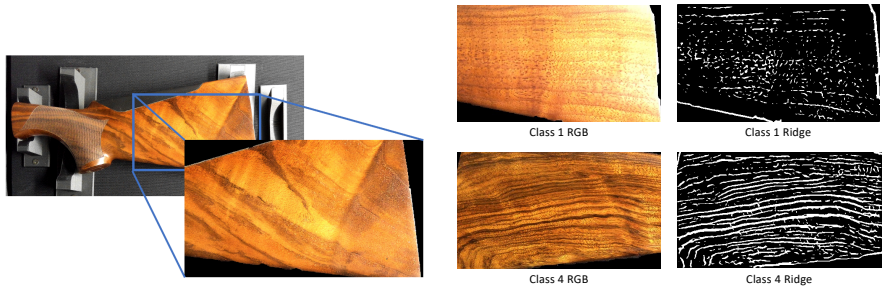


Figure 4.12.: Preprocessing steps for bias mitigation. Left: from each dataset image, a ROI focused on wood part was extracted. Right: ridge filter applied to class 1 and 4 images.

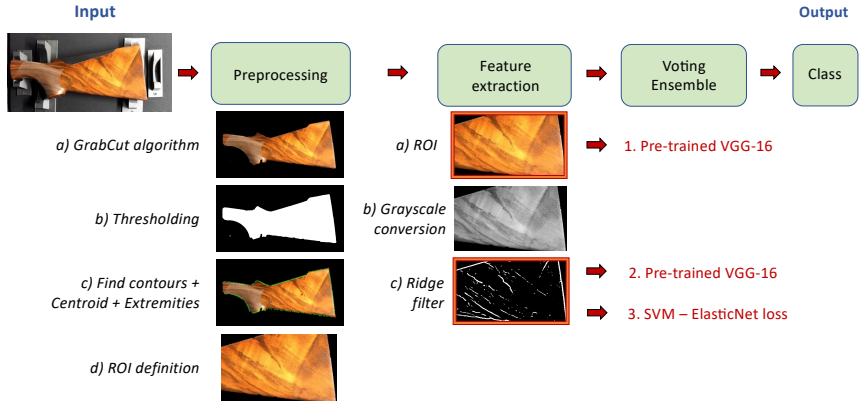


Figure 4.13.: The workflow of the VE predictive algorithm essentially consists of 4 steps: preprocessing, feature extraction, VE method consisting of 3 parallel ML-DL models and the classification output.

- *Grabcut algorithm*: it is image segmentation method based on graph cuts, which allows to remove the background of the image, obtaining a crop of the object of interest. The hyperparameters to be defined are the approximate area around the foreground region and the number of algorithm iterations (20 iterations were validated for this task);
- *Thresholding*: a boolean mask of the previously segmented region is obtained;
- *FindContours algorithm*: the contours of the previous mask are outlined;
- *Centroid detection*: the centroid of the area is calculated based on the defined contour points;
- *ROI definition*: a ROI (470×270 pixels) is defined as a surround of the centroid.

Considering that AQC classes are based on the grain pattern, the cropped images were converted into a grayscale images to enhance the properties of the wood. A binary Ridge filter with $\sigma = 2.0$ was applied. Figure 4.12 shows how effective the filter is to highlight wood grain and differentiate the various grades.

Classification models

The classification task is assigned to a Voting Ensemble (VE). A VE is an ensemble ML model that combines the predictions from multiple other models. This technique is often used in challenging ML problems to improve model performance, where combining more algorithms ideally allows to achieve better performance than any single model used in the ensemble. A VE works by combining the predictions from multiple models. In the case of classification, the predictions for each label are summed and the label with the majority vote is predicted. In particular, the soft voting method consists in summing the predicted probabilities for each class label and predicting the class label with the largest probability. VE are most effective when combining multiple fits of a model with different hyperparameters or when combining different models which consider different data features. In this case, the following methods were employed (Figure 4.13):

- pre-trained VGG-16 on RGB images;
- pre-trained VGG-16 on ridge images;
- SVM with ElasticNet regularization on ridge images.

Performance evaluation and Results

For the VGG-16 network, the same experimental procedure and preprocessing described in Section 4.2.3 was performed. For the SVM algorithm, hyperparameters weighing L2 and L1 were validated on a a separate validation set.

The classification performance was evaluated according to CCR metric and the results for solving the task a), b) and c) described in Section 4.2.1 are reported in Table 4.6. Also in this case, it is worth to noting that models performances decrease according to an increase in the difficulty of the task. However, VE performance outperforms that of all single models in every task, except for the VGG-16 in task b which are comparable.

Validation by human annotator

The VE model was validated with a blind test performed by the same operator who annotated the employed dataset. A subset of 18 test images in which the model predictions were different from the ground-truth (GT) classes was selected. The operator evaluated the images without knowing the grade he had previously assigned and the

Table 4.6.: Classification performance (CCR) of each single model and VE method for solving task a), b) and c). Best model for each task are reported in bold.

Input	Model	CCR	CCR VE
Task a			
RGB	VGG-16	0.821	**0.838**
Ridge	VGG-16	0.773	
Ridge	SVM	0.788	
Task b			
RGB	VGG-16	**0.762**	0.760
Ridge	VGG-16	0.723	
Ridge	SVM	0.745	
Task c			
RGB	VGG-16	0.481	**0.509**
Ridge	VGG-16	0.446	
Ridge	SVM	0.463	

prediction of the VE model. The validation results are reported in Table 4.7. It is interesting to note that only in 5 cases the operator repeated the same classification, proving that the task is very challenging even for an expert human eye. Moreover, in 4 cases out of 18 the new classification corresponds to the same one provided by the model.

Table 4.7.: Result of the blind test. Red rows: cases where validation class is equal to the test class (GT). Green rows: cases where validation class is equal to the class predicted by the VE model.

ID stock	GT	VE prediction	Validation
1865	3^+	3	4^-
1875	1	2	2
352	2	1	2^+
242	3^+	3	3^+
586	3^+	3	3^+
27	3^+	4	4^-
220	3^+	3	3
282	2	1	2
1650	1	2	2
267	1	2	2^-
644	3	1	3^+
201	1	3	3^-
54	3^+	4	3^+
1887	1	3	2
3	3^+	4	4
7	3^+	4	4^-
241	3^+	3	4^-
276	2	1	2

4.4. Ordinal Deep Learning approach

As described in Section 4.1, the collected AQC dataset implies a natural order between classes, because the more rich and fancy is the veining pattern, the higher is the quality class for the item. Following the previous results, approaching this problem with a nominal DL classification method (which does not exploit class order) causes: i) a substantial drop in accuracy performance as the number of classes to be considered is higher and ii) an increase in misclassification errors even between widely distant classes, which represents the main fault from the industrial production perspective. Considering the ordinal nature of the problem, these issues can be addressed by overcoming the limitation of the nominal approach, in which the classes are not arranged in an appropriate ordered scale, by exploiting the gradual rank of the dataset classes with specific methodologies for ordinal classification. This impression has been supported by the fact that exploiting an ordinal loss (i.e. OCCE) allowed to achieve slightly higher performances (see Table 4.4).

This section introduces a DL ordinal methodology for the AQC classification. Differently from other deep ordinal methods, we combined the standard Categorical Cross-Entropy (CCE) with the Cumulative Link Model (CLM) and we imposed the ordinal constraint via the thresholds and slope parameters.

4.4.1. CLM VGG-16 architecture

In this section, we present the proposed deep ordinal model, which consists of convolutional modules for extracting feature maps and an ordinal classification module (see Figure 4.14). The main aspect of the ordinal module is the integration of the CLM in the output layer, parameterized by slope and thresholds, for encoding the ordinal nature of the label.

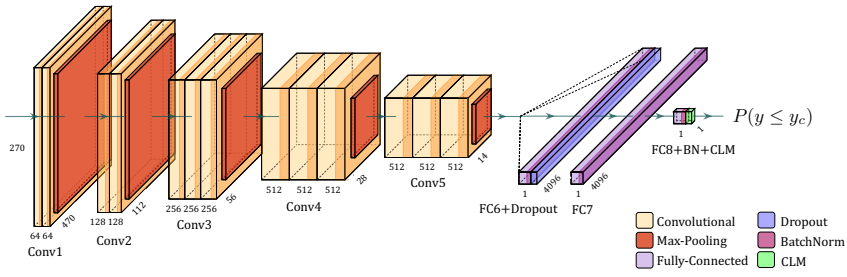


Figure 4.14.: The proposed CLM VGG-16 architecture, which consists of convolutional layers for features extraction and an ordinal head based on CLM.

Feature extractor

We adopted as feature extractor the convolutional part of VGG-16 CNN [136] with 13 convolutional layers. Each of the 5 convolutional blocks has filters with a 3×3 pixels receptive field and is followed by a ReLU activation function. For CNN-parameter dimensionality reduction, max-pooling layers are used after 2 convolutional layers for the first 2 convolutional blocks and after 3 convolutional layers for the other blocks. The activation of the last convolutional block is used as the embedded features ($F \in \mathbb{R}^K$) learned from the feature extractor and is then fed to the ordinal classification head, which computes the output decision of the model. We let $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^K$ the input space and $y \in \mathcal{Y} = \{y_1, y_2, \ldots, y_Q\}$ the output space of Q different ordinal classes defined for the problem.

Ordinal classification module

The output of the convolutional part of the CNN is fed to a sequence of 2 Fully Connected (FC) layers, followed by the output layer. Dropout regularization layer was inserted between the first and the second FC layer with a rate of 0.3. The dropout rate was chosen in the validation stage (see Table 4.8). A batch normalization layer was added in order to stabilize the learning process and reduce the number of training epochs. The last FC layer has only one neuron as it provides the model projection in a 1-dimensional space: its value is used to classify the sample into the corresponding class according to the threshold model. In fact, one common approach to address ordinal classification problems taking into account the ordinal information is to use threshold-based models. Inspiring from [93], the threshold-based approach we adopt in the output layer of the CNN is the Cumulative Link Model (CLM).

CLM [141] are one type of thresholds models which try to predict the probability for each of the categories accounting for the order information implicit to the problem using a set of thresholds that separate different categories and a projection obtained from the input data. Concretely, these models create a 1-dimensional linear projection from the input data, which can be denoted as $f(\mathbf{x}) \in \mathbb{R}$. This 1-D space is divided in Q segments by using a set of thresholds which can be conveniently defined to suit the classes distribution of the current ordinal problem. However, in literature work they are often learned from the training data instead of setting them manually. Thus, the threshold vector can be defined as $\boldsymbol{t} = \{t_0, t_1, ..., t_Q\}$. To divide the output space properly, these thresholds should be in ascending order, satisfying the expression $t_0 < t_1 < ... < t_Q$. The first threshold is always $-\infty$ and the last one $+\infty$.

In the CLM formulation [142], the class order is enforced by the following latent constraint:

$$f^{-1}P(y \preceq y_q|\mathbf{x}) = t_q - f(\mathbf{x}) \tag{4.4}$$

where $q = 1, \ldots, Q-1$, $f^{-1} : [0,1] \rightarrow \infty$ is a monotonic function (inverse link function) and t_q is the threshold defined for class y_q. Hence, the class y_q is predicted if and only if $f(\mathbf{x}) \in [t_{q-1}, t_q]$. It is worth noting that the function f is learned from the training data.

We integrated in the output layer of the architecture different forms of CLM exploring many link functions [143]:

- $\text{logit}(p) = \log \frac{p}{1-p}$,
- $\text{probit}(p) = \Phi^{-1}(p)$,
- $\text{cloglog}(p) = \log(-\log(1-p))$,

where $p = P(y \preceq y_q|\mathbf{x})$, $\Phi = \frac{1}{2}\left(1 + \text{erf}\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)\right)$ is the standard normal cumulative distribution function and $\text{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt$ is the Gauss error function.

4.4.2. Setting the slope and thresholds parameters

CLMs are highly influenced by the right choice of thresholds and slope. The thresholds represent the cutting point between adjacent ordinal classes, while the slope controls the transient of $P(y \preceq y_q|\mathbf{x})$. For instance, a small slope value may lead to a high transient in the CLM that does not enable the ordinal structure modeling (see Fig. 4.15). Following this assumption, the previous defined link functions were defined as follows:

- logit:
$$P(y \preceq y_q|\mathbf{x}) = \frac{1}{1 + e^{-s(t_q - f(\mathbf{x}))}} \tag{4.5}$$
- probit:
$$P(y \preceq y_q|\mathbf{x}) = \int_{-\infty}^{t_q - f(\mathbf{x})} \frac{s}{\sqrt{2\pi}} e^{\frac{1}{2}x^2} dx \tag{4.6}$$
- clog-log:
$$P(y \preceq y_q|\mathbf{x}) = 1 - e^{-e^{s(t_q - f(\mathbf{x}))}} \tag{4.7}$$

where s controls the slope of the CLM. Notice how the introduction of the slope represents one of the main contributions of the proposed work to control the transient between each monotonic link function with the purpose to be adapted according to the specific ordinal problem. We have explored different formulations for the optimization of the thresholds ($\mathbf{t} = \{t_1, t_2, \ldots, t_{Q-2}, t_{Q-1}\}$) and the slope ($s$):

- A): learning the slope s and the thresholds $\mathbf{t}$ from data;
- B): preliminary fixing the values of the slope s and the thresholds $\mathbf{t}$;

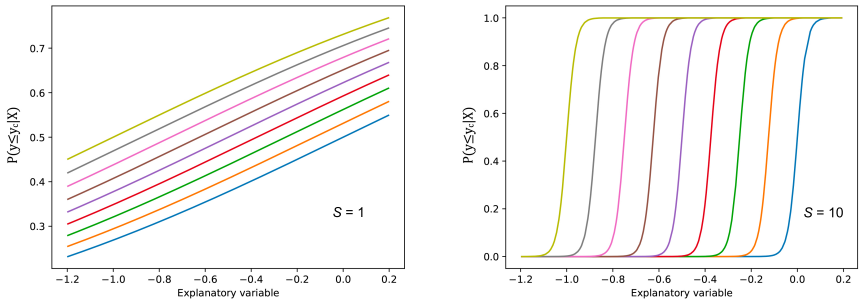


Figure 4.15.: The effect of the slope parameter s regularization in logit link function for defining the $C-1$ thresholds.

- C): preliminary fixing the values of the slope s and learning the thresholds $\mathbf{t}$ from data.

In formulation A), both the slope s and the thresholds are learned during the training process. In particular, the threshold are learned from the following equation:

$$t_q = t_1 + \sum_{i=2}^{Q-1} \gamma_i^2, \tag{4.8}$$

where t_1 is learned to obtain the first threshold, γ is learned to obtain the other thresholds and Q is the number of classes. This formulation for the thresholds ensures that the constraints $t_1 \leq t_2 \leq \cdots \leq t_{Q-1}$ are fulfilled, which is needed for obtaining increasing $P(y \preceq y_q|\mathbf{x})$ with q.

In formulation B), rather than learning the parameter s in the training stage, we have tuned this parameter in the validation stage. Moreover, the imbalanced setting of the ordinal classes is taken into account by fixing the thresholds instead of learning them during the training stage. In particular, we set the thresholds according to the prior probability of each class as follows:

$$t_1 = \frac{\sum_{i=1}^{N} 1_{y=y_1}}{N}, \tag{4.9}$$

$$\gamma_q = \sqrt{P(y = y_q|\mathbf{x})} = \sqrt{\frac{\sum_{i=1}^{N} 1_{y=y_q}}{N}}, \tag{4.10}$$

where t_1 is the value of the first threshold related to the prior probability of the first class, γ_q is the vector of the prior probabilities $P(y = y_q|\mathbf{x})$ associated to each class $q = \{2, \ldots, Q-1\}$ and N is the total number of training points.

In the hybrid formulation C), only the thresholds are learnable parameters while the slope is tuned in the validation stage.

Loss function

The loss function was defined in terms of standard Categorical Cross-Entropy (CCE) as follows:

$$L(\hat{y}, y) = -\sum_{i=1}^{Q} y_i \, log(\hat{y_i}), \tag{4.11}$$

where Q is the number of classes indicating the output size, $\hat{y_i}$ is the i-th scalar value in the model output and y_i is the corresponding target value.

4.4.3. Performance evaluation and Results

Experimental comparisons

It is worth noting that the goal here is to predict the aesthetic quality classes of the rifle models in the most challenging task, i.e. considering all the quality classes (task c, Section 4.2.1). The second version of the collected dataset was employed (Table 4.1). We decided to perform experimental comparisons with respect to baseline nominal VGG-16 [136] and other state-of-the-art ordinal DL methodologies, including ordinal binary decomposition VGG-16 [72] and CLM VGG-16 with weight kappa loss [93]. Figure 4.16 shows the architectures of the state-of-the-art methodologies employed for comparisons.

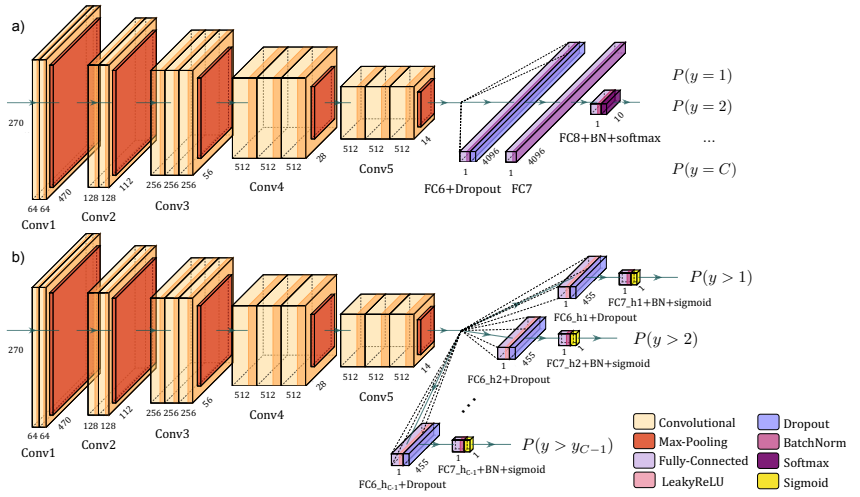


Figure 4.16.: The other state-of-the-art architectures: a) baseline nominal VGG-16 and b) ordinal binary decomposition VGG-16.

Nominal VGG-16

In the nominal classification, the VGG-16 model presents the classic architecture,

where the convolutional part is followed by 3 FC layers and the last one has dimension Q as the number of class labels, as described in Section 4.2.2. The output of this last FC layer is fed to a softmax activation function which maps the output of the CNN model into a set of probabilities belonging to each class. The loss function is the CCE loss, as defined in 4.4.2.

Ordinal Binary Decomposition VGG-16

The Ordinal Binary Decomposition (OBD) is an ordinal approach that consists of decomposing the ordinal problem into a set of $Q-1$ binary problems, where each problem q must determine if $y > y_q$ conditioned to $1 \leq q < Q$. Following the implementation in [72], the convolutional part of the VGG-16 is the input of multiple FC blocks, all of the same dimension. Each block consists of a FC layer, followed by a Leaky ReLU activation function and dropout layer. A final output layer computes the final classification given by the model solving an individual binary classification subproblem. The output of each of the $Q-1$ FC blocks has a sigmoid activation function representing the probability $o_k = P(\mathbf{y} \succ y_k|\mathbf{x}) \in (0,1)$.

The adopted loss functions include MSE defined as follows:

$$L(\hat{y}, y) = \frac{1}{Q-1} \sum_{k=1}^{Q-1} (y_k - \hat{y_k})^2, \tag{4.12}$$

and MAE:

$$L(\hat{y}, y) = \frac{1}{Q-1} \sum_{k=1}^{Q-1} \|y_k - \hat{y_k}\|, \tag{4.13}$$

where Q is the number of classes indicating the output size, $\hat{y_k}$ is the predicted probability of the model output to be greater than y_k and y_k is the corresponding target value that is equal to 1 when $y_i = y_q$ and 0 otherwise.

Cumulative Link Model VGG-16 with Quadratic Weighted Kappa loss

Differently from our approach, in the work made by [93] the CLM structure in the output layer is combined with the continuous version of the Quadratic Weighted Kappa (QWK) loss function. We employed the QWK according to [144] as follows:

$$QWK = 1 - \frac{\sum_{i,j}^{N} \omega_{i,j} O_{i,j}}{\sum_{i,j}^{N} \omega_{i,j} E_{i,j}}, \tag{4.14}$$

where N is the number of training data, N_i is the number of samples for each i-th class, ω is the penalization matrix, O is the confusion matrix, $E_{ij} = \frac{O_{i\bullet} O_{\bullet j}}{N}$, $O_{i\bullet}$ is the sum of the i-th row and $O_{\bullet j}$ is the sum of the j-th column. In our experimental comparisons, linear weights ($\omega_{i,j} = \frac{(i-j)}{(C-1)}$, $\omega_{i,j} \in [0,1]$) and quadratic weights ($\omega_{i,j} = \frac{(i-j)^2}{(C-1)^2}$, $\omega_{i,j} \in [0,1]$) are considered.

Experimental design

A transfer learning approach was used to fine-tune the networks on ImageNet pre-trained weights, in order to reduce computational time while improving the generalization performance [145]. For this reason, all the convolutional layers were frozen. As a preprocessing step, the mean value was removed for each image.

The dataset was split by a stratified holdout procedure, i.e. using 60% of images as training, 20% as validation and 20% as a test. Images belonging to the same shotgun (front and back) were maintained in the same set, ensuring that the model may be able to generalize across different unseen shotgun stocks.

In order to cope with the small dimension of the dataset and the slight unbalance of the classes, a balanced data augmentation strategy was performed on the fly on all the training set samples, applying a horizontal flip to original images. In this process, we ensure that, during training, the number of samples per class follows a uniform distribution, performing an oversampling of the minority classes.

We adopted Adam as optimizer and we explored the best batch size, the initial learning rate and dropout rate as network hyperparameters (see Table 4.8). These network hyperparameters together with the slope parameter for formulations B) and C) were tuned in a separate validation set using a grid-search approach. The number of training epochs was set to 50 while adopting the early stopping strategy with the patience of 10 epochs monitoring validation loss.

All the experiments were performed using TensorFlow 2.0 and Keras 2.3.1 frameworks on Intel Core i7-4790 CPU 3.60GHz with 16GB of RAM and NVIDIA GeForce GTX 970. All the code used in the experiments and the employed dataset is available in a public repository [1].

Table 4.8.: Network hyperparameters and Cumulative Link Model parameters explored in the validation set.

CLM Parameters	**Values**
Slope	10, 50, 100
Netowork Hyperparameters	
Dropout rate	0.1, 0.3, 0.5
Batch size	8, 16, 32
Learning rate	10^{-4}, 10^{-3}, 10^{-2}

Evaluation metrics

Both nominal and ordinal metrics were considered to provide quantitative performance results of the proposed classification models. Regarding the former, the assess-

[1] https://github.com/rosati1392/AQC_Ordinal.git

ment of the DL classification task was performed according to the following metrics:

- Correct Classification Rate (CCR) or *accuracy*, indicating the percentage of correctly classified samples, is the most standard metric for evaluating classification models, also defined as follows:

$$CCR = \frac{1}{N}\sum_{i=1}^{N} 1\{\hat{y}_i = y_i\}, \tag{4.15}$$

  where N denotes the number of test samples, y_i is the class label for sample x_i and $\hat{y}_i$ is the predicted label for sample x_i;

- Top-2 CCR and Top-3 CCR, which are the accuracy where true class matches with any one of the two or three, respectively, most probable classes predicted by the model;

- Minimum Sensitivity (MS)[146], which expresses the lowest percentage of samples correctly predicted to belong to a certain class:

$$MS = \min\left\{ S_c = \frac{O_{qq}}{O_{q\bullet}}, q = 1, ..., Q \right\}, \tag{4.16}$$

  where O is the confusion matrix, Q is the number of classes and S_q is the sensitivity computed for the class q;

- other standard classification metrics such as Precision, Recall and F1_Score, defined as in Section 3.1.3.

In our application context, standard nominal classification metrics may not be significantly representative. For instance, CCR presents two main problems:

- when in presence of class imbalance, it can become an unreliable measure of model performance, as it can be trivially increased by assigning all patterns to the majority class;

- all the prediction mistakes are equally penalized, without considering how much is the deviation from the ground-truth (according to the ordinal scale).

