Universidade do Minho
Escola de Engenharia

Rui Cândido Azevedo Ribeiro

An Intelligent Decision Support System for the Textile Industry

March, 2023

An Intelligent Decison Support System for the Textile Industry

Rui Cândido Azevedo Ribeiro

UMinho | 2023

Universidade do Minho
Escola de Engenharia
Departamento de Sistemas de Informação

Rui Cândido Azevedo Ribeiro

An Intelligent Decision Support System for the Textile Industry

PhD Thesis
Doctoral Program on Information Systems and Technology

Supervisor:
Professor Doctor Paulo Alexandre Ribeiro Cortez

March, 2023

DIREITOS DE AUTOR E CONDIÇÕES DE UTILIZAÇÃO DO TRABALHO POR TERCEIROS

ACKNOWLEDGEMENTS

This PhD has been a long and arduous journey, but fortunately, I have not made it alone. I have been surrounded by amazing people along the way, who have taught and encouraged me immensely, and to whom I would like to dedicate this work.

To my supervisor, Professor Paulo Cortez, who gave me the opportunity to work with him on this project, being a key support throughout this journey. I would like to highlight the unconditional support he provided, the interested, extraordinary, and pertinent way he accompanied the realization of this work. His constructive criticism, discussions and reflections were fundamental throughout this journey. I cannot forget his great contribution to my growth as a researcher. Eternally grateful for all the support.

To my girlfriend Fabi, who has never wavered in her support and understanding of what this personal endeavor means to me.

To my mother and father, who have always encouraged me to realize my aspirations.

To my employers, CCG\ZGDV Institute, for providing me with the conditions to perform this work and achieve this goal and for supporting my growth as a research scientist.

To everyone within EPMQ: Computer Engineering, Process, Data, Maturity, and Quality Lab. To my research colleagues, old and new, for the knowledge shared, the support and motivation, and the insightful discussions.

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

ABSTRACT

In the Textile industry there is a need to design new fabrics more rapidly to meet the demands of consumers. With the fourth industrial revolution bringing the digitalization and automation of manufacturing processes to a new level, the volume of data being stored has increased, enabling the use of Artificial Intelligence tools to enhance textile production processes.

This PhD work was executed as part of a *Research and Development (R&D)* project. The main objective is the development of an *Intelligent Decision Support System (IDSS)*, based on the Adaptive Business Intelligence concept, that will assist in the design of new textile fabrics through predictive and prescriptive analytics. To address this goal several experiments were performed. The first set of experiments was related to the prediction of two laboratory quality tests using an *Automated Machine Learning (AutoML)* tool. In the second predictive experiment, nine quality tests were selected, using an AutoML tool to compare the value of proposed input fabric yarn and finishing feature representations. Then, an initial prescriptive study was performed within the context of a different R&D project, aiming to develop an IDSS for production planning in the Textile industry. Using an AutoML tool for the predictive module, and NSGA-II to minimize both the cost and production time, the IDSS searched for the best subcontractor allocation plan, achieving interesting results.

Based on the previously gained experience, the PhD main IDSS was developed, aiming to support the creation of new textile fabrics. For the predictive module, we compared a Single-Target Regression approach, obtained during the second predictive work, and a Multi-Target Regression, via a deep learning approach, for the prediction of four fabric physical properties and the final textile composition. For the prescriptive module, we compared two methods (NSGA-II and R-NSGA-II), aiming to simultaneously minimize the physical property predictive error and the distance of the optimized values to the learned input space. The full IDSS was evaluated using 100 new fabrics and then presented to the textile company experts, which provided positive feedback.

Keywords: Machine Learning, Modern Optimization, Predictive Analytics, Prescriptive Analytics, Evolutionary Computation, Textile Industry

RESUMO

Na indústria Têxtil há necessidade de desenvolver rapidamente novos tecidos para responder ás exigências dos consumidores. Com a quarta revolução industrial a contribuir para a digitalização e automatização dos processos de fabrico, o volume de dados armazenados aumentou, permitindo a utilização de técnicas de Inteligência Artificial para melhorar processos de produção têxtil.

Este trabalho de doutoramento foi executado no âmbito de um projeto de Investigação e Desenvolvimento (I&D). O principal objetivo é a investigação e implementação de um Sistema Inteligente de Apoio à Decisão (SIAD) baseado no conceito de *Adaptive Business Intelligence*, que irá auxiliar a conceção de novos tecidos, através de análises preditivas e prescritivas. Para responder a este desafio, foram efetuadas várias experiências. As primeiras experiências estão relacionadas com a previsão de dois testes de qualidade utilizando uma ferramenta de *Automated Machine Learning* (AutoML). Nas segundas experiências, também preditivas, foram selecionados nove testes de qualidade, e recorrendo a uma ferramenta AutoML foi comparado o valor de diferentes representações de fios e acabamentos.

De seguida, foi efetuado um estudo prescritivo inicial, no contexto de um projeto de I&D diferente, com o objetivo de desenvolver um SIAD para o planeamento da produção na indústria têxtil. Utilizando uma ferramenta AutoML e NSGA-II para minimizar o custo e o tempo de produção simultaneamente, procurou-se o melhor plano de alocação de subcontratados, obtendo resultados interessantes.

Com base nos estudos anteriores, foi desenvolvido o SIAD deste doutoramento, para o suporte à conceção de novos tecidos. No módulo preditivo, comparámos uma abordagem de regressão de objetivo único, obtida durante a segunda experiência, e uma regressão com vários objetivos, através de uma abordagem de *deep learning*, para a previsão de quatro propriedades físicas de tecidos e a composição têxtil final. Para o módulo prescritivo, comparámos dois métodos (NSGA-II e R-NSGA-II), com o objetivo de minimizar simultaneamente o erro preditivo das propriedades físicas e a distância dos valores otimizados ao espaço de entrada aprendido