Taking into account the ordinal nature of the AQC task, ordinal metrics are potentially more relevant for evaluating the defined ordinal classification task. The following metrics were considered:

- Quadratic Weighted Kappa (QWK), which is a relevant metric for ordinal problems as it gives a higher weight to the errors that are further from the correct class [144]. The continuous formulation of QWK has been reported (for the results, the values reported are generally those from discrete QWK, while the continuous version is used only for the state-of-the-art experimental comparison in the training process [93]) according to Equation 4.14;

- 1-off accuracy, which indicates that the predicted label is off at most by 1 adjacent class from the ground-truth one;
- MAE, which is the average absolute deviation of the prediction from the ground-truth, defined as:

$$MAE = \frac{1}{N} \sum_{i,j=1}^{Q} \|i - j\| O_{ij}, \tag{4.17}$$

where N is the number of test samples, Q is the number of classes and O is the confusion matrix.

Classification performance

The predictive performance of the proposed approach was provided by tuning the network hyperparameters. For formulation A the CLM parameters were learned in the training set while for formulations B and C the slope was tuned in a separate validation set and kept fixed during the training stage. Table 4.9 shows the predictive performance of the proposed approach (in terms of QWK and MS) for each formulation and for each final activation. The adoption of these two metrics is related to the aim of our classification task: we want to maximize the model performance in the ordinal problem while being consistent in prediction among all the dataset classes despite the imbalanced setting.

Table 4.9.: Predictive performance on the test set of the proposed approach for each formulation and for each final CLM activation. In bold, the experiment that leads to the best results both in terms of Quadratic Weight Kappa (QWK) and Minimum Sensitivity (MS) are reported.

Formulations	Final Act	Slope	Thresholds	BS	LR	QWK↑	MS↑
Experiment A	logit	Trainable	Trainable	16	10^-2	0.926	0.128
	probit	Trainable	Trainable	16	10^-2	0.928	0.205
	clog-log	Trainable	Trainable	16	10^-2	0.892	0.179
Experiment B	logit	Fixed	Fixed	16	10^-2	0.846	0.000
	probit	Fixed	Fixed	16	10^-2	-	-
	clog-log	Fixed	Fixed	16	10^-2	-	-
Experiment C	logit	Fixed	Trainable	16	10^-2	**0.937**	0.231
	probit	Fixed	Trainable	16	10^-2	0.922	**0.282**
	clog-log	Fixed	Trainable	16	10^-2	0.927	**0.282**

With respect to these formulations, experiment C achieved the best results both in terms of QWK (with logit as CLM link function) and MS (with probit and clog-log). Notice how in this formulation we fixed and tuned the optimal slope value in the validation set. This procedure provides better generalization performance than fully

learn slope and thresholds in the training procedure. Another relevant aspect is that fixing the thresholds to a preset value does not allow the model to converge for probit and clog-log activations. This highlights that the slope parameter has no effect if the flexibility provided by the threshold model structure, where the threshold of each class is independently adjusted during training, is not guaranteed.

Figure 4.17 shows the test confusion matrices of the proposed approach and the baseline approach (nominal approach). The confusion matrix of the proposed method is more focused on the diagonal, thus penalizing the error among distant AQC classes.

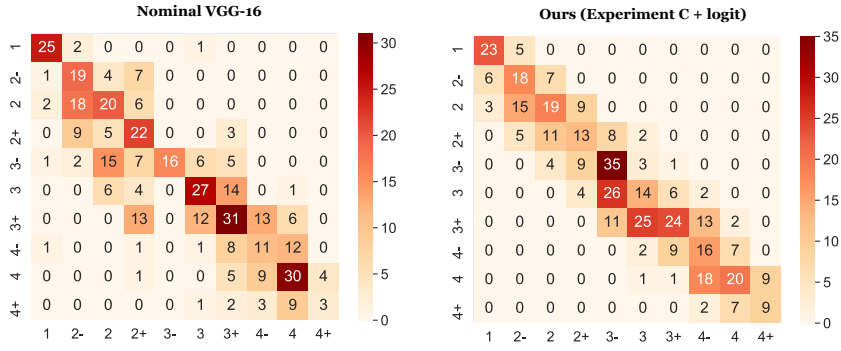


Figure 4.17.: Confusion matrices for nominal and ordinal approaches.

Table 4.10 shows the experimental results of our approach with respect to the baseline approach (nominal approach) and other state-of-the-art deep ordinal methods (OBD VGG-16 [72] and CLM VGG-16 [93]). Experiment C was chosen as the best formulation of our proposed approach. Our proposed deep ordinal model outperforms the nominal approach in terms of QWK, MS, and 1-OFF of about 8%, 68.9%, and 17.4% respectively. It is worth noting that QWK, MS, and 1-OFF represent the most important metrics in order to reduce misclassification errors among distant classes. This requisite fully corresponds to the original company's demand, i.e. the reduction of errors among distant AQC classes. Moreover, the proposed approach overcomes in terms of QWK, MS, and 1-OFF the OBD [72] of about 1,8%, 282% and, 4,3% respectively and CLM [93] of about 0.03%, 37.6%, and 1% respectively.

The comparison with the other ordinal methodologies and the highest values of QWK, MS, and 1-OFF highlighted how the proposed method is more effective to model the ordinal structure of the AQC classes by penalizing the distance of incorrect prediction from the ground truth class.

4.4.4. Model interpretability and Bias mitigation

From the point of view of a domain expert (such as a human operator who is responsible for AQC task), model explanation and interpretability are key points that may

Table 4.10.: Experimental results comparison on the test set in terms of both ordinal and nominal metrics of the proposed approach with respect to baseline nominal VGG-16 model (NOM), ordinal binary decomposition (OBD) implementation for CNN and state-of-the-art cumulative link models (CLM) for deep ordinal classification. The best value for each metric is highlighted in bold. For the final activation, the learnable parameters are specified in parentheses, where "th" stands for thresholds.

Method	**Final Act**	**Loss**	**BS**	**LR**	**QWK↑**	**MS↑**	**MAE↓**	**CCR↑**	**TOP_2↑**	**TOP_3↑**	**1-OFF↑**
NOM	softmax	CCE	16	10^{-2}	0.867	0.167	0.124	0.481	0.731	0.863	0.788
OBD	sigmoid	MSE	16	10^{-2}	0.920	0.000	0.158	0.528	0.783	0.906	0.880
OBD	sigmoid	MAE	16	10^{-2}	0.923	0.000	0.142	**0.554**	**0.802**	0.896	0.887
CLM	logit(th)	QWK	16	10^{-2}	0.924	0.000	0.128	0.424	0.709	0.851	0.875
CLM	probit(th)	QWK	16	10^{-2}	0.929	0.000	0.124	0.429	0.726	0.868	0.908
CLM	clog-log(th)	QWK	16	10^{-2}	0.911	0.000	0.132	0.394	0.670	0.863	0.835
CLM	logit(th)	LWK	16	10^{-2}	0.918	0.000	0.122	0.443	0.715	0.844	0.877
CLM	probit(th)	LWK	16	10^{-2}	0.917	0.000	0.127	0.392	0.698	0.851	0.858
CLM	clog-log(th)	LWK	16	10^{-2}	0.909	0.000	0.135	0.382	0.687	0.849	0.844
CLM	logit(th,slope)	QWK	16	10^{-2}	0.925	0.205	0.113	0.460	0.776	0.911	0.894
CLM	probit(th,slope)	QWK	16	10^{-2}	0.934	0.180	**0.112**	0.467	0.776	0.910	0.915
CLM	clog-log(th,slope)	QWK	16	10^{-2}	0.928	0.051	**0.112**	0.462	0.743	0.858	0.899
Ours	logit(th)	CCE	16	10^{-2}	**0.937**	0.231	0.137	0.455	0.774	**0.927**	**0.925**
Ours	probit(th)	CCE	16	10^{-2}	0.922	**0.282**	0.140	0.434	0.729	0.901	0.881
Ours	clog-log(th)	CCE	16	10^{-2}	0.927	**0.282**	0.137	0.451	0.776	**0.927**	0.889

increase the usefulness and the trustworthiness of the overall DSS. An explanation, specifically tailored to the end-users, on how the DL model achieved the prediction is relevant in order (i) to uncover valuable information that otherwise would have remained hidden within the complexity of the model and (ii) to empower users with powerful new insights.

Starting from this concept, our objective was to encourage the prediction of the proposed model to be as aligned as possible with the human annotation. By designing an ordinal DL methodology, our claim was to penalize large errors (misclassification errors among distant classes) that do not usually happen in human evaluation. After being demonstrated this outcome, we would go further by describing that from one side the model is potentially able to provide new insights on finer-grained wood patterns that can be sometimes unseen by a human operator, and from the other side the model is aligned on what the human operator is checking, thus focusing on the aesthetic quality classification of rifles based on the analysis of wood grains and avoiding the unwanted bias related to the characteristics of the rifle series (see 4.2.4.

This fact is confirmed by exploring the saliency map of the proposed ordinal DL approach, according to the approach proposed by [140]: the extracted saliency maps are constrained to focus on wood grains rather than, for instance, the geometrical edges, as shown in Figure 4.18). Thus, this strategy allowed to further alleviate the bias by separating the two tasks and providing the prediction of quality classes for each rifle macro-series model.

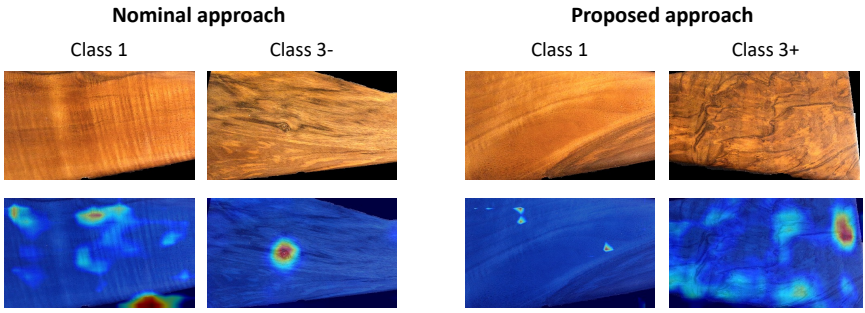


Figure 4.18.: Saliency maps obtained from test images correctly predicted by the nominal and proposed approach. In class 1, it can be seen how the nominal approach is more focused outside the stock, whereas for the proposed approach the map does not show any hot point because veins are not relevant in this class. For the higher classes, notice how the proposed model better focuses on the wood features, following the attention on the pattern of grains.

4.5. Exponential loss regularisation

The described CLM usually achieves good performance when classifying ordinal categories. However, there are some limitations: these kinds of models are affected by the optimal choice of the selected parameters, i.e. the performance of this model is highly influenced by the learned or fixed thresholds. For this reason, the key point was to combine the CLM with a soft labeling approach based on a unimodal regularisation, which is described in Section 4.5.1.

In particular, we have proposed a more flexible exponential regularisation method based on introducing a L_p norm into a previously proposed exponential regularised loss, explained in Section 4.5.2. This loss regularisation is appropriate for ordinal problems where the misclassification errors should be in adjacent classes instead of distant classes, encouraging labels distribution to be soft and unimodal, being centred in the middle of the real class interval.

This approach is useful also for enhancing the classification quality and reflecting possible noise or errors in labeling process. This is especially the case of our AQC task, where the labeling procedure of the QC task may be affected by different sources of error namely the human operator variability, the different operating condition, the different processing steps and the novel and never seen wood aesthetic appearance, as demonstrated by the validation procedure in Section 4.3.2.

4.5.1. Soft labels and unimodal regularisation

Label smoothing [147] is a regularisation technique that is applied to the representation of the labels. When used to train a model, it encourages the classifier to be less confident, giving some probability to the other classes instead of focusing only on the true category. This enhances the robustness of the model in the presence of noisy labels. Label smoothing can be very useful for ordinal problems, where misclassifying a pattern in an adjacent class is more probable than predicting a distant category. The way the label smoothing is performed depends on the problem characteristics and is a way to introduce the ordinal information of the problem into the model. This extra ordinal information usually accelerates the convergence of the model and reduces the number of training examples needed to achieve a good model fitting.

In [148], the authors proposed to sample ordinal smooth labels from a Poisson distribution and a binomial one. Then, they also used an exponential function to obtain soft labels. The Poisson distribution is given by:

$$p_q = P(X = q) = \frac{\lambda^q e^{-\lambda}}{q!}, \tag{4.18}$$

where $X \sim \text{Poisson}(\lambda)$, $q = 0, 1, ..., Q-1$, and $\lambda \in \mathbb{R}^+$. Its mean and variance is determined by the value of its parameter λ. Thus, for some classes, it is not possible to obtain a distribution that is centred in the middle of the class interval while keeping the variance low. For this reason, these kinds of distributions are not very adequate to obtain soft labels with small variance. Therefore, the authors introduced the binomial distribution, which they stated that is more flexible and provides better results. This distribution is given by:

$$p_q = P(X = q) = \binom{Q}{q} p^q (1-p)^{Q-q}, \tag{4.19}$$

where $X \sim \text{Bin}(Q, p)$ and $q = 0, 1, ..., Q-1$. In this case, the binomial distribution has two parameters: Q, or the number of classes, and the probability of the event (belong to a specific class). Note that, even though the mean ($E[x] = Qp$) and the variance ($V[x] = Qp(1-p)$) are determined by different expressions, it is not easy to achieve distributions correctly centred in the middle of the class interval and with a small variance. Finally, they proposed to use a exponential function as a third alternative. This function was defined as follows:

$$f(q) = e^{-|q-y|}, \tag{4.20}$$

where q is the current label and y is the ground truth, both represented as integers in the ordinal scale. However, it provides limited flexibility and, sometimes, when using the softmax function to obtain a probability value from the exponential function, the

probability mass is not sufficiently concentrated in the middle of the interval.

4.5.2. L_p norm exponential regularised cross-entropy loss

The exponential regularised soft labeling in (Eq. 4.20) applies a L_1 norm. In this work, we proposed to sample on a more flexible exponential function based on the introduction of the L_p norm. L_p norms have been used in optimisation algorithms in several fields as a generalisation of L_2 and L_1 norms. Basically, in this proposal there is an extra tunable parameter that can be adjusted by the learning algorithm. In this way, a more flexible L_p normalised exponential function can be defined as:

$$f_p(q) = e^{-|q-y|^p}, \quad 1 \leq p \leq 2, \tag{4.21}$$

where the p parameter can be tweaked manually or cross-validated. Hence, the p parameter controls how much a pattern is penalised when it is classified in class q and its real class is y (both represented as integers). Lower values of p mean that less relevance is given to that error. In this context, cost functions other than the Squared Euclidean norm (L_2) or the Manhattan Distance norm (L_1) might provide better results due to its enhanced flexibility. The range of possible values for the aforementioned parameter should be restricted to the interval of real numbers to respect the formal definition of a geometrical norm, i.e. $p \in [1, 2]$.

This regularisation of the labels is applied as an alternative to the standard categorical cross-entropy loss function:

$$L = -\sum_{q=1}^{Q} h(q)[-\log p(y = y_q|\mathbf{x})], \tag{4.22}$$

where $h(q) = \delta_{q,y}$, q and y are the predicted and ground truth classes, respectively, represented as integers, and $\delta_{q,y}$ is the Dirac delta, which equals to 1 for $q = y$, and 0 otherwise. $h(q)$ can be replaced with a soft version $h'(q)$, which can be obtained by applying the aforementioned exponential function. In this way, the standard definition of the loss function can be replaced by:

$$L = \sum_{q=1}^{Q} h'(q)[-\log p(y = y_q|\mathbf{x})], \tag{4.23}$$

where $h'(q) = (1 - \eta)\delta_{q,y} + \eta f_p(q)$ and η is a parameter that ranges from 0 to 1 controlling the smoothness of the labels. When $\eta = 0$, no smooth factor is applied. On the other side, when $\eta = 1$, the labels are completely smooth and the standard labels are not used.

Figure 4.19 shows classes distributions for the proposed exponential function along with the Poisson, binomial and standard exponential distributions that were described

before. The colour represents the true class of the pattern, while the x-axis represents the class being examined and the y-axis represents the soft label applied. In the case

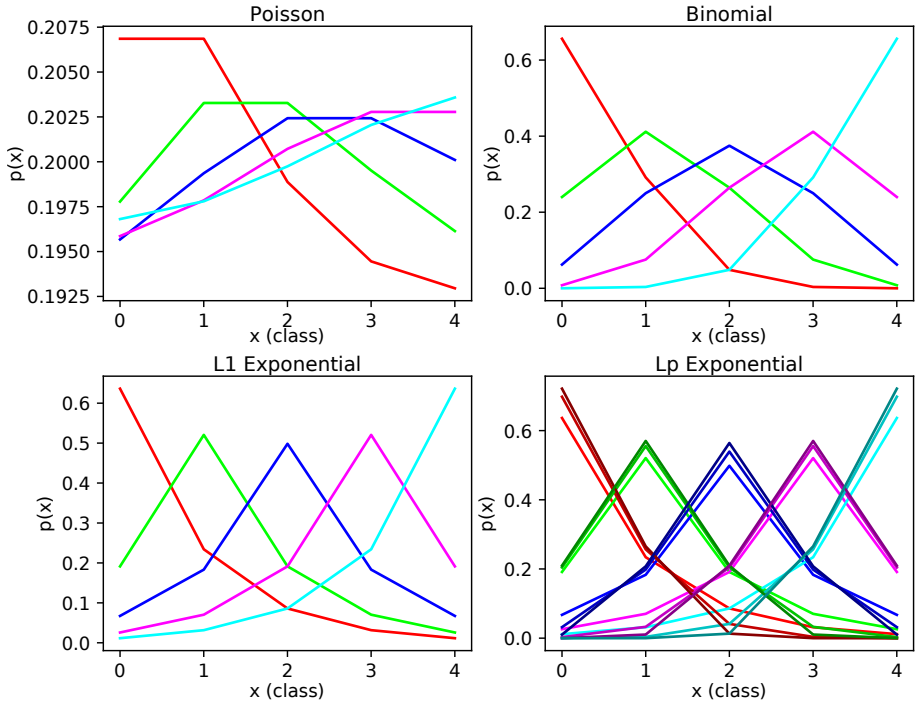


Figure 4.19.: Different types of distributions for a problem with 5 classes. The x-axis represents the evaluated class, the y axis represents the value of the smooth label given for the class and the colours are associated to the true class (red: 0, green: 1, blue: 2, pink: 3, and cyan: 4). Thus, each line represents the probability distribution for one real label. In the L_p Exponential plot, different intensities of the colour show different values of the p parameter (1.0, 1.5 and 2.0, where higher intensity means higher value).

of the L_p exponential, the distributions obtained with the lower and upper bounds and an intermediate value of the parameter is also considered. Therefore, for each class, the distributions for $p \in \{1.0, 1.5, 2.0\}$ are shown. Any other distribution that can be obtained by tweaking this parameter will be in-between the lower and upper bounds distributions. $p = 1.0$ is represented with the darkest colours while $p = 2.0$ has the most intense colours in the plot. As mentioned before, the L_1 exponential is equivalent to using the L_p exponential with $p = 1$, as the latter is a more general and flexible version of the exponential function.

4.5.3. Classification model

The proposed method was applied to the same architecture (i.e. VGG-16) with the same pre-trained weights of previously described methodologies in order to achieve fair comparisons for solving our AQC task.

For the convolutional part of the model, the pre-trained ImageNet weights are used and the parameters of these layers are set to be non-learnable, adjusting only the top part of the model. This method accelerates the convergence and reduces the computational time significantly. In the fully-connected part of the model, a 50% dropout is applied before two dense layers with 4096 units and ReLU activation. To sum up, the whole model comprises a total of 266M of parameters where 251M are trainable, and the remaining are fixed parameters that belong to the convolutional layers that use the pre-trained ImageNet weights.

The function used in the output layer is the softmax, for the baseline experiments, and the CLM with different link types for the rest of the experiments.

4.5.4. Performance evaluation and Results

Experimental procedure

The same dataset images as describe in Section 4.4.3 were used. The performed experiments follow a 10-fold cross-validation scheme that is repeated 3 times (with 3 different seeds) to achieve 30 executions. Each of these partitions defines different and non-overlapping train and test splits with 90% of data for training and 10% for test. Also, for each of the training partitions, a holdout is performed to divide this whole set into train and validation. In this way, the validation set can be used to lead the early-stopping strategy that stops the training process when the validation loss has not improved for several epochs. Moreover, validation metrics are used to adjust the hyperparameters.

Training data is fed into the model during the training process using a generator that performs data augmentation based on random horizontal flips and also creates balanced batches regarding the different classes of the problem. For the problem considered, it is important to use balanced batches as the dataset is fairly imbalanced.

The Adam [149] algorithm is used to optimise the model, and the learning rate is fixed to 0.01 for the whole learning process. During the training process, the training data is processed in batches of size 16 and the learning stage is run for a maximum of 50 epochs.

For comparison purposes, different sets of experiments are performed:

1. Baseline. Softmax in the output layer and standard categorical cross-entropy (CCE) as loss function.

2. CLM with logit, probit and complementary log-log links in the output layer, and Poisson regularised CCE.

3. CLM with logit, probit and complementary log-log links in the output layer, and binomial regularised CCE.

4. CLM with logit, probit and complementary log-log links in the output layer, and exponential regularised CCE.

5. Proposed method. CLM with logit, probit and complementary log-log links, and L_p exponential regularised CCE.

As shown in Section 4.5.2, the proposed loss function has an hyperparameter (p) that must be adjusted. In this work, this hyperparameter is adjusted by cross-validation taking into account the validation Quadratic Weighted Kappa (QWK) [144] metric for each fold and seed. Therefore, different values of the p parameter can be obtained for different folds and seeds, as we noticed that the optimal value of p depends on the data considered. Thus, the experimental procedure to adjust this hyperparameter is described in Algorithm 1.

Algorithm 1 p parameter cross-validation procedure.

foreach *seed* **do**
  Split whole dataset in 10 folds that will be used for training and test. **foreach** p **do**
    foreach *fold* **do**
      Split all the data not included in the current $fold$ into 80% for training and 20% for validation.
      Train for the number of epochs determined by early stopping and evaluate on the validation set.
    end
  end
end
foreach *seed* **do**
  foreach *fold* **do**
    Find p value that achieved the best validation QWK for this fold and seed.
    Evaluate on the test set (current $fold$) with the best validation p value.
  end
end

The η parameter has been fixed to $\eta = 1.0$, which achieves fully soft labels, instead of combining the standard labels with the soft labels obtained through the unimodal distributions.

Evaluation metrics

We considered three metrics that are appropriate for ordinal problems and imbalanced datasets: QWK [144], MS [146] and MAE [146]. The metrics are defined as reported in Section 4.4.3.

Classification performance

Table 4.11 contains the mean results for 30 executions of each of the experiments. For the experiments where the L_p-exponential regularised categorical cross-entropy is used, the value of the p parameter was adjusted through cross-validation for each fold and seed, and the mean value is displayed under the Mean p column.

Table 4.11.: Mean results for 30 executions of each of the alternatives on the test set. The best value for each metric is highlighted with bold font face.

Loss	Output	N	Mean p	QWK↑	MS↑	MAE↓
CCE	Softmax	30	-	0.88712	0.14839	0.76572
CCE-Poisson	CLM Logit	30	-	0.78042	0.00000	1.73867
CCE-Poisson	CLM Probit	30	-	0.77503	0.00000	1.78581
CCE-Poisson	CLM CLogLog	30	-	0.77921	0.00000	1.65733
CCE-Binomial	CLM Logit	30	-	0.92068	0.19775	**0.70338**
CCE-Binomial	CLM Probit	30	-	0.91929	0.20380	0.71846
CCE-Binomial	CLM CLogLog	30	-	0.89189	0.06812	0.86134
CCE-Exp-L_p	CLM Logit	30	1.61	**0.92391**	**0.21883**	0.70563
CCE-Exp-L_p	CLM Probit	30	1.70	**0.92391**	0.19875	0.71013
CCE-Exp-L_p	CLM CLogLog	30	1.66	0.91368	0.15572	0.77624
CCE-Exp-L_1	CLM Logit	30	-	0.92237	0.18290	0.72664
CCE-Exp-L_1	CLM Probit	30	-	0.91596	0.10117	0.78666
CCE-Exp-L_1	CLM CLogLog	30	-	0.89992	0.04978	0.85809

As can be observed from the results in this Table, the L_p-exponential regularised categorical cross-entropy with the logit link obtained the best result for QWK and MS and the second-best for MAE. Also, the same loss function with the probit link achieved the same result for QWK.

The confusion matrices of the best alternative proposed (CCE-Exp-Lp + CLM Logit) and the baseline method (CCE + Softmax) are shown in Figures 4.20, 4.21 and 4.22. Each figure represents the confusion matrices of each of the 3 seeds considered. Each matrix is obtained by accumulating the confusion matrices of all the 10 folds. From these matrices, it can be observed that the baseline method misclassifies some patterns in distant classes, which implies an important cost for this real problem, while the proposed ordinal method has almost all the errors in the adjacent classes.

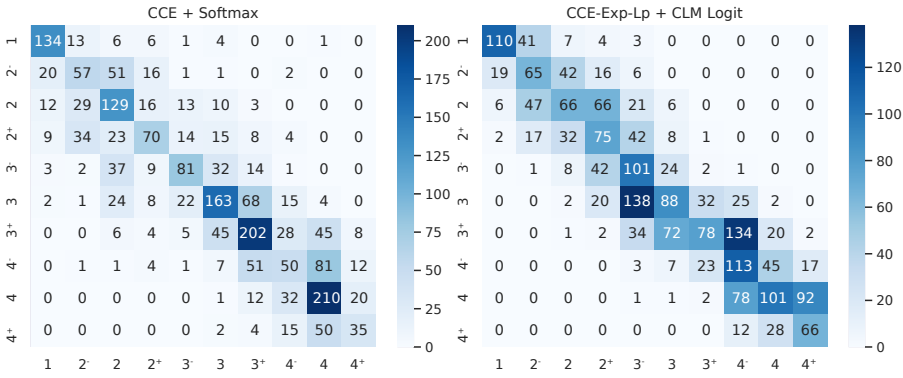


Figure 4.20.: Confusion matrices obtained for the seed 0.

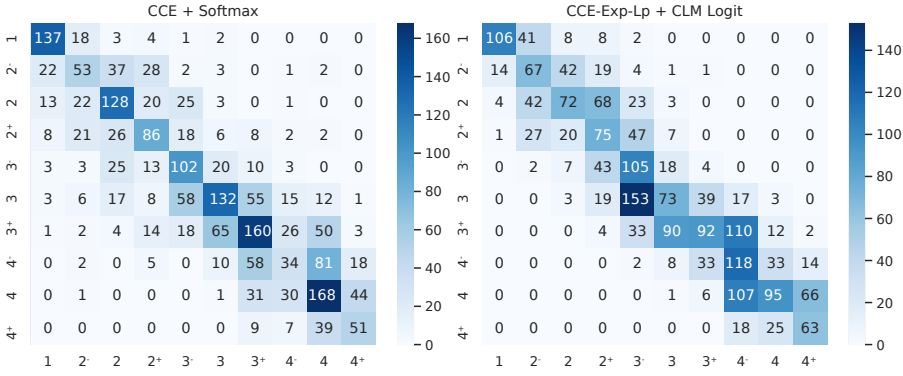


Figure 4.21.: Confusion matrices obtained for the seed 1.

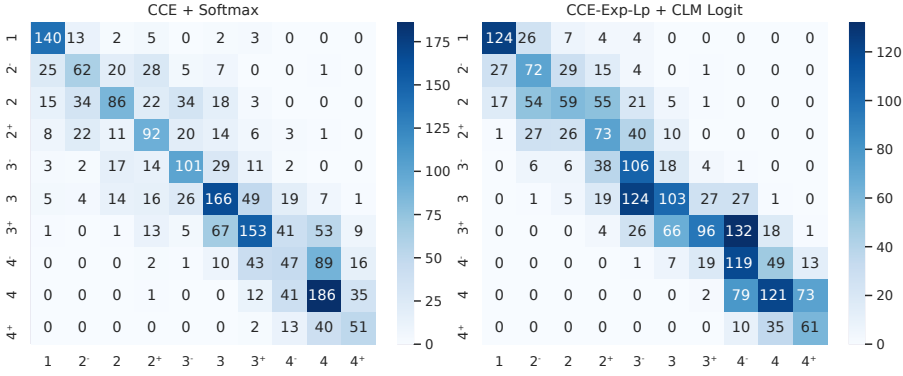


Figure 4.22.: Confusion matrices obtained for the seed 2.

Statistical analysis

A statistical analysis was performed to check whether the proposed alternative provides significantly better results than the baseline and previous proposed methods. To

do that, each of the metrics has been analysed separately.

First, using the QWK metric, a Kolmogorov-Smirnov [150] test has been performed to check if the 30 QWK test values are normally distributed. The test confirmed that the values of this metric follow a normal distribution (p-value< 0.05). After that, a Friedman rank test [151] has been performed to obtain the rank related to each method. The results of this test are shown in Table 4.12.

Table 4.12.: Friedman test results for the QWK metric.

Method	**Rank**
CCE + Softmax (Baseline)	5.43
CCE-Exp-L_p + CLM CLogLog	8.20
CCE-Exp-L_p + CLM Logit	10.87
CCE-Exp-L_p + CLM Probit	**10.90**
CCE-Exp-L_1 + CLM CLogLog	5.77
CCE-Exp-L_1 + CLM Logit	10.17
CCE-Exp-L_1 + CLM Probit	8.73
CCE-Poisson + CLM Probit	2.00
CCE-Poisson + CLM Logit	1.97
CCE-Poisson + CLM CLogLog	2.07
CCE-Binomial + CLM Probit	9.70
CCE-Binomial + CLM Logit	10.13
CCE-Binomial + CLM CLogLog	5.07

The results of the Friedman test show that the proposed L_p Exponential regularised CCE with the probit link achieved the best rank concerning the QWK metric. Also, the same loss with the logit link obtained the second-best rank, which is very close to the first one.

Given that the CCE-Exp-L_p + CLM Logit and the CCE-Exp-L_p + CLM Logit have similar ranks, and the logit link function has better overall results considering all the methods, the L_p Exponential regularised CCE with logit link has been compared with the other methods using a paired sample t-test. The results of this test are shown in Table 4.13. The Paired differences columns show the mean and standard deviation of the differences between both methods indicated in the first column. The t column shows the value of the statistical, the df column indicates the degrees of freedom and, finally the $p - value$ column shows the p-value obtained, which indicates significant differences with a significance level of $\alpha = 0.05$ when it is higher than that value.

As can be observed in Table 4.13, the logit and probit links perform similarly with the L_p exponential regularisation (p-value $= 1.0$). Also, it shows no significant differences with the standard exponential with logit link (p-value $= 0.129$), and the binomial regularisation with probit (p-value $= 0.129$) or logit link (p-value $= 0.314$). However, it performs significantly different than the rest of the methods.

Also, in Table 4.14, the results of the paired t-test comparing the baseline with the

Table 4.13.: Paired sample t-test to compare L_p Exponential regularised CCE + CLM Logit with other methods regarding QWK.

	Paired differences				
Methods	Mean	Std. Dev.	t	df	p-value
Exp-L_p + Logit - Exp-L_p + CLogLog	0.01023	0.00917	6.110	29	< 0.001
Exp-L_p + Logit - Exp-L_p + Probit	-0.00001	0.00676	0.000	29	1.000
Exp-L_p + Logit - CCE + Softmax	0.03679	0.02132	9.454	29	< 0.001
Exp-L_p + Logit - Exp-L_1 + CLogLog	0.02400	0.01069	12.393	29	< 0.001
Exp-L_p + Logit - Exp-L_1 + Logit	0.00154	0.00620	1.562	29	0.129
Exp-L_p + Logit - Exp-L_1 + Probit	0.00795	0.00995	4.377	29	< 0.001
Exp-L_p + Logit - Poisson + Probit	0.14888	0.05688	14.336	29	< 0.001
Exp-L_p + Logit - Poisson + Logit	0.14350	0.05042	15.587	29	< 0.001
Exp-L_p + Logit - Poisson + CLogLog	0.14471	0.06075	13.047	29	< 0.001
Exp-L_p + Logit - Binomial + Probit	0.00484	0.16969	1.562	29	0.129
Exp-L_p + Logit - Binomial + Logit	0.00303	0.01619	1.024	29	0.314
Exp-L_p + Logit - Binomial + CLogLog	0.03202	0.01793	9.782	29	< 0.001

other approaches is shown. These results show that there are significant differences between the baseline and all the other methods (except the binomial regularisation with the complementary log-log link). The method proposed in this work obtained better results than the baseline with all the link functions.

Table 4.14.: Paired sample t-test to compare L_p CCE + Softmax (baseline) with other methods regarding QWK.

	Paired differences				
Methods	Mean	Std. Dev.	t	df	p-value
CCE + Softmax - Exp-L_p + CLogLog	-0.02656	0.02171	-6.702	29	< 0.001
CCE + Softmax - Exp-L_p + Logit	-0.03679	0.02131	-9.454	29	< 0.001
CCE + Softmax - Exp-L_p + Probit	-0.03679	0.02132	-9.453	29	< 0.001
CCE + Softmax - Exp-L_1 + CLogLog	-0.01279	0.02208	-3.173	29	0.004
CCE + Softmax - Exp-L_1 + Logit	-0.03525	0.02157	-8.951	29	< 0.001
CCE + Softmax - Exp-L_1 + Probit	-0.02884	0.02331	-6.776	29	< 0.001
CCE + Softmax - Poisson + Probit	0.11209	0.05466	11.232	29	< 0.001
CCE + Softmax - Poisson + Logit	0.10670	0.05949	9.824	29	< 0.001
CCE + Softmax - Poisson + CLogLog	0.10792	0.05823	10.150	29	< 0.001
CCE + Softmax - Binomial + Probit	-0.03195	0.03031	-5.774	29	< 0.001
CCE + Softmax - Binomial + Logit	-0.03376	0.02759	-6.703	29	< 0.001
CCE + Softmax - Binomial + CLogLog	-0.00477	0.03090	-0.845	29	0.405

The same analysis has been performed accounting for the MS metric results on the test set. The Kolmogorov-Smirnov test reported that the values are distributed following a normal distribution (p-value< 0.05). Therefore, a Friedman rank test has been performed and the results are shown in Table 4.15.

In this case, the L_p exponential regularised CCE loss combined with the CLM with logit link obtained the best rank (10.87). The same loss function with the probit link obtained also high results. After this test, a paired sample t-test has been performed to compare the best alternative according to the ranking with the other methods. The

Table 4.15.: Friedman test results for the MS metric.

Method	**Rank**
CCE + Softmax (Baseline)	8.02
CCE-Exp-L_p + CLM CLogLog	8.43
CCE-Exp-L_p + CLM Logit	**10.87**
CCE-Exp-L_p + CLM Probit	10.10
CCE-Exp-L_1 + CLM CLogLog	4.78
CCE-Exp-L_1 + CLM Logit	9.40
CCE-Exp-L_1 + CLM Probit	6.15
CCE-Poisson + CLM Probit	2.50
CCE-Poisson + CLM Logit	2.50
CCE-Poisson + CLM CLogLog	2.50
CCE-Binomial + CLM Probit	10.30
CCE-Binomial + CLM Logit	10.08
CCE-Binomial + CLM CLogLog	5.37

results of this test are shown in Table 4.16.

Table 4.16.: Paired sample t-test to compare L_p Exponential regularised CCE + CLM Probit with other methods regarding MS.

	Paired differences				
Methods	Mean	Std. Dev.	t	df	p-value
Exp-L_p + Logit - Exp-L_p + CLogLog	0.06311	0.07868	4.393	29	< 0.001
Exp-L_p + Logit - Exp-L_p + Probit	0.02009	0.05444	2.021	29	0.049
Exp-L_p + Logit - CCE + Softmax	0.07044	0.11548	3.341	29	0.002
Exp-L_p + Logit - Exp-L_1 + CLogLog	0.16905	0.07357	12.585	29	< 0.001
Exp-L_p + Logit - Exp-L_1 + Logit	0.03593	0.05634	3.494	29	0.002
Exp-L_p + Logit - Exp-L_1 + Probit	0.11766	0.07202	8.949	29	< 0.001
Exp-L_p + Logit - Binomial + Probit	0.02193	0.10174	1.181	29	0.247
Exp-L_p + Logit - Binomial + Logit	0.01418	0.10152	0.765	29	0.450
Exp-L_p + Logit - Binomial + CLogLog	0.15071	0.08263	9.990	29	< 0.001

The results related to the Poisson regularisation have been omitted in this table since all the MS results for this method obtained a value of 0. As can be observed in Table 4.16, the L_p exponential regularised loss with logit link resulted significantly better than most of the other alternatives. Only the binomial regularised loss with probit and logit links obtained similar results. Another paired t-test was performed to compare the baseline with the rest of the methods. The results of this test shown that the proposed method obtained better results than the baseline and is significantly better when using the logit (p-value $= 0.002$) or probit (p-value $= 0.036$) links.

Finally, the results concerning the MAE metric have been analysed in the same way that in the previous analyses. First, a Kolmogorov-Smirnov test has been used to confirm that the results are normally distributed (p-value< 0.05). Then, a Friedman rank test has been performed to obtain a ranking of the methods regarding the MAE

metric. The results are shown in Table 4.17.

Table 4.17.: Friedman test results for the MAE metric.

Method	**Rank**
CCE + Softmax (Baseline)	5.75
CCE-Exp-L_p + CLM CLogLog	6.48
CCE-Exp-L_p + CLM Logit	3.23
CCE-Exp-L_p + CLM Probit	3.48
CCE-Exp-L_1 + CLM CLogLog	9.13
CCE-Exp-L_1 + CLM Logit	4.13
CCE-Exp-L_1 + CLM Probit	7.20
CCE-Poisson + CLM Probit	12.17
CCE-Poisson + CLM Logit	12.17
CCE-Poisson + CLM CLogLog	11.67
CCE-Binomial + CLM Probit	3.87
CCE-Binomial + CLM Logit	**2.90**
CCE-Binomial + CLM CLogLog	8.82

The test reported that the best method is the one that uses the Binomial regularisation combined with the CLM with logit link. However, the method that uses the L_p exponential regularisation with the logit link obtained very close results. In these terms, to compare this method with rest of alternatives, a paired sample t-test was performed. The results of the aforementioned test are shown in Table 4.18.

Table 4.18.: Paired sample t-test to compare L_p Exponential regularised CCE + CLM Probit with other methods regarding MAE.

	Paired differences				
Methods	Mean	Std. Dev.	t	df	p-value
Exp-L_p + Logit - Exp-L_p + CLogLog	-0.07061	0.06233	-6.205	29	< 0.001
Exp-L_p + Logit - Exp-L_p + Probit	-0.00450	0.04988	-0.494	29	0.625
Exp-L_p + Logit - CCE + Softmax	-0.06009	0.09471	-3.475	29	0.002
Exp-L_p + Logit - Exp-L_1 + CLogLog	-0.15246	0.06992	-11.943	29	< 0.001
Exp-L_p + Logit - Exp-L_1 + Logit	-0.02101	0.03605	-3.192	29	0.003
Exp-L_p + Logit - Exp-L_1 + Probit	-0.08103	0.06855	-6.475	29	< 0.001
Exp-L_p + Logit - Poisson + Probit	-1.08018	0.32809	-18.033	29	< 0.001
Exp-L_p + Logit - Poisson + Logit	-1.03304	0.28901	-19.578	29	< 0.001
Exp-L_p + Logit - Poisson + CLogLog	-0.95170	0.34386	-15.159	29	< 0.001
Exp-L_p + Logit - Binomial + Probit	-0.01324	0.09093	-0.798	29	0.432
Exp-L_p + Logit - Binomial + Logit	0.00266	0.07882	0.185	29	0.854
Exp-L_p + Logit - Binomial + CLogLog	-0.15571	0.07550	-11.296	29	< 0.001

The analysed method shows significant differences with almost all the other alternatives. However, the L_p exponential regularisation with the probit link and the binomial regularisation with probit or logit link are not statistically different. In the same way we did for the other metrics, another paired t-test was performed to compare the base-

line with the other approaches. The test reported significant differences between the baseline and the proposed method when using the logit (p-value $= 0.002$) or probit (p-value $= 0.001$) links.

To sum up, the Exp-L_p + Logit obtained the best overall results for most of the metrics. It is the best in terms of QWK and MS metrics, showing significant differences for the latter. Also, it is better than the standard exponential regularisation (L1) in all the three metrics and provides significant improvements for MS and MAE. Finally, the Binomial regularisation with logit link achieved slightly better results than the Exp-L_p with the same link concerning the MAE metric, but there are no significant differences between these methods. Nevertheless, it is worth mentioning that the L_p exponential improved not only the baseline results but also the results of the standard exponential function.

4.6. Hierarchical Deep Learning framework

As described in Section 4.1, apart from following an ordinal relation, the 10 categories of our AQC problem are grouped in four macro classes: 1, 2, 3, and 4. The macro classes can be easily classified by an expert. However, each of these macro classes contains several micro labels ($-$, $\cdot$, $+$) which are harder to classify. Figure 4.23 shows the hierarchical structure of the classes in a more detailed manner.

Hence, the aim of this work was to propose a hierarchical approach that simplifies, generalises and automatises the AQC task by using multiple ordinal CNN models to predict hierarchically the final label in two steps: one for the macro label and one for the micro. A similar approach was introduced in a previous work [152], where, in a medical application, an initial prediction was done to obtain a positive or negative result, and, a posterior classification determined different grades when the first result was positive. However, in this work, instead of using a binary classifier for the first step, an ordinal classifier with four classes is used. For the second step, three different ordinal classifiers are employed to obtain the micro label. A combination of the labels obtained in both steps results in the final label.

The description of the hierarchical method is divided into two parts: in Section 4.6.1 the ensemble architecture and the method to combine the predictions obtained from the individuals models, and in Section 4.6.2 the architecture of the deep network that is used for each individual model that the hierarchical structure contains.

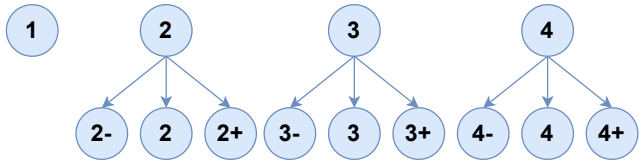


Figure 4.23.: Hierarchical dataset classes structure.

4.6.1. Ensemble architecture

In this work, we propose a new hierarchical method to build a classifier a classifier that exploits the hierarchical and ordinal properties of our AQC problem. The main idea behind this method is to do a multi-step classification, where, in the first step, we try to distinguish the macro classes (1, 2, 3 or 4, in this case) and, in the second step, we aim to classify the micro classes ($-$, $\cdot$ or $+$). Completely independent models are used for each step: for the first step, a single model is used to derive the macro class. In the second step, one separate model is used to determine the micro class for each of the aforementioned macro classes. Therefore, for this problem, four different models are used to obtain the final prediction. The number of models can vary for other problems where the number of classes or their hierarchy is different.

The model used to predict the macro class is defined as $f_M(\mathbf{x}) \rightarrow y_M$, where y_M is the predicted macro label. On the other hand, the models used to predict the micro classes are denoted as $f_{m_i}(\mathbf{x}) \rightarrow y_{m_i}$, where i-th classifier is associated with macro class i, and y_{m_i} is the micro label predicted by the classifier associated with i-th macro class. The predictor y_M is trained using all the samples in the training set, but these samples are labelled using only their macro class. In the same way, the classifiers y_{m_i} are trained using only the samples that belong to the i-th macro class and they are labelled using only the micro labels. Therefore, the complete hierarchical model can be defined as a ensemble model which is composed of 4 independent classifiers, whose decision function can be defined as:

$$f(\mathbf{x}) = \begin{cases} f_M(\mathbf{x}), & \text{if } \mathcal{O}(f_M(\mathbf{x})) = 1, \\ f_M(\mathbf{x}) \cup f_{m_i}(\mathbf{x}), & \text{if } \mathcal{O}(f_M(\mathbf{x})) > 1, \end{cases} \tag{4.24}$$

where $i = \mathcal{O}(f_M(\mathbf{x}))$, and $\mathcal{O}(\cdot)$ represents the order of any given ordinal class. Therefore, the final labels predicted by the hierarchical model can be denoted as:

$$y = \begin{cases} y_M, & \text{if } \mathcal{O}(y_M) = 1, \\ y_M \cup y_{m_i}, & \text{if } \mathcal{O}(y_M) > 1, \end{cases} \tag{4.25}$$

where $i = \mathcal{O}(y_M)$.

Taking into account the problem tackled in this work, $y_M \in \{1, 2, 3, 4\}$, and $y_{m_i} \in \{-, \cdot, +\}$, where $i \in \{2, 3, 4\}$.

Figure 4.24 illustrates the hierarchical scheme that has been described. On the left side, the models considered for this approach are defined, followed by the training procedure and the way that the final predictions are obtained.

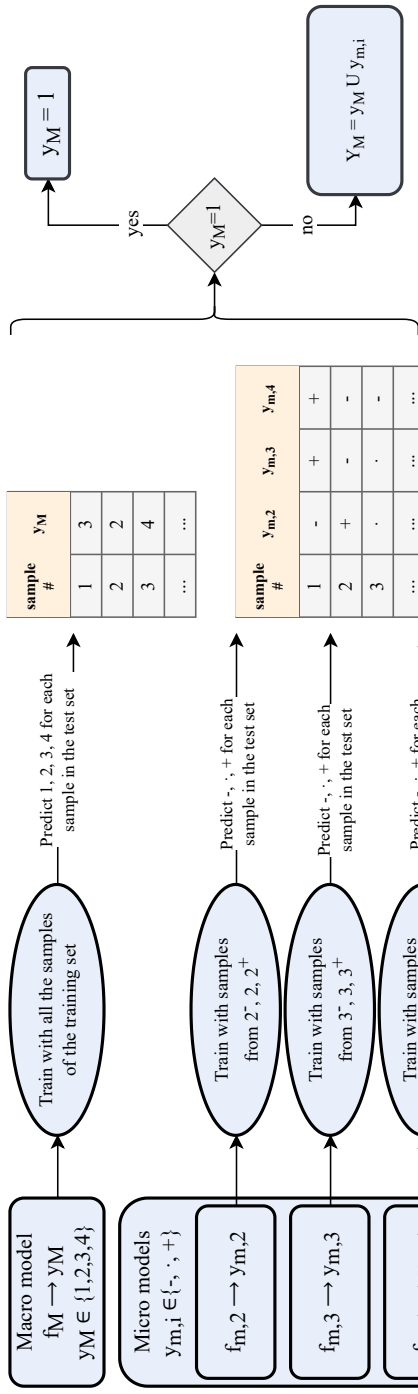


Figure 4.24.: Hierarchical models structure.

4.6.2. Classification models

In order to obtain more robust results regarding the proposed hierarchical methods, three different well-known architectures have been used: VGG-16 [136], ResNet-101 [137] and DenseNet-121 [153]. In this way, we can prove that the proposed methodology is not architecture dependant, and it can work with any CNN model.

Also, the convolutional parts of the models use the pre-trained ImageNet weights instead of adjusting them from a random initialisation. Therefore, only the top part of the model needs to be adjusted. For the top part of the model, a 50% dropout is performed before two dense layers with 4096 units. In the output layer, two different functions have been employed:

- Softmax function, which is the standard output function for classification tasks;
- Cumulative Link Model (CLM),as described in Section 4.4.1, that enhance ordinal classification performance. In this case, two different link functions have been used: logit and probit.

The VGG-16, ResNet-101 and DenseNet-121 contain 266M (251M trainable), 67M (25M trainable) and 493M (486M trainable) parameters, respectively. There are some fixed parameters due to the transfer learning approach. It is worth noting that the ResNet-101 has less parameters compared to the other alternatives. However, the residual neural network architectures have demonstrated having an outstanding generalisation capability with a reduced number of parameters.

4.6.3. Performance evaluation and Results

Experimental procedure

The models described in Section 4.6.2 were evaluated following a holdout scheme, where 80% of the whole set is used to adjust the model while the remaining 20% forms the test set. From the training set, another 15% of the samples are taken for the validation set, which is used to stop the training process when the model performance stops improving. All the experiments were repeated 30 times using different seeds to create the data partitions and initialise the model parameters. In this way, we obtained robust results from the point of view of a statistical analysis.

When using the hierarchical approach, different loss functions can be employed for the model which predicts the macro class and the models that predict the micro class. The experiments are performed using different loss functions to guide the optimisation algorithm:

- Categorical cross-entropy (CCE): the standard CCE is commonly used for nominal classification problems where classes do not follow any order, as defined in 4.11;

- Quadratic Weighted Kappa (QWK) Loss: the quadratic weighted kappa loss function for ordinal classification, as defined in 4.14;
- Beta regularised categorical cross-entropy (CCE-Beta): the unimodal regularised CCE that was described in [154].

In this way, we tested different loss functions combinations. Also, the output function can vary from one model to the others, leading to using an ordinal output like the CLM in the first step and the standard softmax in the second one.

Regarding the proposed hierarchical methods, different methodologies were used for the first classifier, where 4 classes are considered: C1 $= \{1\}$, C2 $= \{2^-, 2, 2^+\}$, C3 $= \{3^-, 3, 3^+\}$ and C4 $= \{4^-, 4, 4^+\}$, and the second classifier, which considers only three different classes $(-, \cdot, +)$. These methodologies are listed below and summarised in Table 4.19:

1. Hierarchical baseline. Softmax in the output layer and the standard CCE for both the first classifier and the next three classifiers.
2. Hierarchical CLM with logit link in the output layer and Beta regularised cross-entropy as loss function for the first and the subsequent classifiers.
3. Hierarchical CLM with probit link in the output layer and Beta regularised cross-entropy as loss function for the macro and the micro classifiers.
4. Hierarchical CLM with logit link in the output layer and Beta regularised cross-entropy loss for the first model and softmax function with the standard CCE loss for the micro models of the second stage.
5. Hierarchical CLM with probit link in the output layer and Beta regularised cross-entropy loss for the first model and softmax function with the standard CCE loss for the micro models of the second stage.
6. Hierarchical CLM with logit link in the output layer for the first and the next three models and QWK loss function for all of them.
7. Hierarchical CLM with probit link in the output layer and QWK loss function for the first and the second stage.

Models described in items 4 and 5 use an ordinal output function and an ordinal loss function for the first models, which tries to distinguish between 4 classes, and a nominal approach, for the next three models. Even though the problem that is solved in the second step is ordinal too, the number of classes is too small to benefit from the advantages of using an ordinal approach. Therefore, using a nominal approach for these models has been considered as a good alternative and is going to be compared with the rest of the experiments that have been proposed.

Table 4.19.: Hierarchical experiments types. Number of classes refers to the classes considered for each task.

	Macro loss	Macro output	# classes	Micros loss	Micros output	# classes
1	CCE	Softmax	4	CCE	Softmax	3, 3, 3
2	CCE Beta	CLM Logit	4	CCE Beta	CLM Logit	3, 3, 3
3	CCE Beta	CLM Probit	4	CCE Beta	CLM Probit	3, 3, 3
4	CCE Beta	CLM Logit	4	CCE	Softmax	3, 3, 3
5	CCE Beta	CLM Probit	4	CCE	Softmax	3, 3, 3
6	QWK	CLM Logit	4	QWK	CLM Logit	3, 3, 3
7	QWK	CLM Probit	4	QWK	CLM Probit	3, 3, 3

For comparison purposes, the non-hierarchical alternatives previously proposed in Section 4.2.2 and 4.4.1 were also run. In these cases, the number of classes considered is 10. These alternatives are listed below and summarised in Table 4.20:

8. Baseline. Softmax function in the output layer and the standard CCE as loss function.

9. CLM with logit link in the output layer and Beta regularised CCE as loss function.

10. CLM with probit link in the output layer and Beta regularised CCE as loss function.

11. CLM with logit link in the output layer and QWK loss function.

12. CLM with probit link in the output layer and QWK loss function.

13. ECOC with codes that represent the hierarchical structure of the classes [107]. In this case, each code is composed of 7 bits, where the first 4 bits are related to the macro class and the last 3 bits represent the micro class (e.g. for class 2^+, 0100 001).

14. ECOC with codes that contain the ordinal information of the labels [109], also defined as Ordinal Binary Decomposition (OBD) approach, describe in 4.4.3. They are composed of $Q-1$ bits and each bit q is set to 1 when the class that the code is associated to is higher than q (e.g. for class 2^+, which is the $4th$ class, $q=4$, 111000000).

The optimisation algorithm used for all the experiments is the Adam algorithm with a learning rate of 0.01. Taking into account the size of the dataset, the mini-batch size is fixed to 16. The model is trained for a maximum of 50 epochs. However, the early stopping strategy stops the training process when the validation loss stops improving. This strategy uses a patience value of 15, which determines the number of epochs without validation loss improvements before stopping the training process.

Table 4.20.: Different sets of non-hierarchical experiments. The ECOC alternatives (13 and 14) consist of 7 and 9 binary tasks respectively, given that the multi-class problem is decomposed in multiple binary tasks.

	Loss	Output	# classes
8	CCE	Softmax	10
9	CCE Beta	CLM Logit	10
10	CCE Beta	CLM Probit	10
11	QWK_L	CLM Logit	10
12	QWK_L	CLM Probit	10
13	CCE	Softmax	2 (7 tasks)
14	CCE	Softmax	2 (9 tasks)

Evaluation metrics

We considered four metrics that are appropriate for ordinal problems and imbalanced datasets: QWK [144], MS [146], MAE [146] and CCR. The metrics are defined as reported in Section 4.4.3. QWK, MS and CCR should be maximised, while MAE should be minimised.

Classification performance

The results of the experiments with the three different architectures are shown in Tables 4.21, 4.22 and 4.23. The experiments that are marked as hierarchical were run using the proposed hierarchical approach, while the non-hierarchical methods were run for comparison purposes.

Table 4.21.: Mean results for the test set and 30 executions using the VGG-16 architecture. The Hier. column indicates whether the method uses the proposed hierarchical methodology or not.

	Hier.	Macro loss	Macro output	Micros loss	Micros output	QWK↑	MS↑	MAE↓	CCR↑
1	Yes	CCE	Softmax	CCE	Softmax	0.8921	0.2311	0.7435	0.5129
2	Yes	CCE Beta	CLM Logit	CCE Beta	CLM Logit	0.9088	0.1778	0.7058	0.4908
3	Yes	CCE Beta	CLM Probit	CCE Beta	CLM Probit	0.9099	0.1782	0.7012	0.4937
4	Yes	CCE Beta	CLM Logit	CCE	Softmax	0.9055	**0.2400**	**0.6947**	**0.5238**
5	Yes	CCE Beta	CLM Probit	CCE	Softmax	0.9033	0.1893	0.7057	0.5230
6	Yes	QWK	CLM Logit	QWK	CLM Logit	0.9056	0.1349	0.7097	0.5031
7	Yes	QWK	CLM Probit	QWK	CLM Probit	0.9062	0.1334	0.7152	0.4948
8	No	CCE	Softmax	-	-	0.8713	0.1643	0.8253	0.4839
9	No	CCE Beta	CLM Logit	-	-	**0.9192**	0.2152	0.7121	0.4478
10	No	CCE Beta	CLM Probit	-	-	0.9161	0.2110	0.7299	0.4389
11	No	QWK	CLM Logit	-	-	0.9106	0.0000	0.7370	0.4544
12	No	QWK	CLM Probit	-	-	0.9123	0.0000	0.7358	0.4517
13	No	CCE	Softmax	-	-	0.8735	0.1552	0.8625	0.4666
14	No	CCE	Softmax	-	-	0.8852	0.0457	0.8420	0.4313

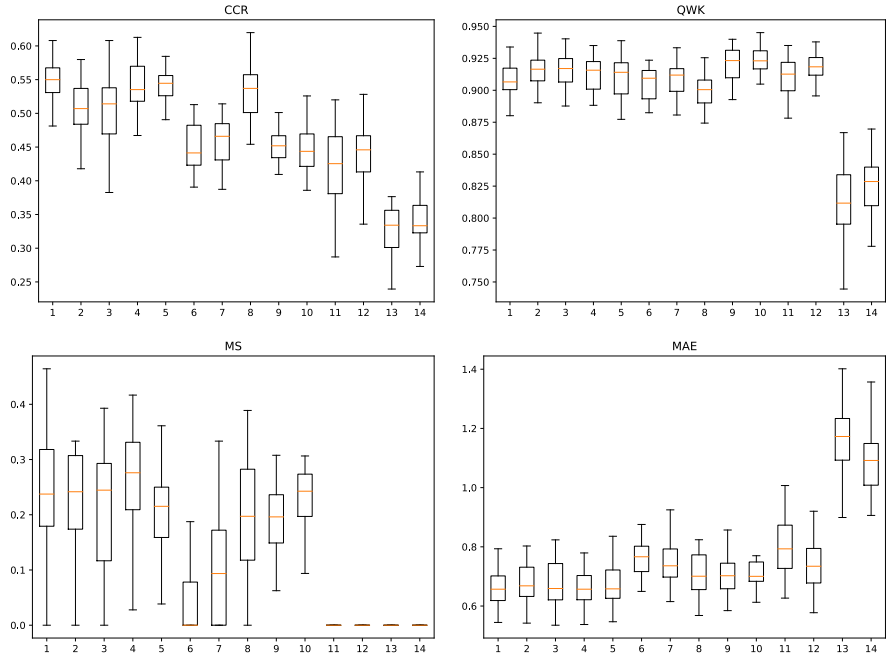


Figure 4.25.: Boxplots for all the test metrics using the ResNet-101 architecture. Methods are identified with the numbers defined in Table 4.22.

Table 4.22.: Mean results for the test set and 30 executions using the ResNet-101 architecture. The Hier. column shows whether the method uses the proposed hierarchical methodology or not.

	Hier.	Macro loss	Macro output	Micros loss	Micros output	QWK↑	MS↑	MAE↓	CCR↑
1	Yes	CCE	Softmax	CCE	Softmax	0.9080	0.2310	0.6647	**0.5479**
2	Yes	CCE Beta	CLM Logit	CCE Beta	CLM Logit	0.9165	0.2221	0.6737	0.5075
3	Yes	CCE Beta	CLM Probit	CCE Beta	CLM Probit	0.9155	0.2036	0.6863	0.5031
4	Yes	CCE Beta	CLM Logit	CCE	Softmax	0.9129	**0.2469**	**0.6612**	0.5408
5	Yes	CCE Beta	CLM Probit	CCE	Softmax	0.9108	0.2126	0.6765	0.5352
6	Yes	QWK	CLM Logit	QWK	CLM Logit	0.9065	0.0429	0.7599	0.4500
7	Yes	QWK	CLM Probit	QWK	CLM Probit	0.9059	0.0953	0.7586	0.4517
8	No	CCE	Softmax	-	-	0.9002	0.1981	0.7050	0.5318
9	No	CCE Beta	CLM Logit	-	-	0.9215	0.1910	0.7162	0.4427
10	No	CCE Beta	CLM Probit	-	-	**0.9239**	0.2314	0.7075	0.4435
11	No	QWK	CLM Logit	-	-	0.9091	0.0000	0.8078	0.4166
12	No	QWK	CLM Probit	-	-	0.9169	0.0000	0.7461	0.4357
13	No	CCE	Softmax	-	-	0.8120	0.0010	1.1705	0.3306
14	No	CCE	Softmax	-	-	0.8228	0.0000	1.0966	0.3395

From a solely descriptive point of view, the results show that the hierarchical method achieved the best results for CCR, MAE and MS for all the architectures. However, the non-hierarchical model that uses the Beta regularised cross-entropy with the CLM

Table 4.23.: Mean results for the test set and 30 executions using the DenseNet-121 architecture. The Hier. column indicates whether the method uses the proposed hierarchical methodology or not.

	Hier.	Macro loss	Macro output	Micros loss	Micros output	QWK↑	MS↑	MAE↓	CCR↑
1	Yes	CCE	Softmax	CCE	Softmax	0.8880	**0.1988**	**0.7261**	**0.5276**
2	Yes	CCE Beta	CLM Logit	CCE Beta	CLM Logit	0.8924	0.1508	0.7869	0.4560
3	Yes	CCE Beta	CLM Probit	CCE Beta	CLM Probit	0.8912	0.1506	0.7910	0.4631
4	Yes	CCE Beta	CLM Logit	CCE	Softmax	0.8888	0.1551	0.7621	0.5031
5	Yes	CCE Beta	CLM Probit	CCE	Softmax	0.8827	0.1703	0.7825	0.4997
6	Yes	QWK	CLM Logit	QWK	CLM Logit	0.8823	0.0906	0.8380	0.4344
7	Yes	QWK	CLM Probit	QWK	CLM Probit	0.8859	0.0933	0.8234	0.4435
8	No	CCE	Softmax	-	-	0.7696	0.0299	1.1420	0.4081
9	No	CCE Beta	CLM Logit	-	-	0.8867	0.1548	0.9041	0.3663
10	No	CCE Beta	CLM Probit	-	-	**0.8992**	0.1797	0.8314	0.4049
11	No	QWK	CLM Logit	-	-	0.8710	0.0000	1.0047	0.3467
12	No	QWK	CLM Probit	-	-	0.8790	0.0000	0.9606	0.3608
13	No	CCE	Softmax	-	-	0.8776	0.1773	0.8390	0.4707
14	No	CCE	Softmax	-	-	0.8921	0.1148	0.7996	0.4651

logit resulted in better performance for the QWK metric. For the VGG-16 models, the hierarchical alternative which uses the Beta regularised cross-entropy for the macro model combined with the CLM with logit link, and the standard cross-entropy loss with the standard softmax output for the micro models obtained the best results. In the case of the ResNet-101 architecture, the best results regarding MS and MAE were produced by the same alternative that achieved the best results in VGG-16. However, the best value for CCR was obtained when using the standard cross-entropy loss and the softmax for both, the macro and the micro models. Finally, when using the DenseNet-121 architecture, the best results are achieved with the standard cross-entropy and the softmax function for both steps. Also, in Figure 4.25, a boxplot is represented for each of the metrics considered for the architecture that obtained the best results (i.e. ResNet-101). For a more in depth comparison, the boxplots corresponding to the other model architectures have been added in A.1.

Another important aspect to consider is the model computational cost. It is worth noting that the ECOC approach usually spend more time for the training process, given that it decompose the original multi-class problem in multiple binary problems, and each of them is trained using all the training samples. Then, the computational cost of the proposed hierarchical approach should be lower than the cost associated with the ECOC approaches. To confirm this fact, the mean time required to complete both experiments was compared for each of the architectures. In these terms, for the VGG-16 architecture, the hierarchical approach took 20s while the ECOC needed 45 seconds per epoch. For the ResNet-101 model, the first took 27s and the second 91s. For the DenseNet-121 model, they took 23s and 51s respectively.

To sum up, the following conclusions can be obtained from the results tables:

- Our hierarchical approaches obtained the best results for MS, MAE and CCR considering all the model architectures.
- Our hierarchical model that uses the CCE Beta + CLM Logit for the macro task and CCE + Softmax for the micro classifiers obtained the best performance in most of the cases. The second best alternative is the hierarchical model which uses CCE + Softmax for both steps.
- The computational cost of the proposed hierarchical approach is lower than the cost associated with the ECOC approaches.

Statistical analysis

Even though the hierarchical method obtained the best performance for most of the metrics in all the model architectures, a statistical analysis was performed to determine which of the tested alternatives are significantly better than the others. Also, the aim was to check whether the proposed approach obtains better results than the baseline approach and previously proposed methods. To do that, we considered all the model architectures and each of the metrics was analysed separately.

First, for each of the four analysed metrics, a Kolmogorov-Smirnov [150] test was performed to check whether the 30 test values obtained, for each method and architecture, from the different seeds are normally distributed. The test confirmed that the values are normally distributed (p-value < 0.001) for all the metrics and methodologies and architectures, except for the MS metric when using the QWK loss. Therefore, an Analysis of variance II (ANOVA II) [155] test, where the factors considered are the methodology applied and the model architecture used, was performed for each of the metrics. It is worth noting that the statistical tests where performed using 90 points for each method (30 for each architecture).

First, the CCR metric was considered. $CCR_{ij}, (i = 1, ..., 12; j = 1, 2, 3)$ denotes all the methodologies considered. The observations fit the following equation:

$$CCR_{ijk} = \mu + \alpha_i + \beta_j + \gamma_{ij} + \epsilon_{ijk}, \quad k = 1, ..., 30, \tag{4.26}$$

where μ is the fixed effect that is common to all the populations, α_i is the effect associated with the i-th level of the first factor, β_j is the effect associated with the j-th level of the second factor, γ_{ij} is the interaction between the i-th level of the first factor and the j-th level of the second factor, and the term ϵ_{ijk} is the influence of the random effects in the final result. The results of this test are shown in Table 4.24.

When the p-value represented in the ANOVA table is smaller than 0.01, the factor effect is statistically significant at a level of confidence of 99%. The results obtained from this test reported that the methodology and the architectures used significantly influence the test accuracy value obtained. Also, there is an interaction between both factors that also influences the final result.

Table 4.24.: Results of the ANOVA II test for the CCR metric. SS stands for Sum of Squares, DF refers to the Degrees of Freedom, MSq are the Mean Squares, and F is the F-ratio.

Source	SS	DF	MSq	F	p-value
Corrected model	3.843	41	0.094	42.924	< 0.001
Intercept	267.244	1	267.244	122396.130	< 0.001
Method	2.252	13	0.173	79.338	< 0.001
Model	0.346	2	0.173	79.255	< 0.001
Method * Model	1.245	26	0.048	21.923	< 0.001
Error	2.659	1218	0.002		
Total	273.746	1260			
Corrected total	6.502	1259			

Given that there are significant differences in mean CCR depending on the methodology considered, a post-hoc HSD Tukey's [156] test was performed to compare the mean CCR values in the test set between all the methodologies. The results of this test are summarised in Table 4.25. It groups the methodologies into four different subsets according to their performance such that the elements within a subset are not significantly different between them, while the differences between members of different groups are significant. The first subset contains the worst methodologies while the last subset groups the best ones.

Table 4.25.: Results of the post-hoc HSD Tukey's test for the CCR metric.

						Subsets			
	Hier.	Macro loss	Macro out	Micro loss	Micro out	1	2	3	4
11	No	QWK	CLM Logit	-	-	0.4059			
14	No	CCE	Softmax	-	-	0.4120			
12	No	QWK	CLM Probit	-	-	0.4161			
9	No	CCE Beta	CLM Logit	-	-	0.4189			
13	No	CCE	Softmax	-	-	0.4226			
10	No	CCE Beta	CLM Probit	-	-	0.4289			
11	Yes	QWK	CLM Logit	QWK	CLM Logit		0.4625		
12	Yes	QWK	CLM Probit	QWK	CLM Probit		0.4633	0.4633	
8	No	CCE	Softmax	-	-		0.4746	0.4746	
2	Yes	CCE Beta	CLM Logit	CCE Beta	CLM Logit		0.4848	0.4848	
3	Yes	CCE Beta	CLM Probit	CCE Beta	CLM Probit			0.4866	
5	Yes	CCE Beta	CLM Probit	CCE	Softmax				0.5193
4	Yes	CCE Beta	CLM Logit	CCE	Softmax				0.5226
1	Yes	CCE	Softmax	CCE	Softmax				0.5295
p-values						0.060	0.080	0.052	0.975

The results in Table 4.25 depict that the ordinal methodologies tend to reduce the

accuracy even though they improve the performance regarding ordinal metrics. However, using the proposed hierarchical approach, the accuracy metric is statistically improved, obtaining higher values than the standard nominal approach. In this case, the best methodology was the hierarchical one that used the standard categorical cross-entropy loss and the softmax output for both, the macro and the micro models. However, there are no significant differences with the hierarchical methods which use the Beta regularised CCE and the CLM with logit or probit link for the macro model, and the standard CCE + softmax for the micro models. The fact that using a nominal approach for the second phase achieves better results is due to the lower number of labels in the micro-tasks.

The same statistical analysis has been performed for the QWK metric. The ANOVA II test also reported significant differences for the different factors (p-value < 0.001) and a significant interaction between them. Therefore, the results of the post-hoc HSD Tukey's test for the different methods considered are shown in Table 4.26.

Table 4.26.: Results of the post-hoc HSD Tukey's test for the QWK metric.

						Subsets			
	Hier.	**Macro loss**	**Macro out**	**Micro loss**	**Micro out**	1	2	3	4
8	No	CCE	Softmax	-	-	0.8470			
13	No	CCE	Softmax	-	-	0.8544	0.8544		
14	No	CCE	Softmax	-	-		0.8667		
11	No	QWK	CLM Logit	-	-			0.8952	
1	Yes	CCE	Softmax	CCE	Softmax			0.8960	
6	Yes	QWK	CLM Logit	QWK	CLM Logit			0.8982	
5	Yes	CCE Beta	CLM Probit	CCE	Softmax			0.8989	0.8989
7	Yes	QWK	CLM Probit	QWK	CLM Probit			0.8993	0.8993
12	No	QWK	CLM Probit	-	-			0.9007	0.9007
4	Yes	CCE Beta	CLM Logit	CCE	Softmax			0.9024	0.9024
3	Yes	CCE Beta	CLM Probit	CCE Beta	CLM Probit			0.9056	0.9056
2	Yes	CCE Beta	CLM Logit	CCE Beta	CLM Logit			0.9059	0.9059
9	No	CCE Beta	CLM Logit	-	-			0.9091	0.9091
10	No	CCE Beta	CLM Probit	-	-				0.9130
p-values						0.916	0.198	0.076	0.065

In this case, the results show that the best methodology is the non-hierarchical one that uses the Beta regularised cross-entropy and the CLM with logit link in the output. However, there are no significant differences with the other methods in the same group. The worst results were obtained by the standard nominal approach and the ECOC methods. The ordinal and hierarchical methodologies highly improved the performance concerning the standard nominal approach.

Following the same methodology, the MS metric was analysed. Again, the ANOVA II test performed over the MS test results reported that there are significant differences between the methodologies and between the different architectures. Moreover, there is

a significant interaction between these two factors. Therefore, a posthoc HSD Tukey's Test taking into account the methodologies was performed and the results are shown in Table 4.27.

Table 4.27.: Results of the post-hoc HSD Tukey's test for the MS metric.

						Subsets						
	Hier.	**Macro loss**	**Macro out**	**Micro loss**	**Micro out**	1	2	3	4	5	6	7
11	No	QWK	CLM Logit	-	-	0.000						
12	No	QWK	CLM Probit	-	-	0.000						
14	No	CCE	Softmax	-	-		0.080					
6	Yes	QWK	CLM Logit	QWK	CLM Logit		0.090	0.090				
7	Yes	QWK	CLM Probit	QWK	CLM Probit		0.107	0.107				
8	No	CCE	Softmax	-	-			0.1308	0.1308			
13	No	CCE	Sofmtax	-	-				0.166	0.166		
3	Yes	CCE Beta	CLM Probit	CCE Beta	CLM Probit					0.178	0.178	
2	Yes	CCE Beta	CLM Logit	CCE Beta	CLM Logit					0.184	0.184	0.184
9	No	CCE Beta	CLM Logit	-	-					0.187	0.187	0.187
10	No	CCE Beta	CLM Probit	-	-					0.207	0.207	0.207
5	Yes	CCE Beta	CLM Probit	CCE	Softmax					0.208	0.208	0.208
4	Yes	CCE Beta	CLM Logit	CCE	Softmax						0.214	0.214
1	Yes	CCE	Softmax	CCE	Softmax							0.220
p-values						1.000	0.667	0.064	0.220	0.063	0.179	0.173

In this case, the same hierarchical approach that performed the best for the CCR metric also obtained the best MS results.

By analysing the composition of each of the seven subsets, some conclusions can be obtained:

1. The non-hierarchical methodologies that use the QWK loss function fail to classify at least one of the classes, obtaining always a value of 0 for the minimum sensitivity. Therefore, they should be discarded even though they perform well regarding the other metrics.

2. The hierarchical approach solves the problem related to the QWK loss: all the classes are represented in the final predictions.

3. Five out of seven methods grouped in the best three subsets are hierarchical, including the three methods that obtained the best average performance.

Finally, for the MAE metric, the same analysis was performed. The ANOVA II test reported significant differences between the methods considered and the architectures tested (p-value < 0.001). Also, there is a significant interaction between factors (p-value < 0.001) Then, a post-hoc HSD Tukey's test was performed to determine which methods achieve better performance. The results of this test are given in Table 4.28.

The results in this table show that the best mean result was obtained by the hierarchical methodology which employs the Beta regularised cross-entropy loss with the CLM Logit in the first model and the CCE with softmax for the others. However,

Table 4.28.: Results of the post-hoc HSD Tukey's test for the MAE metric.

						Subsets							
	Hier.	**Macro loss**	**Macro out**	**Micro loss**	**Micro out**	1	2	3	4	5	6	7	8
4	Yes	CCE Beta	CLM Logit	CCE	Softmax	0.706							
1	Yes	CCE	Softmax	CCE	Softmax	0.711	0.711						
5	Yes	CCE Beta	CLM Probit	CCE	Softmax	0.722	0.722	0.722					
2	Yes	CCE Beta	CLM Logit	CCE Beta	CLM Logit	0.722	0.722	0.722					
3	Yes	CCE Beta	CLM Probit	CCE Beta	CLM Probit	0.726	0.726	0.726					
10	No	CCE Beta	CLM Probit	-	-	0.756	0.756	0.756	0.756				
7	Yes	QWK	CLM Probit	QWK	CLM Probit	0.766	0.766	0.766	0.766				
6	Yes	QWK	CLM Logit	QWK	CLM Logit		0.769	0.769	0.769				
9	No	CCE Beta	CLM Logit	-	-			0.778	0.778				
12	No	QWK	CLM Probit	-	-				0.814	0.814			
11	No	QWK	CLM Logit	-	-					0.850	0.850		
8	No	CCE	Softmax	-	-						0.891	0.891	
14	No	CCE	Softmax	-	-							0.913	0.913
13	No	CCE	Softmax	-	-								0.957
p-values						0.063	0.086	0.113	0.084	0.791	0.590	0.995	0.442

there are no significant differences with the other methods in group 1. Also, the table shows that most of the methods in group 1 are hierarchical methods. The last four groups only contain non-hierarchical methods and the ECOC approaches. Therefore, the overall results with hierarchical methods are better than the ones obtained by the corresponding non-hierarchical ones.

After comparing the different methods using the post-hoc tests, the three model architectures were compared too. The post-hoc HSD Tukey's test shown that the VGG-16 and the ResNet-101 are not significantly different and they both are significantly better than the DenseNet-121 concerning the QWK and MS metrics. For the CCR metric, the ResNet-101 is significantly better than the other two architectures, which also are significantly different between them. Finally, regarding the MAE, the VGG-16 architecture achieved the best mean results. The complete statistical comparison of the architectures can be found in A.2.

Therefore, taking into account all the metrics, some general conclusions can be derived:

1. The proposed hierarchical methodologies perform significantly better concerning the CCR, MS and MAE metrics, while the non-hierarchical ones achieve better results for the QWK. However, there are no significant differences concerning QWK with most of the other methodologies.

2. Using an ordinal loss function and output function for the macro classifier and the standard nominal approach for the micro classifier usually obtained the best results.

3. The ResNet-101 obtained the best results followed by the VGG-16 architecture. Nevertheless, the proposed methodology obtains a significant performance en-

hancement for all architectures tested. Therefore, the proposed method can be generalised to different types of architectures.

4.7. Learning ordinal-hierarchical constraints simultaneously

In previous sections, we first described how to deal with the ordinal nature of the proposed AQC problem. Then, we faced the hierarchical structure of dataset categories, highlighting that our classification task can be defined as a hierarchical ordinal problem where the categories are displayed in an ordinal structure on different hierarchical levels. However, the simultaneous managing of both the hierarchical and ordinal nature is not a trivial problem, involving the use of multiple models to perform the task (see Section 4.6.1).

Hence, we proposed to simultaneously learn hierarchical-ordinal constraints by using a DL methodology consisting in a single network. The proposed approach is designed according to a novel hierarchical formulation that models local and global losses, where local losses act as auxiliary losses to strengthen the hierarchical-ordinal dependencies. To integrate ordinal relation within global and local losses we proposed to include a CLM combined with QWK loss and an OBD approach with MAE loss.

The overall methodology is described in Figure 4.26. The hierarchical-ordinal problem is represented using a graph structure. Our hierarchical problem is decomposed by local and global classes to learn consistently different classes in the hierarchy. Note that the order of nodes within a level reflects the natural ordinal structure of the classes. However, this natural order in the labels does not necessarily correspond with the order defined by the corresponding super-classes. Accordingly, the ordinal constraints are integrated using CLM and OBD approaches. Thus our framework leads to the design of two different methodologies called Hierarchical cumulative link model (HCLM) and Hierarchical ordinal binary decomposition (HOBD) that can be generalized for solving generic and real-world hierarchical ordinal problems.

In Section 4.7.1, the notation we used to formulate our approach is set and the formulation of the proposed HCLD and HOBD approaches is described. Network architecture and prediction phase are treated in Section 4.7.2.

4.7.1. HCLM & HOBD methodologies

Notation

The adopted notations are described in Table 4.29. Figure 4.27 shows an example of general hierarchical constraint settings in terms of graph structure. In our problem definition, the labels of each child node associated with different parent nodes can be completely different, reflecting different structures (i.e. $h|\bigcap_{i=1}^{Y_L^h} \text{children}(y_i^h) = \emptyset$).

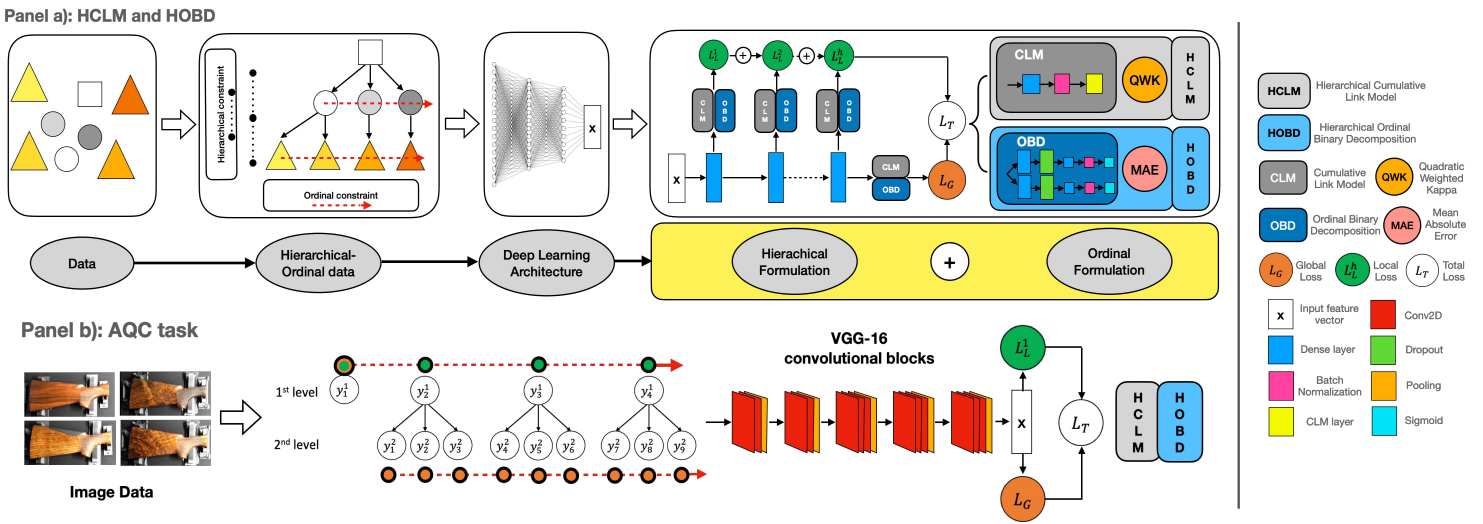


Figure 4.26.: Overview of the proposed method. Panel a) describes the general formulation by which the hierarchical-ordinal problem is decomposed and modulated via HOBD and HCLM alternatively. Panel b) shows how the proposed framework was adapted for the Aesthetic Quality Control (AQC) dataset.

Table 4.29.: Commonly used notations.

Notation	Description
X	Input space ($\mathbb{R}$)
M	Number of training points
H	Number of hierarchical levels of the task
Q	Number of classes of the general problem
Y	Set of classes of the general problem
Y_L^h	Subset of local classes of the h-th level
Y_L	Set of all the local classes
$\lvert Y_L^h \rvert$	Number of local classes of the h-th hierarchical level
Y_G	Subset of global classes
$\lvert Y_G \rvert$	Number of global classes
Y_T	Set of global and local classes
y_i^h	i-th class label of h-th hierarchical level
children(y_i^h)	Set of child classes of y_i^h
$f(\cdot)$	Predictive model
$\mathcal{F}(\cdot)$	Set of Predictive models
$\mathcal{L}$	Set of all the possible loss functions
L_L^h	Local loss related to local classes associated to the h-th level
L_G	Global loss related to global classes
L_T	Total loss

Definition 1 *The local classes of the h-th level of the hierarchy (Y_L^h) are defined as follows:*

$$Y_L^h = \bigcup_{i=1}^{|Y_L^h|} y_i^h \tag{4.27}$$

where $h \in \{1, 2, \ldots, H-1\}$. Note that the local classes include all the nodes that are not in the last hierarchical level (see Figure 4.27).

Definition 2 *The global classes (Y_G) are defined as follows:*

$$Y_G = \bigcup_{h=1}^{H} \left[\bigcup_{i|children(y_i^h)=\emptyset} y_i^h \right], \tag{4.28}$$

where children$(\cdot)$ represents the set of child classes of a given node, and those nodes fulfilling children$(y_i^h) = \emptyset$ correspond to leaves of the graph. Indeed, the global classes consist of classes which have not any descendants (see Figure 4.27), thus reflecting the original categories of the classification problem without considering the parent nodes.

This definition of local and global classes allows to deal with classification problems that can also be arranged on different hierarchical levels. As shown in the example of Figure 4.27, the global classes consists on the set

$Y_G = \{y_1^H, y_2^H, y_3^H, y_4^H, y_5^H, y_6^H, y_7^H, y_8^H, y_9^H, y_4^2, y_6^2, y_9^2, \dots\}$ while the local classes consists on $Y_L^1 = \{y_1^1, y_2^1, y_3^1\}$, $Y_L^2 = \{y_1^2, y_2^2, y_3^2, y_4^2, y_5^2, y_6^2, y_7^2, y_8^2, y_9^2\}, \dots$.

Definition 3 *The definition of local and global classes leads to:*

$$Y_T = Y_L \cup Y_G, \tag{4.29}$$

where

$$Y_L = \bigcup_{h=1}^{H-1} Y_L^h. \tag{4.30}$$

Note that those classes which are leaves but are not placed on the H-th level (last level) will be simultaneously global and local classes. The reason is that they are part of the original global classification task but should also be taken into account to represent the ordinal structure of the corresponding level.

Considering the proposed AQC task, the local classes consist on $Y_L^1 = \{y_1^1, y_2^1, y_3^1, y_4^1\} = \{1, 2, 3, 4\}$, while the global classes consist on the set $Y_G = \{y_1^1, y_1^2, y_2^2, y_3^2, y_4^2, y_5^2, y_6^2, y_7^2, y_8^2, y_9^2\} = \{1, 2^-, 2^c, 2^+, 3^-, 3^c, 3^+, 4^-, 4^c, 4^+\}$ (see Figure 4.26, panel b).

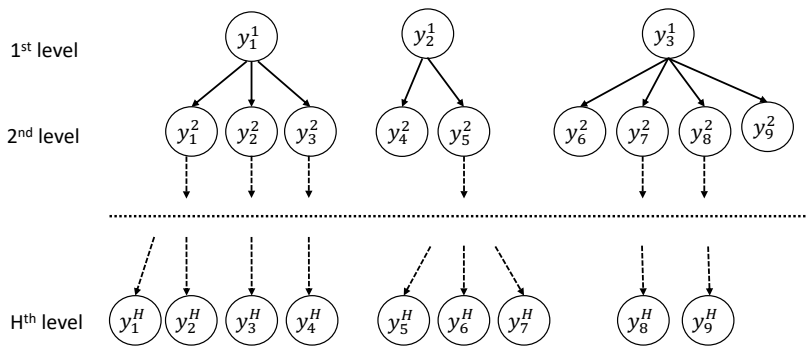


Figure 4.27.: Example of the definition proposed for global and local classes when describing the hierarchical task. In this example, the global classes consists on the set $Y_G = \{y_1^H, y_2^H, y_3^H, y_4^H, y_5^H, y_6^H, y_7^H, y_8^H, y_9^H, y_4^2, y_6^2, y_9^2, \dots\}$ while the local classes consists on $Y_L^1 = \{y_1^1, y_2^1, y_3^1\}$, $Y_L^2 = \{y_1^2, y_2^2, y_3^2, y_4^2, y_5^2, y_6^2, y_7^2, y_8^2, y_9^2\}, \dots$, while the relationships between levels are described by children$(y_1^1) = \{y_1^2, y_2^2, y_3^2\}$, children$(y_2^1) = \{y_4^2, y_5^2\}$, children$(y_3^1) = \{y_6^2, y_7^2, y_8^2, y_9^2\}$, children$(y_4^2) = \emptyset$, children$(y_6^2) = \emptyset$, children$(y_9^2) = \emptyset, \dots$.

Hierarchical formulation

We let $f(\cdot)$ be the predictive model for mapping the input vector to the global and local classes. In our formulation we combine local and global losses as follows:

$$L_T = \beta \frac{\sum_{h=1}^{H-1} L_L^h}{H-1} + (1-\beta) L_G, \tag{4.31}$$

where $\beta \in [0, 1]$ is the hyperparameter that regulates the trade-off regarding local and global information. L_L^h refers to the local loss computed according to the h-th level:

$$L_L^h \in \mathcal{L}(X, Y_L^h), \tag{4.32}$$

where $\mathcal{L}(X, Y_L^h)$ is the set of loss functions computed using the input data, X, and the classes of the h-th hierarchical level, Y_L^h.

On the other hand, L_G refers to the global loss computed:

$$L_G \in \mathcal{L}(X, Y_G), \tag{4.33}$$

where $\mathcal{L}(X, Y_G)$ considers in this case the set of global classes.

In our formulation we both aggregate and minimize the local losses (i.e. specific models responsible for the prediction of local classes) and the global loss (i.e. global model responsible for the prediction of global classes). The rationale behind this choice lies in both maximizing a consistent global prediction in the final hierarchical level and obtaining a consistent prediction for each local node. By considering also the ordinal nature of the task we aim to solve, we also ensure that each local loss function reinforces the propagation of gradients leading to proper local-information ordering among classes of the corresponding hierarchical level. At the same time, the global loss function keeps track of the label dependency in the hierarchy as a whole, adapting, at the same time, the ordinal constraint according to the leaves nodes. This aspect is described in details in the next paragraph.

Hierarchical-ordinal formulation

Here the proposed hierarchical-ordinal methodologies, namely Hierarchical cumulative link model (HCLM) and Hierarchical ordinal binary decomposition (HOBD) are presented.

A) Hierarchical cumulative link model

In this method, we integrated the CLM, defined as in Section 4.4.1, for modeling both the local and global losses (see Equation 4.31). Thus, the candidate classes Y become Y_G and Y_L, and the number of classes Q becomes $|Y_G|$ and $|Y_L^h|$, for the global and local losses, respectively.

The employed loss function is the QWK loss. Starting from the definition in 4.14, the QWK local loss computed for the h-th level is reformulated as follows:

$$L_L^h = \frac{\sum_{m=1}^{M} \sum_{y_i \in Y_L^h} \omega_i P(y = y_i | \mathbf{x}_m)}{\sum_{i=1}^{|Y_L^h|} \frac{M_i}{M} \sum_{y_j \in Y_L^h} (\omega_{i,j} \sum_{k=1}^{M} P(y = y_j | \mathbf{x}_k))}, \tag{4.34}$$

where $\mathbf{x}_m$ is the input data of the m-th sample, M_i is the number of training samples of the i-th local class, $P(y = y_i|\mathbf{x}_m)$ is the model posterior probability that the m-th sample belongs to local class y_i, and $\omega_{i,j}$ are the elements of the penalization matrix (for the quadratic version, $\omega_{i,j} = \frac{(i-j)^2}{(|Y_L^h|-1)^2}$).

Similarly, the QWK global loss is defined as follows:

$$L_G = \frac{\sum_{m=1}^{M} \sum_{y_i \in Y_G} \omega_i P(y = y_i | \mathbf{x}_m)}{\sum_{i=1}^{|Y_G|} \frac{M_i}{M} \sum_{y_j \in Y_G} (\omega_{i,j} \sum_{k=1}^{M} P(y = y_j | \mathbf{x}_k))}, \tag{4.35}$$

where $P(y = y_i|\mathbf{x}_m)$ is the model posterior probability that the m-th sample belongs to global class y_i.

B) Hierarchical ordinal binary decomposition

As introduced in 4.4.3, OBD is an ordinal approach that is based on decomposing the classification task of Q classes of the original problem into a set of $Q - 1$ binary problems, where each problem q consists in determining, for a given class y_q, if $y \succ y_q$ conditioned to $\mathbf{x}$ ($1 \leq q < Q$) [72], given that $y \succ y_Q$ is trivially false.

In this case, the employed loss function is the MAE loss. For the local losses, the MAE computed for the h-th hierarchical level is defined as follows:

$$L_L^h = \frac{1}{|Y_L^h| - 1} \sum_{y_i \in Y_L^h} \|1\{y \succ y_i\} - P(y \succ y_i|\mathbf{x})\|, \tag{4.36}$$

where $\mathbf{x}$ is the input data matrix and $P(y \succ y_i|\mathbf{x})$ is the model posterior cumulative probability for local class y_i, and $1\{y \succ y_i\}$ is the corresponding target vector.

The MAE global loss is defined as follows:

$$L_G = \frac{1}{|Y_G| - 1} \sum_{y_i \in Y_G} \|1\{y \succ y_i\} - P(y \succ y_i|\mathbf{x})\|, \tag{4.37}$$

where $P(y \succ y_i|\mathbf{x})$ is the model posterior cumulative probability for global class y_i, and $1\{y \succ y_i\}$ is the corresponding target vector.

4.7.2. Classification model

Architecture

The structure of $\mathcal{F}(\cdot)$ does not require any special assumption, thus different model structures are equally acceptable. The hierarchical constraint is modulated on the top fully-connected (FC) layers of the networks, by forming a multi-output classification head to achieve both local and global ordinal optimizations. This classification head is composed of $H-1$ local outputs and one global output. The main flow is composed of H FC layers with ReLU activation to which local sub-modules are connected. Each local sub-module is characterized by its own FC layer before the local output. In our approach, this ensures that each local sub-module learns the ordinal constraint from a given hierarchical level. In our architecture, Batch Normalization (BN) [157] was inserted to accelerate the convergence of the networks and to improve the stability of training.

As regards the proposed HCLM approach, each local output and the global one present only one neuron, which provides the model projection in a 1-dimensional space. This value is used to classify the sample into the corresponding class according to the CLM with logit activation function. Accordingly, for the proposed HOBD approach, each local sub-module FC layer is decomposed into a set of $|Y_L^h|-1$ FC blocks (with the same dimension). Each block consists of a FC layer, followed by a Leaky ReLU activation function and a dropout layer. Each final output layer with sigmoid activation function solves an individual binary classification subproblem. The global output also assumes the same decomposition, presenting $|Y_G|-1$ binary outputs. These structures are reported in Figure 4.27, panel a).

In our experiments we employed a VGG-16 architecture as feature extractor, maintaining the pre-trained ImageNet weights for the convolutional part of the model according to a transfer learning approach. Also in this case, the choice of VGG-16 architecture as baseline model is related to our earlier work, where VGG-16 achieved the best results among other state-of-the-art classification models in the standard nominal classification (see Section 4.2.3. Nevertheless, as previously stated, our approach can be generalized to different types of architecture (i.e. $\mathcal{F}(\cdot)$). A dropout regularization layer was inserted in the first FC layer of the classification head. The rate of dropout and the size of all dense layers are selected within the hyperparameters optimization procedure (Table 4.30).

Prediction

According to the formulation in 4.7.1, predictions can be obtained both for local or global classes. Predicted classes are differently obtained depending on the considered methodology (HCLM or HOBD). As CLM provides us with posterior probabilities,

the predicted classes of HCLM are obtained by:

$$\hat{y}_L^h = \underset{y_i \in Y_L^h}{\operatorname{argmax}} P(y = y_i|\mathbf{x}), \tag{4.38}$$

$$\hat{y}_G = \underset{y_i \in Y_G}{\operatorname{argmax}} P(y = y_i|\mathbf{x}). \tag{4.39}$$

In the case of HOBD, we compute the distance between the model posterior probability vector and the ground-truth of global and local labels. The predicted label is the one that has the minimum distance and can be computed for the global and local classes as follows:

$$\hat{y}_L^h = \underset{y \in Y_L^h}{\operatorname{argmin}} ||\mathbf{p}_L^h - \mathbf{t}_L^h(y)||, \tag{4.40}$$

$$\hat{y}_G = \underset{y \in Y_G}{\operatorname{argmin}} ||\mathbf{p}_G - \mathbf{t}_G(y)||, \tag{4.41}$$

where $\mathbf{p}_L^h = \left(P(y \succ y_i|\mathbf{x}) : y_i \in Y_L^h\right)$ and $\mathbf{p}_G = \left(P(y \succ y_i|\mathbf{x}) : y_i \in Y_G\right)$ are two vectors containing all posterior cumulative probabilities (associated to the independent binary subproblems of OBD) for global and local classes, respectively, and $\mathbf{t}_L^h(y) = \left(1\{y \succ y_i\} : y_i \in Y_L^h\right)$ and $\mathbf{t}_G(y) = \left(1\{y \succ y_i\} : y_i \in Y_G\right)$ are the corresponding target vectors for global and local classes, respectively.

Note that, in our experiments, we have evaluated only the prediction for global classes, as these are the most important ones in terms of cost for the considered real problem. However, local predictions could also be useful for other practical contexts.

4.7.3. Performance evaluation and Results

Experimental procedure

HCLM and HOBD methodologies were evaluated following a stratified over rifles holdout procedure: the dataset was split by maintaining 80% of the whole set for the training phase and the remaining 20% for the test set. From the training set, another 15% of the samples were taken for the validation set. Experiments with on-the-fly data augmentation strategy were performed for balancing global classes of AQC dataset, randomly applying a horizontal flip to all the training samples.

Adam [149] was adopted as optimizer and the best learning rate, batch size and dropout rate were selected as hyperparameters. Moreover, we evaluated also the contribution of local and global ordinal losses exploring values for the β parameter in range $\{0.2, 0.5, 0.8\}$. All hyperparameters were tuned in the separate validation set using a grid-search approach (see Table 4.30).

For all the experiments, the number of training epochs was set to 50 while adopting the early stopping strategy with a patience value of 15 epochs monitoring validation loss. To achieve robust results from a statistical perspective, all the experiments were

performed 30 times using different seeds to create the data partitions and initialize the model parameters.

All the experiments were run using TensorFlow 2.0 and Keras 2.3.1 frameworks on an Intel Core i7-4790 CPU 3.60GHz with 16GB of RAM and NVIDIA GeForce GTX 970.

Table 4.30.: Model hyperparameters explored in the validation set. All hyperparameters were tuned in the separate validation set using a grid-search approach.

Model	Approach	Hyperparameters	Range
VGG-16	HCLM	FC neurons	$\{2048, 4096\}$
		Dropout rate	$\{0.1, 0.3, 0.5\}$
		Batch size	$\{8, 16, 32, 64\}$
		Learning rate	$\{10^{-4}, 10^{-3}, 10^{-2}\}$
		β	$\{0.2, 0.5, 0.8\}$
VGG-16	HOBD	FC neurons	$\{2048, 4096\}$
		Dropout rate	$\{0.1, 0.3, 0.5\}$
		Batch size	$\{8, 16, 32, 64\}$
		Learning rate	$\{10^{-4}, 10^{-3}, 10^{-2}\}$
		β	$\{0.2, 0.5, 0.8\}$

Experimental comparisons

Different from other state-of-the-art work, our proposed methodologies are conceived for learning both ordinal and hierarchical dependencies. For that reason we decided to compare the proposed HCLM and HOBD with respect to other hierarchical and ordinal formulations widely employed in the ML literature. The state-of-the-art comparisons include:

- Global (GLB) approach [104]. This approach maps hierarchical problem into a standard classification problem that fully embeds the parent level information. The nominal global approach (GLB-NOM) ignores the class hierarchy predicting only leaf nodes classes as a standard multi-class classification [110] with categorical cross-entropy (CCE) loss. The ordinal variant of this approach was implemented by integrating ordinal relationship through OBD [72] (GLB-OBD) and CLM [93] (GLB-CLM);

- Local Classifier per Parent Node (LCPN) [158, 159, 160] approaches. Models are trained for solving each local task (i.e. a separate model for each parent node) using nominal (LCPN-NOM) or ordinal classifiers. The ordinal relationship was encoded with an OBD (LCPN-OBD) and a CLM (LCPN-CLM) layer, by considering, respectively, MAE and QWK as loss functions. Binary sub-

problems were treated with a sigmoid activation function on the output neuron and Binary Cross-Entropy (BCE) loss;

- Multi-Task Learning (MTL) approach [161]. The hierarchical constraints is decomposed in two different tasks: this strategy is viable only when the global label can be handled as a combination of labels from two distinct tasks (i.e. macro ($\{1, 2, 3, 4\}$ classes) and micro ($\{+, \ , -\}$ classes) task for AQC dataset).

  In the MTL-CLM formulation, we computed the QWK macro and micro losses related to the macro and micro classes respectively. We also extended this comparison (MTL-CLM$_{loc}$) with the aim of including a hierarchical constraint by minimizing the micro loss locally for each macro class. Considering that class 1 has no child labels, a further post-processing step was necessary to put this hierarchical constraint. It is worth noting that this further step is not need in our approach. In MTL-OBD and MTL-OBD$_{loc}$, OBD approach is applied to both decomposable tasks and MAE losses are minimized.

- Hierarchical multi-label classification network (HMCN) [111]. This method consists of a multi-label binary encoding strategy in which a BCE loss is minimized for each hierarchical level. In this case, the output is a binary class vector (expected output) containing all classes in the hierarchy. It is worth noting that in contrast with our approach, in this case, the model may lead to the prediction of non-admissible paths, thus requiring a further post-processing stage (i.e. violation constraint) to avoid inconsistent global classes.

Evaluation metrics

Note that the proposed HCLM and HOBD approach can be used for both predicting local and global classes. However, we decided to focus in evaluating the more relevant classes (i.e. global aggregated classes for AQC task) related to the real dataset we used. Considering the ordinal nature of global and local classes, ordinal metrics were chosen for evaluating our hierarchical-ordinal problem, in addition to standard nominal CCR. As highlighted in previous sections, these metrics (i.e. QWK, MAE, and 1-off accuracy, defined in Section 4.4.3) properly reflect the deviation of a misclassification error from the actual class. CCR, QWK and 1-off accuracy are to be maximized, MAE is to be minimized.

Classification performance

Table 4.31 shows the comparison of the proposed approach with respect to GLB, LCPN, MTL and HMCN competitors for the AQC dataset. With QWK=$0.921(0.009)$ and MAE = $0.635(0.048)$, the HOBD approach outperforms all the baseline algorithms. Moreover the averaged ranking in terms of QWK ($R_{\text{QWK}} = 2.100(1.213)$) and MAE ($R_{\text{MAE}} = 2.000(1.203)$) for HOBD are higher than those reported by all

state-of-the-art competitors. The HCLM discloses lower performance compared with HOBD and comparable results with respect to state-of-the-art methods. Overall, the proposed HOBD approach is consistent to the imbalanced setting of this task. Indeed, the adoption of the data augmentation procedure reflects no improvement in terms of QWK and an improvement of 0.6% in terms of MAE.

Statistical Analysis

We evaluated the statistical significance of our best performing approach with respect to other competitors. First, we performed an Anderson-Darling test [162] to test that the values of the metrics follow a normal distribution considering $\alpha = 0.05$. In this way, the QWK ($p = 0.697$) and MAE ($p = 0.894$) scores for the proposed HOBD approach were found to follow a normal distribution.

Hence, a paired-sample one-sided t-test ($\alpha = 0.05$) was performed to compare the QWK and MAE of the best-performing HOBD with respect to the best-performing state-of-the-art methodologies. QWK scores were found to be significantly ($\alpha = 0.01$) higher for HOBD than all GLB, LCPN, MTL and HMCN models. Accordingly, we also found the MAE scores to be be significantly ($\alpha = 0.01$) lower for HOBD than all GLB, LCPN, MTL and HMCN models

Therefore, HOBD method proved to be effective in dealing with an hierarchical-ordinal problem as the AQC one, by overcoming the other state-of-the-art nominal, hierarchical and ordinal approaches. Thus, the proposed approaches can be suited for solving any other real-world classification tasks that exhibits hierarchical and ordinal properties.

4.8. DSS for AQC classification

The main function of AQC is to build a method to objectify the result of visual inspections, which are still purely dependent on the evaluation of human operators, mitigating the inter- and intra-operator variability. Thus, a DSS based on the AI techniques previously described can make this control more reliable, fast, and standardized. The integration of the proposed methodologies as the main core of a DSS for solving AQC task is described in Figure 4.28. The DSS platform is comprised of the acquisition bench, GUI interface and cloud architecture.

As regards the cloud environment, a container logic was adopted for packaging the predictive model and all its dependencies, allowing the inference phase to run reliably from one computing environment to another. A docker image is essentially a snapshot of a container. Microsoft Azure framework was adopted for providing a cloud-based environment using virtualized containers. This environment can ensure hardware and software isolation, flexibility, and inter-dependencies between data collection, model building, and prediction phases. Indeed, the proposed approach is integrated into a

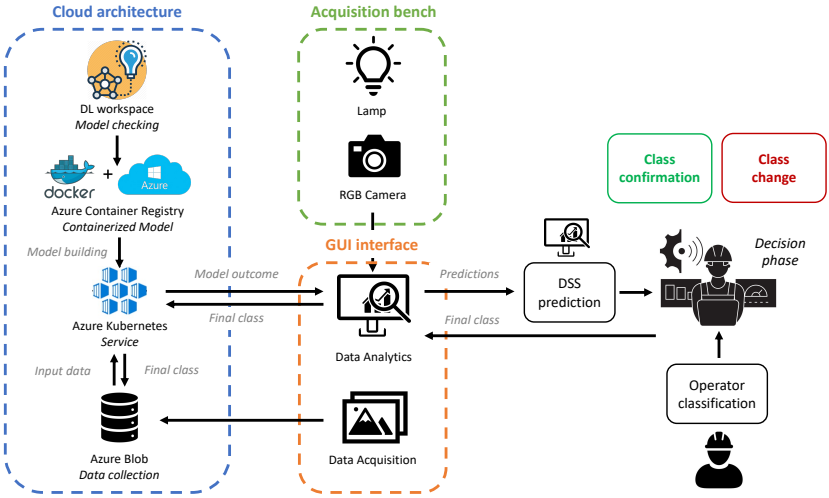


Figure 4.28.: DSS cloud interface.

AQC serverless platform where the predicted quality class is obtained by ingestion event. The technician may trigger a cloud function that could invoke the DL model to provide the inference. This setting may ensure the high scalability of the system while allowing the continuous fine-tuning of the model based on new images of rifles available. All prediction results were stored in the Azure blob storage and displayed to human operator in a GUI interface.

The GUI software presents three basic features:

- *Levels registry*: define the quality labels on which the ML/DL algorithms are trained and, consequently, the classes considered for model prediction. The levels are stored on the system in the working database.
- *Annotation*: collect and store images of the annotated dataset. The software, once the item has been placed on the acquisition bench, allows photo acquisition and awaits the operator annotation. Images are stored in the file system as .png files and associated with the annotation tag (including date and time of collection) on database. The DSS platform was also used to collect the employed image dataset described in Section 4.1.
- *Supervised inference*: this is the operational working mode of the system once it comprises trained models. The software, after photo acquisition, waits for the operator classification. After, it allows the operator to view the results of the predictive models. The operator can then decide whether to confirm his or her initial assessment or perform a class change based on the results of the models. Recording of these case histories can allow the network to be retrained in future (continuous learning).

Table 4.31.: Performance evaluation on Aesthetic Quality Control Dataset. Experimental results: average over 30 executions expressed with mean(std). The performance of each model is measured on the test set. The best results are in bold. Stars indicate whether the best performing algorithm is significantly better than state-of-the-art approaches (**: $\alpha = 0.05$). DA: Data Augmentation; Act: activation function; CCE: Categorical Cross-Entropy; BCE: Binary Cross-Entropy; R_{QWK}: averaged rank for QWK metric; R_{MAE}: averaged rank for the MAE metric

Model	DA	Act	Loss	β	CCR↑	1-off↑	QWK↑	MAE↓	$\mathbf{R_{QWK}}\downarrow$	$\mathbf{R_{MAE}}\downarrow$
GLB-NOM [104]	yes	Softmax	CCE		0.504(0.038)	0.803(0.029)	0.870(0.047)**	0.809(0.119)**	16.400(6.015)	21.200(4.374)
GLB-NOM [104]	no	Softmax	CCE		0.528(0.033)	0.817(0.023)	0.882(0.022)**	0.756(0.069)**	11.933(6.017)	18.567(5.456)
GLB-OBD [72]	yes	Sigmoid	MAE		0.410(0.024)	0.783(0.023)	0.791(0.041)**	0.976(0.074)**	25.167(0.531)	25.233(0.430)
GLB-OBD [72]	no	Sigmoid	MAE		0.413(0.023)	0.772(0.024)	0.768(0.048)**	1.017(0.079)**	25.333(1.295)	25.667(0.606)
GLB-CLM [93]	yes	Logit	QWK		0.432(0.030)	0.835(0.020)	0.904(0.011)**	0.785(0.051)**	15.833(5.240)	9.467(4.688)
GLB-CLM [93]	no	Logit	QWK		0.426(0.028)	0.834(0.022)	0.902(0.011)**	0.790(0.049)**	16.500(4.424)	10.600(5.090)
LCPN-NOM [158, 159, 160]	yes	Softmax	CCE		0.529(0.029)	0.835(0.021)	0.898(0.011)**	0.709(0.052)**	7.233(4.408)	13.600(5.934)
LCPN-NOM [158, 159, 160]	no	Softmax	CCE		**0.538**(0.029)	0.840(0.020)	0.903(0.010)**	0.688(0.048)**	5.167(3.343)	9.367(5.034)
LCPN-OBD	yes	Sigmoid	MAE		0.529(0.032)	0.845(0.026)	0.908(0.016)**	0.684(0.066)**	5.533(3.803)	7.633(6.278)
LCPN-OBD	no	Sigmoid	MAE		0.524(0.041)	0.847(0.024)	0.906(0.015)**	0.690(0.070)**	6.500(5.316)	8.667(6.557)
LCPN-CLM	yes	Logit	QWK		0.446(0.045)	0.837(0.030)	0.896(0.019)**	0.782(0.082)**	15.100(6.970)	13.567(6.334)
LCPN-CLM	no	Logit	QWK		0.446(0.044)	0.826(0.030)	0.891(0.017)**	0.802(0.081)**	16.867(5.835)	16.133(5.551)
MTL-OBD [161]	yes	Sigmoid	MAE	0.8	0.475(0.023)	0.821(0.023)	0.897(0.012)**	0.781(0.048)**	15.333(4.147)	13.633(4.165)
MTL-OBD [161]	no	Sigmoid	MAE	0.8	0.490(0.023)	0.828(0.026)	0.902(0.014)**	0.755(0.054)**	11.500(4.224)	10.567(5.793)
MTL-CLM [161]	yes	Logit	QWK	0.5	0.460(0.025)	0.820(0.022)	0.892(0.012)**	0.801(0.052)**	17.733(4.968)	16.800(4.781)
MTL-CLM [161]	no	Logit	QWK	0.5	0.469(0.023)	0.817(0.023)	0.893(0.012)**	0.794(0.047)**	16.233(4.688)	15.800(4.649)
MTL-OBD$_{loc}$	yes	Sigmoid	MAE	0.8	0.474(0.026)	0.819(0.026)	0.896(0.014)**	0.786(0.058)**	15.633(4.295)	14.233(3.812)
MTL-OBD$_{loc}$	no	Sigmoid	MAE	0.8	0.491(0.025)	0.823(0.022)	0.901(0.013)**	0.757(0.053)**	11.600(3.616)	10.067(4.009)
MTL-CLM$_{loc}$	yes	Logit	QWK	0.5	0.456(0.030)	0.808(0.023)	0.888(0.014)**	0.820(0.051)**	19.700(3.395)	18.533(3.560)
MTL-CLM$_{loc}$	no	Logit	QWK	0.5	0.463(0.031)	0.811(0.024)	0.890(0.016)**	0.803(0.062)**	17.467(5.342)	17.500(4.762)
HMCN [111]	yes	Sigmoid	BCE	0.5	0.524(0.034)	0.823(0.022)	0.890(0.016)**	0.737(0.058)**	10.633(5.720)	16.233(6.388)
HMCN [111]	no	Sigmoid	BCE	0.5	0.532(0.030)	0.824(0.023)	0.887(0.020)**	0.733(0.0691)**	9.500(5.722)	16.667(6.194)
HOBD	yes	Sigmoid	MAE	0.5	0.534(0.028)	**0.868**(0.200)	**0.921**(0.009)	**0.635**(0.048)	**2.100**(1.213)	**2.000**(1.203)
HOBD	no	Sigmoid	MAE	0.5	0.536(0.028)	0.864(0.018)	0.921(0.008)	0.639(0.044)	2.533(1.676)	2.200(1.448)
HCLM	yes	Logit	QWK	0.5	0.426(0.036)	0.833(0.029)	0.905(0.011)	0.795(0.065)	16.367(6.049)	8.267(4.017)
HCLM	no	Logit	QWK	0.5	0.433(0.032)	0.834(0.021)	0.905(0.011)	0.783(0.055)	14.667(5.274)	8.700(5.167)

Chapter 5.

Discussions

In this Section the author discusses and answers the research questions provided in Section 1.2. In particular, in Section 5.1 the experimental results and findings about 1.2.1 are discussed, and in Section 5.2 research questions raised in Section 1.2.2 are treated.

5.1. Predictive Quality Control problem

The aim of this study was to design and develop a DSS for predicting the processing quality during the machining of a tool with the purpose of implementing PdM actions. In this work we firstly presented a ML based solution for solving a PdM task in an unexplored application, that is the RUL estimation for ATM devices. Then, we introduced and tested a DSS for solving the machining quality prediction in a real industrial use case, demonstrating the capabilities of the proposed approach in supporting the human operator during a PQC task. In particular, the proposed approach led to the following answers:

i) How can supervised ML model be applied to predict the processing quality of a tool starting from a machine sensors raw data?
Starting from industrial Big Data scenario, it was highlighted the importance of design i) a feature engineering stage, performed in collaboration with domain expert maintainers, to ensure the building of a representative dataset and ii) a feature extraction strategy for transforming raw data into numerical features that can be processed while preserving the information in the original data set. In particular, the Trapezoidal Numerical Integration (TNI) was performed to compute a Key Performance Indicator (KPI) for each processing parameter during each production cycle. The feature extraction was based on specific topics published in the MQTT broker and collected on the lower and upper levels of the production system. In particular, lower level topics were used to extract salient KPI predictors for feeding ML model.

ii) How is it possible to obtain and manage reliable data annotation?
The upper-level topics reflect the ground-truth variables that are closely related to processing quality and thus to productivity losses and maintenance issues. These condi-

tion monitoring data were acquired by a robotic part loading system for 3D coordinate measuring machine. This system allows the acquisition of high-quality labeled data, which are suitable for a supervised ML approach.

iii) Is it feasible to ensure at the same time high predictive performance and model interpretability?
Taking into account the achieved experimental results in Section 3.2.3, we demonstrated the effectiveness of our theoretical frameworks into a real industrial environment. In fact, our DSS approach based on RF model was demonstrated to be the best trade-off between predictive performance, computation effort, and interpretability. In particular, the interpretability of the proposed RF model was measured according to the feature/permutation importance (see Fig. 3.10).

iv) Does the proposed ML algorithm outperform standard algorithms widely used in literature?
We proved how RF algorithm is suitable both to predict whether a machine will fail or not in the next 6 days (classification task) and to estimate the error % related to the machining quality (regression task). In particular, as reported in Section 3.2.3 - predictive performance, R^2 score distribution of RF is significantly higher ($p < .05$) than state of the art ML based regression models and DL based regression models. Despite the predictive performance of RF is similar to XBG, the training and validation of RF model are significantly faster.

v) How can the proposed algorithm be integrated in a DSS to provide suitable feedback for supporting human operator during production stages?
A GUI interface was created to display the predicted error % and to provide a timely indication to the machine operator when the error exceeds a certain tolerance threshold. Moreover, the average value of the KPI predictors and their trend over a specific tool are shown. This GUI interface may empower the overall machining quality process by supporting the operators to (i) predict alarm situation (i.e. significant machining error) and (ii) interpret and localize the source of the error by focusing on the average and temporal value of the most discriminative KPI predictors.

vi) How should DSS be designed to support real-time data acquisition and model inference?
The ML algorithm was integrated in a scalable cloud-based architecture, which is the main core of the DSS (see Fig. 3.8). This environment can ensure hardware and software isolation, flexibility, and inter-dependencies between machine. These properties are suitable for working with more machines at the same time and provide the capability to be scaled up to collect a huge amount of data from different interconnected machines.

5.1.1. Limitations and Future work

A current limitation is represented by the employed ML model, which is re-trained every time from scratch when a certain amount of new data is stored. As future work, we aim to integrate a fully-automated incremental learning procedure by updating continuously the model parameter [163]. The new data continuously acquired and stored over time in the proposed cloud framework can be used to refine ML model and improve its predictive performance according to an incremental learning procedure. As a further limitation, the proposed model only works for certain tools installed on the processing machines. However, as future work, for the proposed ML model we aim to improve the generalization performance of the proposed approach across different tools and types of processing. In addition, meta-heuristic algorithms [164, 165] could be implemented to help in selecting the optimal hyperparameters for ML model to improve the stability and the testing predictive performance [166, 41]. For the full applicability of PdM in all the company's tasks, another future work direction could be addressed to build integrated cost-benefit models that include the impact and the benefit of our approach on the entire asset management of the company [167]. However, approaching the RUL estimation as a nominal classification task leads to lose some relevant information about the risk of failure. Future work could be addressed by considering ML ordinal methodologies and recurrent neural networks to embrace both ordinal and temporal constraints in raw time series data.

5.2. Aesthetic Quality Control problem

The aim of this study was to design and develop a DSS for assessing the aesthetic classification of wooden stocks with the purpose of supporting the human operator in the final decision. We introduced and tested several DL methodologies for solving the challenges in this AQC task. In particular, the proposed approaches led to the following answers:

i) How can supervised DL model be applied to perform a classification task based on qualitative aesthetic properties of a material?
Being trained on examples annotated by experts rather than composed of strict descriptive rules, as first we demonstrated how a nominal state of the art DL approach, i.e. VGG-16 network with an ordinal categorical cross-entropy (CCE) loss, with the proper training procedure, is able to generalize across different unseen rifle stocks, automatizing and standardizing the overall AQC process. In particular, in Section 4.2.3 it was demonstrated how VGG-16 outperforms other state of the art classification models in terms of CCR, Recall, Precision and F1 score.

ii) How is it possible to detect and mitigate unwanted bias in data?
Performing a Cramer's analysis correlation, it was demonstrated how a significant correlation was present between VGG-16 predictions and rifles series, also greater than

between rifle series and ground-truth classes. This was also confirmed by exploring the saliency map of the VGG-16 model, demonstrating how the presence of bias in the dataset and in the learning procedure could be a disruptive finding that may lead to an overestimation of the performance. The proposed HUVGG-16 solution based on two-stages hierarchical networks (see Fig. 4.9), even if specific for this use case, allows to mitigate the detected bias by learning the characteristics that properly describe the quality of wood, rather than other confound characteristics.

Moving toward a more scalable solution, after a image preprocessing stage, a voting ensamble approach comprising of standard ML/DL algorithms (based on different data features) was proposed, for combining different models predictions according to a majority vote. Despite the lower performance (0.509 CCR, Table 4.6), the effectiveness of this method was proven with a validation stage performed by human operator.

iii) How can errors between distant classes be minimized?
A DL ordinal methodology, specifically tailored for solving the proposed AQC task, was introduced, based on CLMs and VGG-16 as a feature extractor. The proposed method driven by ordinal constraints was properly conceived to model the natural ordinal structure of the dataset classes while penalizing the misclassification errors that are far from the correct label. The introduction of the slope parameter allows to model the transient between CLM functions for each learnable ordinal threshold.

The higher performance obtained by the CLM VGG-16 for quality class prediction with respect to a nominal and other ordinal DL approaches suggests how the proposed method represents a valuable solution for automatizing the overall AQC procedure. Moreover, the experimental findings shown in Section 4.4.3 demonstrated how a standard CCE together with CLM can be sufficient to model the ordinal structure of the label, also without requiring the minimization of an ordinal loss. This is also in line with recent findings in the ordinal classification literature [168]. As additional gain, the ordinal constraint allows the network to better learn the characteristics that properly describe the quality of rifle (i.e. wood grains), rather than other confounds.

iv) How is it possible to mitigate noise or errors in labeling process?
It was proposed to apply the L_p norm into a previously proposed exponential regularised loss for obtaining soft labels with a more flexible distribution for an ordinal classification problem. Comparing with a baseline approach, which uses the standard categorical cross-entropy loss and the softmax at the output of the model, and also with using state of the art regularisation methods, the results in Section 4.5.4 demonstrated that the proposed alternative achieved the values for QWK and MS, and the second-best result for MAE. Also, the statistical tests demonstrated the robustness and the effectiveness of the proposed approach and the gain with respect to previous alternatives. Moreover, as regards the AQC task we aimed to solve, we achieved a high QWK value, which implies small classification errors, where most of them occur in the adjacent classes.

v) Is it feasible to design a DL methodology for exploiting ordinal and/or hierarchical properties of the dataset?
For exploiting both the ordinal and hierarchical properties of our AQC dataset, firstly a hierarchical approach was proposed for learning ordinal classes in two separate phases (see Fig. 4.24). In the first one, a single model was used to predict the macro label of each pattern. In the second one, one model was used for each macro label to predict the corresponding micro class. This method was combined with an ordinal loss regularisation and an output layer based on the CLM to encourage, at the same time, the ordinal classification. Different alternatives were tested with three different model architectures and the experimental results showed that the hierarchical methods obtained the best results for most of the metrics and architectures. In general terms, the hierarchical approaches obtained better results than other state of the art non-hierarchical approaches. The main benefit of the described approach is that it improves the performance of this kind of tasks at the same time that it simplifies the problem by dividing the classification task into multiple models.

In order to overcome the limitation of learning separated multiple models, then we proposed novel approaches where a single model is used for solving the overall problem, which simultaneously learn hierarchical and ordinal constraints (i.e. HCLM and HOBD). These approaches are able to model the ordinal structure within different hierarchical levels of the labels. Considering the experimental comparisons reported in Section 4.7.3, the proposed HOBD proved to be effective in dealing with these hierarchical-ordinal tasks, by overcoming all the other state of the art nominal, hierarchical and ordinal approaches.

vi) How can the proposed algorithm be integrated in a DSS to provide suitable feedback for supporting human operator during final QC decisions?
A DSS platform comprised of acquisition bench, GUI interface and cloud architecture, as described in Section 4.8, can make the QC procedure more reliable, fast, standardized, and scalable at the same time. The integration of the proposed DL methodologies in the cloud environment represents the main core of a DSS for solving the AQC task. The setup of the system is fairly simple, as it only requires the acquisition box, which takes the pictures of the items, and the DL based DSS interfaced with a GUI. Thanks to these features, it helps the human operator in taking the final decision, significantly reducing the inference time.

5.2.1. Limitations and Future work

As regards the exploiting of structural properties of the dataset, given that the proposed ordinal-hierarchical approaches improved the state of the art methodologies, in future work these methodologies can be extended to deal with other tasks respect to the AQC problem solved in this thesis work. In these terms, new DSS for other real problems can be developed. Possible applications include, but are not limited, to

any other type of QC problem, also related to the overall quality of a product from the engineering point of view. The only limitation of the proposed approach is that the labels must follow a natural order and must also be decomposed hierarchically. However, there are some problems where the hierarchical structure is not given a priori, but can be inferred from the characteristics of data. In this context, our approach could be extended via meta-learning formulation to simultaneously customize and preserve ordinal and hierarchical task knowledge [169].

Moreover, the proposed approach does not take into account the presence of sparse or missing labels. In different application scenarios, some classes in the hierarchy could be potentially missing or not available. For example, in cross-domain recommendation, providing reliable recommendations to newly joined users (so-called cold-start users) is a challenging task, i.e. the unlabeled data are easily available and large while labeled data are difficult to collect. Future work may be handled to generalize the proposed methodology to weakly-supervised and semi-supervised settings scenarios, using self-learning [170] and incremental learning approaches [171].

Another interesting future direction includes the possibility to model inter-operator variability by providing multiple annotations from different operators for the same image. This direction includes the possibility to design a MTL deep ordinal approach to simultaneously monitor correlation and variability among raters. Moreover, the regularised loss function with the ordinal output model described in this work can be applied to more complex CNN models, which could lead to enhanced performance.

Finally, although the bias problem is not always present or easily detectable, other bias mitigation approaches can be investigated in future work, such as the adversarial learning method [172].

Chapter 6.

Conclusions

6.1. Conclusive remarks

The main contribution of this thesis is the design and implementation of Decision Support Systems, based on different Machine Learning and Deep Learning algorithms, for supporting the human operator during Quality Control procedure. In particular, the role of DSS to bridge the gap between smart factory advances and the application of these technologies in a Quality 4.0 real scenario has been highlighted, facing several challenges in this context. Two problems were formulated and answered:

- Design and implementation of a DSS in Predictive Quality Control for predicting the processing quality and anomaly situations during the machining of a tool;
- Design and implementation of a DSS in Aesthetic Quality Control for evaluating the aesthetic properties of a material for the manufactured product.

The effectiveness of the proposed approaches was proven on two real-world industrial use cases. The DSSs described in this thesis have been developed and are currently up and running in the company, representing a valuable technological transfer between the University and industrial world. Implementing these industrial systems for monitoring the health and quality of the instrumentation/products/materials enables manufacturers to support the technicians during the process while reducing resource costs, intrinsic variability and improving productivity.

The thesis has presented reviews, perspectives, new methods and applications in the field of ML/DL methods for Industry 4.0 scenario, leading to the publication of 8 scientific papers. The described contributions reflect a significant advancement both in the state of the art methodologies as well as for the application scenarios.

Chapter 1 presents an in-depth analysis of the Quality Control tasks and the corresponding importance of DSS in the Industry 4.0 scenario, highlighting the challenges present in this context. Chapter 2 describes the state of art, with a specific focus on the methodologies employed in literature to solve the identified challenges and the main contributions of this thesis with respect to the research gaps. Chapter 3 and Chapter 4

describe the faced PQC and AQC problems respectively, treating the real use cases and the proposed ML/DL based DSS frameworks, demonstrating how to solve the related domain challenges. Chapter 5 discusses the obtained results and revisits the scientific contributions of this thesis in terms of new methodologies and knowledge created to benefit Quality 4.0 scenario and their validity in real-world industrial applications.

6.2. Future perspectives

To conclude, it has been done the first step in introducing these methodologies for real and challenging Quality Control applications. These approaches can be extended to deal with other tasks respect to the PQC and AQC problems solved in this thesis work. In fact, the generalization and extension of Quality 4.0 solutions is a relevant issue in this context. As it has been noticed also from the state of the art analysis, the focus of applied research tends to be on implementing specific solutions within the boundaries of manufacturing companies. The problems of horizontal and vertical data integration and methodologies extension are due to the lack of data fusion from different sources (e.g. ERP, MES and PLM) and data sharing among companies [13]. To this end, research directions in this field focus on generalizing predictive models for scalable solutions, developing secure and standardized communication protocols, and elaborating advanced data mining procedures. In this sense, also the introduction and release of open benchmark datasets could attract the Machine Learning community in quest toward advancing the state of the art and generalizing methodologies.

Appendix A.

Chapter 4.6 - Performance evaluation and Results

A.1. Boxplots of VGG-16 and DenseNet-121

The boxplots corresponding to the results of the VGG-16 and DenseNet-121 architectures are shown. Figure A.1 shows the boxplots for each metric for the VGG-16 architecture, while Figure A.2 shows the boxplots for the DenseNet-121 architecture.

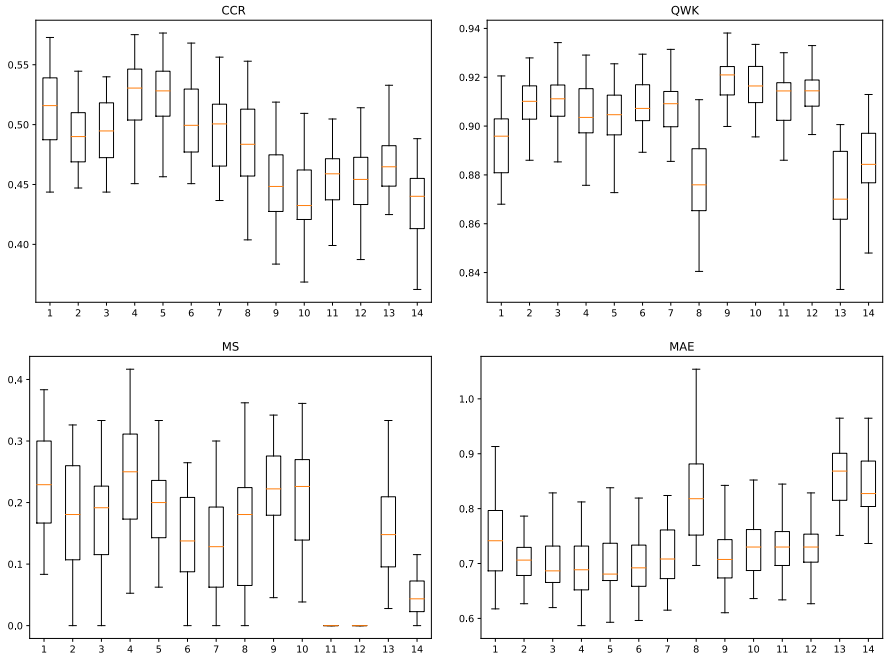


Figure A.1.: Boxplots for all the test metrics using the VGG-16 architecture. Methods are identified with the numbers defined in Table 4.21.

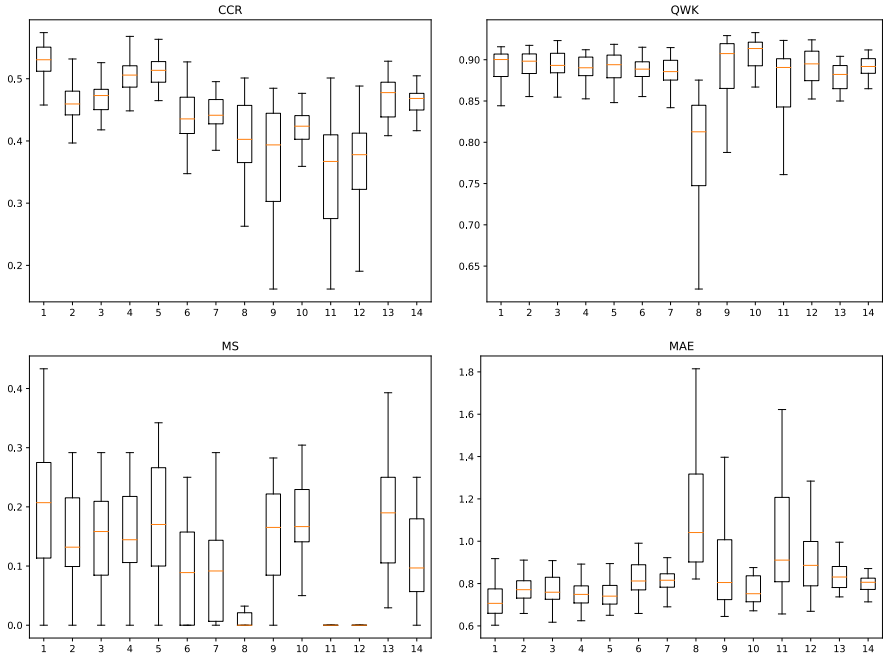


Figure A.2.: Boxplots for all the test metrics using the DenseNet-121 architecture. Methods are identified with the numbers defined in Table 4.23.

A.2. Model architectures statistical comparison

In addition, to complete the statistical analysis performed in Section 4.6.3, the three model architectures considered in our work are compared using statistical tests. During the general statistical analysis, the ANOVA II tests reported significant differences between the architectures for all the metrics. Therefore, a post-hoc HSD Tukey's test is performed for each of the metrics.

First of all, the statistical test was performed for the accuracy metric. The results of the post-hoc test are shown in Table A.1. The two best architectures for the CCR metric are the VGG-16 and the ResNet-101. They are significantly better than the DenseNet-121 model.

For the QWK metric, the same analysis is performed. The results are shown in Table A.2, and, this time, they show that all the architectures are significantly different. The DenseNet-121 model is, again, the worst, but, in this case, the VGG-16 and the ResNet-101 show significant differences and the residual network is better.

Then, the MS metric values are analysed. The results of the post-hoc test are shown in Table A.3. In this case, the conclusions are the same that were obtained for the accuracy metric. The ResNet-101 and the VGG-16 architectures obtained the best results and are significantly better than the DenseNet-121.

Table A.1.: Results of the post-hoc HSD Tukey's test for the model and the CCR metric.

	Subsets		
Model	1	2	3
DenseNet-121	0.4393		
ResNet-101		0.4626	
VGG-16			0.4797
p-values	1.000	1.000	1.000

Table A.2.: Results of the post-hoc HSD Tukey's test for the model and the QWK metric.

	Subsets	
Model	1	2
DenseNet-121	0.8776	
VGG-16		0.8987
ResNet-101		0.9006
p-values	1.000	0.624

Table A.3.: Results of the post-hoc HSD Tukey's test for the model and the MS metric.

	Subsets	
Model	1	2
DenseNet-121	0.1190	
VGG-16		0.1519
ResNet-101		0.1562
p-values	1.000	0.734

Table A.4.: Results of the post-hoc HSD Tukey's test for the model and the MAE metric.

	Subsets		
Model	1	2	3
VGG-16	0.7443		
ResNet-101		0.7736	
DenseNet-121			0.8565
p-values	1.000	1.000	1.000

Finally, the MAE metric is analysed. The results are shown in Table A.4. Again, the results are similar: the ResNet-101 and the VGG-16 models obtained the best results. The differences between the results obtained using these two architectures and the results obtained using the DenseNet-121 model are significant.

Therefore, from these tests, we can conclude that the best model architecture is ResNet-101, given that it is significantly better than the other alternatives regarding the QWK metric, and it is as good as the VGG-16 model considering the other metrics.

Appendix B.

Other publications

The following publications, which are only partially related to the topic of the doctorate and will not be discussed in the thesis, result from intra- and inter-VRAI research group collaborations:

- **Rosati, R.**, Romeo, L., Silvestri, S., Marcheggiani, F., Tiano, L., & Frontoni, E. (2020). Faster R-CNN approach for detection and quantification of DNA damage in comet assay images. Computers in Biology and Medicine, 123, 103912.
- **Rosati, R.**, Romeo, L., Goday, C. A., Menga, T., & Frontoni, E. (2020). Machine Learning in Capital Markets: Decision Support System for Outcome Analysis. IEEE Access, 8, 109080-109091.
- Pazzaglia, G., Martini, M., **Rosati, R.**, Romeo, L., & Frontoni, E. (2022). A Deep Learning-Based Approach for Automatic Leather Classification in Industry 4.0. In International Conference on Pattern Recognition (pp. 662-674). Springer, Cham.
- Manilii, A., Lucarelli, L., **Rosati, R.**, Romeo, L., Mancini, A., & Frontoni, E. (2021). 3D Human Pose Estimation Based on Multi-Input Multi-Output Convolutional Neural Network and Event Cameras: A Proof of Concept on the DHP19 Dataset. In International Conference on Pattern Recognition (pp. 14-25). Springer, Cham.
- Pierdicca, R., Tonetto, F., Mameli, M., **Rosati, R.**, & Zingaretti, P. (2022). Can AI Replace Conventional Markerless Tracking? A Comparative Performance Study for Mobile Augmented Reality Based on Artificial Intelligence. In International Conference on Extended Reality (pp. 161-177). Springer, Cham.
- Pauls, A., Romeo, L., **Rosati, R.** & Kuznetsov, A. (2022). Deep Learning Model for Detecting Copy-Move Attack in Images: Testing and Verification. In Next Generation Cybersecurity Systems and Applications 2022.
- Kuznetsov, A., Luhanko, N., Romeo, L. & **Rosati, R.** (2022). Deep Learning Based Image Steganalysis. In 2022 IEEE International Conference on Problems of Infocommunications Science and Technology.

- Kuznetsov, A., Zakharov, D., Romeo, L. & **Rosati, R.** (2022). Deep Learning Based Fuzzy Extractor for Generating Strong Keys from Biometric Face Images. In 2022 IEEE International Conference on Problems of Infocommunications Science and Technology.
- Kuznetsov, A., Kvaratskheliia, N., Romeo, L. **Rosati, R.**, Maranesi, A. & Muscatello, A. (2022). Deep Learning Based Face Liveliness Detection. In 2022 IEEE International Conference on Problems of Infocommunications Science and Technology.

Bibliography

[1] C. Yin, J. Xi, R. Sun, and J. Wang, "Location privacy protection based on differential privacy strategy for big data in industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3628–3636, 2017.

[2] J. Lee, H.-A. Kao, and S. Yang, "Service innovation and smart analytics for industry 4.0 and big data environment," *Procedia cirp*, vol. 16, pp. 3–8, 2014.

[3] M. Javaid, A. Haleem, R. Pratap Singh, and R. Suman, "Significance of quality 4.0 towards comprehensive enhancement in manufacturing sector," *Sensors International*, vol. 2, p. 100109, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666351121000309

[4] S. K. Jagatheesaperumal, M. Rahouti, K. Ahmad, A. Al-Fuqaha, and M. Guizani, "The duo of artificial intelligence and big data for industry 4.0: Applications, techniques, challenges, and future research directions," *IEEE Internet of Things Journal*, vol. 9, no. 15, pp. 12 861–12 885, 2022.

[5] N. Sakib and T. Wuest, "Challenges and opportunities of condition-based predictive maintenance: A review," *Procedia CIRP*, vol. 78, pp. 267–272, 2018.

[6] M. Calabrese, M. Cimmino, F. Fiume, M. Manfrin, L. Romeo, S. Ceccacci, M. Paolanti, G. Toscano, G. Ciandrini, A. Carrotta, M. Mengoni, E. Frontoni, and D. Kapetis, "Sophia: An event-based iot and machine learning architecture for predictive maintenance in industry 4.0," *Information*, vol. 11, no. 4, p. 202, 2020.

[7] L. Romeo, J. Loncarski, M. Paolanti, G. Bocchini, A. Mancini, and E. Frontoni, "Machine learning-based design support system for the prediction of heterogeneous machine parameters in industry 4.0," *Expert Systems with Applications*, vol. 140, p. 112869, 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417419305792

[8] S. Doltsinis, P. Ferreira, M. M. Mabkhot, and N. Lohse, "A decision support system for rapid ramp-up of industry 4.0 enabled production systems," *Computers in Industry*, vol. 116, p. 103190, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166361519306876

[9] J. P. U. Cadavid, S. Lamouri, B. Grabot, R. Pellerin, and A. Fortin, "Machine learning applied in production planning and control: A state-of-the-art in the era of industry 4.0," *Journal of Intelligent Manufacturing*, pp. 1–28, 2020.

[10] C. A. Escobar and R. Morales-Menendez, "Machine learning techniques for quality control in high conformance manufacturing environment," *Advances in Mechanical Engineering*, vol. 10, no. 2, p. 1687814018755519, 2018.

[11] T. Wang, Y. Chen, M. Qiao, and H. Snoussi, "A fast and robust convolutional neural network-based defect detection model in product quality control," *The International Journal of Advanced Manufacturing Technology*, vol. 94, no. 9-12, pp. 3465–3471, 2018.

[12] A. Angelopoulos, E. T. Michailidis, N. Nomikos, P. Trakadas, A. Hatziefremidis, S. Voliotis, and T. Zahariadis, "Tackling faults in the industry 4.0 era—a survey of machine-learning solutions and key aspects," *Sensors*, vol. 20, no. 1, p. 109, 2019.

[13] D. Powell, M. C. Magnanini, M. Colledani, and O. Myklebust, "Advancing zero defect manufacturing: A state-of-the-art perspective and future research directions," *Computers in Industry*, vol. 136, p. 103596, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166361521002037

[14] F. Psarommatis, G. May, P.-A. Dreyfus, and D. Kiritsis, "Zero defect manufacturing: state-of-the-art review, shortcomings and future directions in research," *International journal of production research*, vol. 58, no. 1, pp. 1–17, 2020.

[15] K. Stylidis, C. Wickman, and R. Söderberg, "Perceived quality of products: a framework and attributes ranking method," *Journal of Engineering Design*, vol. 31, no. 1, pp. 37–67, 2020.

[16] G. May and D. Kiritsis, "Zero defect manufacturing strategies and platform for smart factories of industry 4.0," in *International Conference on the Industry 4.0 model for Advanced Manufacturing*. Springer, 2019, pp. 142–152.

[17] B. Caiazzo, M. Di Nardo, T. Murino, A. Petrillo, G. Piccirillo, and S. Santini, "Towards zero defect manufacturing paradigm: A review of the state-of-the-art methods and open challenges," *Computers in Industry*, vol. 134, p. 103548, 2022.

[18] L. Liu, Q. Guo, D. Liu, and Y. Peng, "Data-driven remaining useful life prediction considering sensor anomaly detection and data recovery," *IEEE access*, vol. 7, pp. 58 336–58 345, 2019.

[19] A. Petrillo, A. Picariello, S. Santini, B. Scarciello, and G. Sperlí, "Model-based vehicular prognostics framework using big data architecture," *Computers in Industry*, vol. 115, p. 103177, 2020.

[20] D. Tabernik, S. Šela, J. Skvarč, and D. Skočaj, "Segmentation-based deep-learning approach for surface-defect detection," *Journal of Intelligent Manufacturing*, vol. 31, no. 3, pp. 759–776, 2020.

[21] K. He, Q. Zhang, and Y. Hong, "Profile monitoring based quality control method for fused deposition modeling process," *Journal of Intelligent Manufacturing*, vol. 30, no. 2, pp. 947–958, 2019.

[22] H. Dong, K. Song, Y. He, J. Xu, Y. Yan, and Q. Meng, "Pga-net: Pyramid feature fusion and global context attention network for automated surface defect detection," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 12, pp. 7448–7458, 2019.

[23] F. E. Horita, J. P. de Albuquerque, V. Marchezini, and E. M. Mendiondo, "Bridging the gap between decision-making and emerging big data sources: An application of a model-based framework to disaster management in brazil," *Decision Support Systems*, vol. 97, pp. 12–22, 2017.

[24] L. Romeo, J. Loncarski, M. Paolanti, G. Bocchini, A. Mancini, and E. Frontoni, "Machine learning-based design support system for the prediction of heterogeneous machine parameters in industry 4.0," *Expert Systems with Applications*, vol. 140, p. 112869, 2020.

[25] M. Khatun and S. J. Miah, "Design of a decision support system framework for small-business managers: A context of b2c e-commerce environment," in *2016 Future Technologies Conference (FTC)*. IEEE, 2016, pp. 1274–1281.

[26] A. Karmarkar and N. Gilke, "Fuzzy logic based decision support systems in variant production," *Materials Today: Proceedings*, vol. 5, no. 2, pp. 3842–3850, 2018.

[27] M. A. Villegas and D. J. Pedregal, "Supply chain decision support systems based on a novel hierarchical forecasting approach," *Decision Support Systems*, vol. 114, pp. 29–36, 2018.

[28] EU, "Ethics guidelines for trustworthy ai," https://ec.europa.eu/digital-single-market/en/news/ethics-guidelines-trustworthy-ai, 2019.

[29] F. Sgarbossa, E. H. Grosse, W. P. Neumann, D. Battini, and C. H. Glock, "Human factors in production and logistics systems of the future," *Annual Reviews in Control*, vol. 49, pp. 295–305, 2020.

[30] J. Dalzochio, R. Kunst, E. Pignaton, A. Binotto, S. Sanyal, J. Favilla, and J. Barbosa, "Machine learning and reasoning for predictive maintenance in industry 4.0: Current status and challenges," *Computers in Industry*, vol. 123, p. 103298, 2020.

[31] S. Vollert, M. Atzmueller, and A. Theissler, "Interpretable machine learning: A brief survey from the predictive maintenance perspective," in *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2021, pp. 01–08.

[32] H. Tercan and T. Meisen, "Machine learning and deep learning based predictive quality in manufacturing: a systematic review," *Journal of Intelligent Manufacturing*, pp. 1–27, 2022.

[33] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on knowledge and data engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[34] K. A. Houser, "Can ai solve the diversity problem in the tech industry: Mitigating noise and bias in employment decision-making," *Stan. Tech. L. Rev.*, vol. 22, p. 290, 2019.

[35] S. Addanke, M. V. Krishna, K. Pradeep, K. P. Jayant, K. Ilyassova, and K. L. Sainath, "Machine learning on the role of eliminating human error on the manufacturing industry," in *2022 6th International Conference on Trends in Electronics and Informatics (ICOEI)*. IEEE, 2022, pp. 1–6.

[36] W. Dai, K. Yoshigoe, and W. Parsley, "Improving data quality through deep learning and statistical models," in *Information Technology-New Generations*. Springer, 2018, pp. 515–522.

[37] M. Bertolini, D. Mezzogori, M. Neroni, and F. Zammori, "Machine learning for industrial applications: A comprehensive literature review," *Expert Systems with Applications*, vol. 175, p. 114820, 2021.

[38] L. Abualigah, D. Yousri, M. Abd Elaziz, A. A. Ewees, M. A. Al-qaness, and A. H. Gandomi, "Aquila optimizer: A novel meta-heuristic optimization algorithm," *Computers & Industrial Engineering*, vol. 157, p. 107250, 2021.

[39] L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz, and A. H. Gandomi, "The arithmetic optimization algorithm," *Computer methods in applied mechanics and engineering*, vol. 376, p. 113609, 2021.

[40] B. Shahbazi and S. H. A. Rahmati, "Developing a flexible manufacturing control system considering mixed uncertain predictive maintenance model: a simulation-based optimization approach," in *Operations Research Forum*, vol. 2, no. 4. Springer, 2021, pp. 1–43.

[41] P. Shah, R. Sekhar, A. J. Kulkarni, and P. Siarry, *Metaheuristic algorithms in industry 4.0*. CRC Press, 2021.

[42] Y. Xu, Y. Sun, X. Liu, and Y. Zheng, "A digital-twin-assisted fault diagnosis using deep transfer learning," *IEEE Access*, vol. 7, pp. 19 990–19 999, 2019.

[43] P. Adhikari, H. G. Rao, and M. Buderath, "Machine learning based data driven diagnostics & prognostics framework for aircraft predictive maintenance," in *10th International Symposium on NDT in Aerospace*, 2018.

[44] C. P. Gatica, M. Koester, T. Gaukstern, E. Berlin, and M. Meyer, "An industrial analytics approach to predictive maintenance for machinery applications," in *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, 2016, pp. 1–4.

[45] S. Khan and T. Yairi, "A review on the application of deep learning in system health management," *Mechanical Systems and Signal Processing*, vol. 107, pp. 241–265, 2018.

[46] L. Jin, F. Wang, and Q. Yang, "Performance analysis and optimization of permanent magnet synchronous motor based on deep learning," in *20th International Conference on Electrical Machines and Systems (ICEMS)*, 2017, pp. 1–5.

[47] A. Rivas, J. M. Fraile, P. Chamoso, A. González-Briones, I. Sittón, and J. M. Corchado, "A predictive maintenance model using recurrent neural networks," in *International Workshop on Soft Computing Models in Industrial and Environmental Applications*. Springer, 2019, pp. 261–270.

[48] A. Cachada, J. Barbosa, P. Leitño, C. A. Gcraldcs, L. Deusdado, J. Costa, C. Teixeira, J. Teixeira, A. H. Moreira, P. M. Moreira, and L. Romero, "Maintenance 4.0: Intelligent and predictive maintenance system architecture," in *2018 IEEE 23rd international conference on emerging technologies and factory automation (ETFA)*, vol. 1. IEEE, 2018, pp. 139–146.

[49] R. S. Peres, A. D. Rocha, P. Leitao, and J. Barata, "Idarts–towards intelligent data analysis and real-time supervision for industry 4.0," *Computers in Industry*, vol. 101, pp. 138–146, 2018.

[50] Z. Li, Y. Wang, and K.-S. Wang, "Intelligent predictive maintenance for fault diagnosis and prognosis in machine centers: Industry 4.0 scenario," *Advances in Manufacturing*, vol. 5, no. 4, pp. 377–387, 2017.

[51] Z. Liu, C. Jin, W. Jin, J. Lee, Z. Zhang, C. Peng, and G. Xu, "Industrial ai enabled prognostics for high-speed railway systems," in *2018 IEEE International Conference on Prognostics and Health Management (ICPHM)*, 2018, pp. 1–8.

[52] Z. C. Lipton, "The mythos of model interpretability," *Queue*, vol. 16, no. 3, pp. 31–57, 2018.

[53] B. Schmidt and L. Wang, "Predictive maintenance of machine tool linear axes: A case from manufacturing industry," *Procedia manufacturing*, vol. 17, pp. 118–125, 2018.

[54] C. Zhou and C.-K. Tham, "Graphel: A graph-based ensemble learning method for distributed diagnostics and prognostics in the industrial internet of things," in *2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS)*. IEEE, 2018, pp. 903–909.

[55] C. M. Carbery, R. Woods, and A. H. Marshall, "A bayesian network based learning system for modelling faults in large-scale manufacturing," in *2018 IEEE international conference on industrial technology (ICIT)*. IEEE, 2018, pp. 1357–1362.

[56] F. Ansari, R. Glawar, and W. Sihn, "Prescriptive maintenance of cpps by integrating multimodal data with dynamic bayesian networks," in *Machine Learning for Cyber Physical Systems*. Springer, 2020, pp. 1–8.

[57] D. L. Nuñez and M. Borsato, "Ontoprog: An ontology-based model for implementing prognostics health management in mechanical machines," *Advanced Engineering Informatics*, vol. 38, pp. 746–759, 2018.

[58] S. Selcuk, "Predictive maintenance, its implementation and latest trends," *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 231, no. 9, pp. 1670–1679, 2017.

[59] A. Sarazin, S. Truptil, A. Montarnal, and J. Lamothe, "Toward information system architecture to support predictive maintenance approach," in *Enterprise Interoperability VIII*. Springer, 2019, pp. 297–306.

[60] M. P. Lambán, P. Morella, J. Royo, and J. C. Sánchez, "Using industry 4.0 to face the challenges of predictive maintenance: A key performance indicators development in a cyber physical system," *Computers & Industrial Engineering*, vol. 171, p. 108400, 2022.

[61] N. Kang, C. Zhao, J. Li, and J. A. Horst, "A hierarchical structure of key performance indicators for operation management and continuous improvement in production systems," *International journal of production research*, vol. 54, no. 21, pp. 6333–6350, 2016.

[62] O. Sénéchal and D. Trentesaux, "A framework to help decision makers to be environmentally aware during the maintenance of cyber physical systems," *Environmental Impact Assessment Review*, vol. 77, pp. 11–22, 2019.

[63] Y. Li, W. Zhao, and J. Pan, "Deformable patterned fabric defect detection with fisher criterion-based deep learning," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 2, pp. 1256–1264, 2017.

[64] C.-Y. Lee, J.-Y. Lin, and R.-I. Chang, "Improve quality and efficiency of textile process using data-driven machine learning in industry 4.0," *International Journal of Technology and Engineering Studies*, vol. 4, no. 1, pp. 64–76, 2018.

[65] J. Villalba-Diez, D. Schmidt, R. Gevers, J. Ordieres-Meré, M. Buchwitz, and W. Wellbrock, "Deep learning for industrial computer vision quality control in the printing industry 4.0," *Sensors*, vol. 19, no. 18, p. 3987, Sep 2019. [Online]. Available: http://dx.doi.org/10.3390/s19183987

[66] R. Ozdemir and M. Koc, "A quality control application on a smart factory prototype using deep learning methods," in *2019 IEEE 14th International Conference on Computer Sciences and Information Technologies (CSIT)*, vol. 1, 2019, pp. 46–49.

[67] J. Francis and L. Bian, "Deep learning for distortion prediction in laser-based additive manufacturing using big data," *Manufacturing Letters*, vol. 20, pp. 10 – 14, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S221384631830172X

[68] A. Muniategui, A. G. de la Yedra, J. A. del Barrio, M. Masenlle, X. Angulo, and R. Moreno, "Mass production quality control of welds based on image processing and deep learning in safety components industry," in *Fourteenth International Conference on Quality Control by Artificial Vision*, C. Cudel, S. Bazeille, and N. Verrier, Eds., vol. 11172, International Society for Optics and Photonics. SPIE, 2019, pp. 148 – 155.

[69] R. S. Peres, J. Barata, P. Leitao, and G. Garcia, "Multistage quality control using machine learning in the automotive industry," *IEEE Access*, vol. 7, pp. 79 908–79 916, 2019.

[70] Y. Bai, C. Li, Z. Sun, and H. Chen, "Deep neural network for manufacturing quality prediction," in *2017 Prognostics and System Health Management Conference (PHM-Harbin)*, 2017, pp. 1–5.

[71] P. A. Gutierrez, M. Perez-Ortiz, J. Sanchez-Monedero, F. Fernandez-Navarro, and C. Hervas-Martinez, "Ordinal regression methods: survey and experimental study," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 1, pp. 127–146, 2016.

[72] J. Barbero-Gómez, P.-A. Gutiérrez, V.-M. Vargas, J.-A. Vallejo-Casas, and C. Hervás-Martínez, "An ordinal cnn approach for the assessment of neurolog-

ical damage in parkinson's disease patients," *Expert Systems with Applications*, vol. 182, p. 115271, 2021.

[73] G. Singer, A. Ratnovsky, and S. Naftali, "Classification of severity of trachea stenosis from eeg signals using ordinal decision-tree based algorithms and ensemble-based ordinal and non-ordinal algorithms," *Expert Systems with Applications*, vol. 173, p. 114707, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417421001482

[74] X. Liu, F. Fan, L. Kong, Z. Diao, W. Xie, J. Lu, and J. You, "Unimodal regularized neuron stick-breaking for ordinal classification," *Neurocomputing*, vol. 388, pp. 34–44, 2020.

[75] O. Rudovic, V. Pavlovic, and M. Pantic, "Context-sensitive dynamic ordinal regression for intensity estimation of facial action units," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 5, pp. 944–958, 2015.

[76] K.-Y. Chang and C.-S. Chen, "A learning framework for age rank estimation based on face images with scattering transform," *IEEE Transactions on Image Processing*, vol. 24, no. 3, pp. 785–798, 2015.

[77] R. He, T. Tan, L. Davis, and Z. Sun, "Learning structured ordinal measures for video based face recognition," *Pattern Recognition*, vol. 75, pp. 4–14, 2018, distance Metric Learning for Pattern Recognition. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320317300493

[78] R. Hirk, K. Hornik, and L. Vana, "Multivariate ordinal regression models: an analysis of corporate credit ratings," *Statistical Methods & Applications*, vol. 28, no. 3, pp. 507–539, 2019.

[79] E. Balugani, F. Lolli, M. Pini, A. M. Ferrari, P. Neri, R. Gamberini, and B. Rimini, "Dimensionality reduced robust ordinal regression applied to life cycle assessment," *Expert Systems with Applications*, vol. 178, p. 115021, 2021.

[80] P. A. Gutiérrez, M. Pérez-Ortiz, J. Sánchez-Monedero, F. Fernández-Navarro, and C. Hervás-Martínez, "Ordinal regression methods: Survey and experimental study," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 1, pp. 127–146, 2016.

[81] K. Uematsu and Y. Lee, "Statistical optimality in multipartite ranking and ordinal regression," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 5, pp. 1080–1094, 2015.

[82] V. C. Raykar, R. Duraiswami, and B. Krishnapuram, "A fast algorithm for learning a ranking function from large-scale data sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 7, pp. 1158–1170, 2008.

[83] D. Xu, F. Zhu, Q. Liu, and P. Zhao, "Arail: Learning to rank from incomplete demonstrations," *Information Sciences*, vol. 565, pp. 422–437, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025521001213

[84] G. Singer, R. Anuar, and I. Ben-Gal, "A weighted information-gain measure for ordinal classification trees," *Expert Systems with Applications*, vol. 152, p. 113375, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417420302001

[85] O. Rudovic, V. Pavlovic, and M. Pantic, "Context-sensitive dynamic ordinal regression for intensity estimation of facial action units," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 5, pp. 944–958, 2015.

[86] H. Zhu, H. Shan, Y. Zhang, L. Che, X. Xu, J. Zhang, J. Shi, and F.-Y. Wang, "Convolutional ordinal regression forest for image ordinal estimation," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–12, 2021.

[87] O. C. Hamsici and A. M. Martinez, "Multiple ordinal regression by maximizing the sum of margins," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 10, pp. 2072–2083, 2016.

[88] Z. Ma and S. Chen, "A convex formulation for multiple ordinal output classification," *Pattern Recognition*, vol. 86, pp. 73–84, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320318303248

[89] L. Pfannschmidt, J. Jakob, F. Hinder, M. Biehl, P. Tino, and B. Hammer, "Feature relevance determination for ordinal regression in the context of feature redundancies and privileged information," *Neurocomputing*, vol. 416, pp. 266–279, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231220305038

[90] Q. Tian, W. Zhang, L. Wang, S. Chen, and H. Yin, "Robust ordinal regression induced by lp-centroid," *Neurocomputing*, vol. 313, pp. 184–195, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S092523121830777X

[91] F. Fernández-Navarro, P. A. Gutiérrez, C. Hervás-Martánez, and X. Yao, "Negative correlation ensemble learning for ordinal regression," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 11, pp. 1836–1849, 2013.

[92] F. Fernández-Navarro, A. Riccardi, and S. Carloni, "Ordinal neural networks without iterative tuning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 11, pp. 2075–2085, 2014.

[93] V. M. Vargas, P. A. Gutiérrez, and C. Hervás-Martínez, "Cumulative link models for deep ordinal classification," *Neurocomputing*, vol. 401, pp. 48–58, 2020.

[94] C. Gentile, "The robustness of the p-norm algorithms," *Machine Learning*, vol. 53, no. 3, pp. 265–299, 2003.

[95] Y.-F. Ye, Y.-H. Shao, and C.-N. Li, "Wavelet lp-norm support vector regression with feature selection," *Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol. 19, no. 3, pp. 407–416, 2015.

[96] C. Zhou, J. Zhang, and J. Liu, "Lp-wgan: Using lp-norm normalization to stabilize wasserstein generative adversarial networks," *Knowledge-Based Systems*, vol. 161, pp. 415–424, 2018.

[97] M. Liu and D. F. Gleich, "Strongly local p-norm-cut algorithms for semi-supervised learning and local graph clustering," *arXiv preprint*, 2020.

[98] K. Thurnhofer-Hemsi, E. López-Rubio, N. Roe-Vellve, and M. A. Molina-Cabello, "Multiobjective optimization of deep neural networks with combinations of lp-norm cost functions for 3d medical image super-resolution," *Integrated Computer-Aided Engineering*, no. Preprint, pp. 1–19, 2020.

[99] T. Ke, L. Zhang, X. Ge, H. Lv, and M. Li, "Construct a robust least squares support vector machine based on L_p-norm and L_∞-norm," *Engineering Applications of Artificial Intelligence*, vol. 99, p. 104134, 2021.

[100] J. A. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural processing letters*, vol. 9, no. 3, pp. 293–300, 1999.

[101] Q. Ye, L. Fu, Z. Zhang, H. Zhao, and M. Naiem, "Lp-and ls-norm distance based robust linear discriminant analysis," *Neural Networks*, vol. 105, pp. 393–404, 2018.

[102] J. Kivinen, M. K. Warmuth, and B. Hassibi, "The p-norm generalization of the lms algorithm for adaptive filtering," *IEEE Transactions on Signal Processing*, vol. 54, no. 5, pp. 1782–1793, 2006.

[103] C. Vens, J. Struyf, L. Schietgat, S. Džeroski, and H. Blockeel, "Decision trees for hierarchical multi-label classification," *Machine learning*, vol. 73, no. 2, p. 185, 2008.

[104] L. Masera and E. Blanzieri, "Awx: An integrated approach to hierarchical-multilabel classification," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2018, pp. 322–336.

[105] Y. Li, S. Wang, R. Umarov, B. Xie, M. Fan, L. Li, and X. Gao, "Deepre: sequence-based enzyme ec number prediction by deep learning," *Bioinformatics*, vol. 34, no. 5, pp. 760–769, 2018.

[106] E. Giunchiglia and T. Lukasiewicz, "Coherent hierarchical multi-label classification networks," *NeurIPS Proceedings*, vol. 33, 2020.

[107] T. G. Dietterich and G. Bakiri, "Solving multiclass learning problems via error-correcting output codes," *Journal of artificial intelligence research*, vol. 2, pp. 263–286, 1994.

[108] M. B. Bora, D. Daimary, K. Amitab, and D. Kandar, "Handwritten character recognition from images using cnn-ecoc," *Procedia Computer Science*, vol. 167, pp. 2403–2409, 2020.

[109] J. Barbero-Gómez, P. A. Gutiérrez, and C. Hervás-Martínez, "Error-correcting output codes in the framework of deep ordinal classification," *Neural Processing Letters*, pp. 1–32, 2022.

[110] C. N. Silla and A. A. Freitas, "A survey of hierarchical classification across different application domains," *Data Mining and Knowledge Discovery*, vol. 22, no. 1, pp. 31–72, 2011.

[111] J. Wehrmann, R. Cerri, and R. Barros, "Hierarchical multi-label classification networks," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5075–5084.

[112] E. Florian, F. Sgarbossa, and I. Zennaro, "Machine learning-based predictive maintenance: A cost-oriented model for implementation," *International Journal of Production Economics*, vol. 236, p. 108114, 2021.

[113] S. Schwendemann, Z. Amjad, and A. Sikora, "A survey of machine-learning techniques for condition monitoring and predictive maintenance of bearings in grinding machines," *Computers in Industry*, vol. 125, p. 103380, 2021.

[114] S. Ayvaz and K. Alpay, "Predictive maintenance system for production lines in manufacturing: A machine learning approach using iot data in real-time," *Expert Systems with Applications*, vol. 173, p. 114598, 2021.

[115] V. Nguyen, J. Seshadrinath, D. Wang, S. Nadarajan, and V. Vaiyapuri, "Model-based diagnosis and rul estimation of induction machines under interturn fault," *IEEE Transactions on Industry Applications*, vol. 53, no. 3, pp. 2690–2701, 2017.

[116] T. P. Carvalho, F. A. Soares, R. Vita, R. d. P. Francisco, J. P. Basto, and S. G. Alcalá, "A systematic literature review of machine learning methods applied

to predictive maintenance," *Computers & Industrial Engineering*, vol. 137, p. 106024, 2019.

[117] S. Kotsiantis, D. Kanellopoulos, P. Pintelas *et al.*, "Handling imbalanced datasets: A review," *GESTS international transactions on computer science and engineering*, vol. 30, no. 1, pp. 25–36, 2006.

[118] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[119] T. Hasanin and T. Khoshgoftaar, "The effects of random undersampling with simulated class imbalance for big data," in *2018 IEEE International Conference on Information Reuse and Integration (IRI)*. IEEE, 2018, pp. 70–79.

[120] S. Mishra, "Handling imbalanced data: Smote vs. random undersampling," *Int. Res. J. Eng. Technol*, vol. 4, no. 8, pp. 317–320, 2017.

[121] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[122] J. Phillips, E. Cripps, J. W. Lau, and M. Hodkiewicz, "Classifying machinery condition using oil samples and binary logistic regression," *Mechanical Systems and Signal Processing*, vol. 60, pp. 316–325, 2015.

[123] G. A. Susto, A. Schirru, S. Pampuri, S. McLoone, and A. Beghi, "Machine learning for predictive maintenance: A multiple classifier approach," *IEEE transactions on industrial informatics*, vol. 11, no. 3, pp. 812–820, 2014.

[124] P. Bilski, "Application of support vector machines to the induction motor parameters identification," *Measurement*, vol. 51, pp. 377 – 386, 2014.

[125] J. D. Kelleher, B. Mac Namee, and A. D'arcy, *Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies*. MIT press, 2020.

[126] K. B. Andrew Banks, Ed Briggs and R. Gupta. (2019) Mqtt version 5.0 oasis standard. [Online]. Available: https://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.html

[127] N. Moustafa, J. Slay, and G. Creech, "Novel geometric area analysis technique for anomaly detection using trapezoidal area estimation on large-scale networks," *IEEE Transactions on Big Data*, vol. 5, no. 4, pp. 481–494, 2019.

[128] R. Bulirsch and J. Stoer, *Introduction to numerical analysis*. Springer, 2002, vol. 3.

[129] C. Molnar, G. Casalicchio, and B. Bischl, "Quantifying interpretability of arbitrary machine learning models through functional decomposition," *Ulmer Informatik-Berichte*, p. 41, 2019.

[130] A.-L. Boulesteix, S. Janitza, J. Kruppa, and I. R. König, "Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics," *Data Mining and Knowledge Discovery*, vol. 2, no. 6, pp. 493–507, 2012.

[131] S. Janitza, E. Celik, and A.-L. Boulesteix, "A computationally fast variable importance test for random forests for high-dimensional data," *Advances in Data Analysis and Classification*, vol. 12, no. 4, pp. 885–915, 2018.

[132] G. A. Susto, A. Schirru, S. Pampuri, and A. Beghi, "A predictive maintenance system based on regularization methods for ion-implantation," in *2012 SEMI Advanced Semiconductor Manufacturing Conference*. IEEE, 2012, pp. 175–180.

[133] G. C. Cawley and N. L. Talbot, "On over-fitting in model selection and subsequent selection bias in performance evaluation," *Journal of Machine Learning Research*, vol. 11, no. 7, pp. 2079–2107, 2010.

[134] A. Di Bucchianico, "Coefficient of determination (R^2)," *Encyclopedia of Statistics in Quality and Reliability*, vol. 1, 2008.

[135] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[136] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[137] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[138] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[139] H. Cramér, *Mathematical methods of statistics*. Princeton university press, 1999, vol. 43.

[140] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.

[141] A. Agresti, *Analysis of ordinal categorical data*. J. Wiley & Sons, 2010, vol. 656.

[142] R. Herbrich, T. Graepel, and K. Obermayer, "Large margin rank boundaries for ordinal regression," *Advances in large margin classifiers*, vol. 88, no. 2, pp. 115–132, 2000.

[143] V. M. Vargas, P. A. Gutiérrez, and C. Hervás, "Deep ordinal classification based on the proportional odds model," in *Proceedings of the International Work-Conference on the Interplay Between Natural and Artificial Computation*. Springer, 2019, pp. 441–451.

[144] J. de la Torre, D. Puig, and A. Valls, "Weighted kappa loss function for multi-class classification of ordinal data in deep learning," *Pattern Recogn. Lett.*, vol. 105, no. C, p. 144–154, Apr. 2018.

[145] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[146] M. Cruz-Ramírez, C. Hervás-Martínez, J. Sánchez-Monedero, and P. A. Gutiérrez, "Metrics to guide a multi-objective evolutionary algorithm for ordinal classification," *Neurocomputing*, vol. 135, pp. 21–31, 2014.

[147] C.-B. Zhang, P.-T. Jiang, Q. Hou, Y. Wei, Q. Han, Z. Li, and M.-M. Cheng, "Delving deep into label smoothing," *IEEE Transactions on Image Processing*, vol. 30, pp. 5984–5996, 2021.

[148] X. Liu, F. Fan, L. Kong, Z. Diao, W. Xie, J. Lu, and J. You, "Unimodal regularized neuron stick-breaking for ordinal classification," *Neurocomputing*, vol. 388, no. 7, pp. 34–44, 2020.

[149] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations*, 2015, pp. 1–15.

[150] F. J. Massey Jr, "The kolmogorov-smirnov test for goodness of fit," *Journal of the American statistical Association*, vol. 46, no. 253, pp. 68–78, 1951.

[151] G. A. Mack and J. H. Skillings, "A friedman-type rank test for main effects in a two-factor anova," *Journal of the American Statistical Association*, vol. 75, no. 372, pp. 947–951, 1980.

[152] J. Sánchez-Monedero, M. Pérez-Ortiz, A. Saez, P. A. Gutiérrez, and C. Hervás-Martínez, "Partial order label decomposition approaches for melanoma diagnosis," *Applied Soft Computing*, vol. 64, pp. 341–355, 2018.

[153] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[154] V. M. Vargas, P. A. Gutiérrez, and C. Hervás-Martínez, "Unimodal regularisation based on beta distribution for deep ordinal regression," *Pattern Recognition*, vol. 122, pp. 1–10, 2022.

[155] R. G. Miller Jr, *Beyond ANOVA: basics of applied statistics*. Chapman and Hall/CRC, 1997.

[156] J. W. Tukey, "Comparing individual means in the analysis of variance," *Biometrics*, vol. 5, no. 2, pp. 99–114, 1949.

[157] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37, 2015, pp. 448–456.

[158] M. Kulmanov, M. A. Khan, and R. Hoehndorf, "Deepgo: predicting protein functions from sequence and interactions using a deep ontology-aware classifier," *Bioinformatics*, vol. 34, no. 4, pp. 660–668, 2018.

[159] C. Xu and X. Geng, "Hierarchical classification based on label distribution learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 5533–5540.

[160] R. M. Pereira, Y. M. Costa, and C. N. Silla, "Handling imbalance in hierarchical classification problems using local classifiers approaches," *Data Mining and Knowledge Discovery*, pp. 1–58, 2021.

[161] V. Sanh, T. Wolf, and S. Ruder, "A hierarchical multi-task approach for learning embeddings from semantic tasks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 6949–6956.

[162] F. W. Scholz and M. A. Stephens, "K-sample anderson–darling tests," *Journal of the American Statistical Association*, vol. 82, no. 399, pp. 918–924, 1987.

[163] Z. Chai and C. Zhao, "Multiclass oblique random forests with dual-incremental learning capacity," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 12, pp. 5192–5203, 2020.

[164] L. Abualigah, M. Abd Elaziz, P. Sumari, Z. W. Geem, and A. H. Gandomi, "Reptile search algorithm (rsa): A nature-inspired meta-heuristic optimizer," *Expert Systems with Applications*, vol. 191, p. 116158, 2022.

[165] L. Abualigah, A. Diabat, P. Sumari, and A. H. Gandomi, "Applications, deployments, and integration of internet of drones (iod): a review," *IEEE Sensors Journal*, 2021.

[166] P. Zhang, H.-N. Wu, R.-P. Chen, and T. H. Chan, "Hybrid meta-heuristic and machine learning algorithms for tunneling-induced settlement prediction: A comparative study," *Tunnelling and Underground Space Technology*, vol. 99, p. 103383, 2020.

[167] M. Compare, P. Baraldi, and E. Zio, "Challenges to iot-enabled predictive maintenance for industry 4.0," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4585–4597, 2019.

[168] L. Kook, L. Herzog, T. Hothorn, O. Dürr, and B. Sick, "Deep and interpretable regression models for ordinal outcomes," *Pattern Recognition*, vol. 122, p. 108263, 2022.

[169] H. Yao, Y. Wei, J. Huang, and Z. Li, "Hierarchically structured meta-learning," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 7045–7054.

[170] X. Zhai, A. Oliver, A. Kolesnikov, and L. Beyer, "S4l: Self-supervised semi-supervised learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1476–1485.

[171] H. Chen, Y. Jia, J. Ge, and B. Gu, "Incremental learning algorithm for large-scale semi-supervised ordinal regression," *Neural Networks*, vol. 149, pp. 124–136, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608022000375

[172] B. H. Zhang, B. Lemoine, and M. Mitchell, "Mitigating unwanted biases with adversarial learning," in *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, 2018, pp. 335–340.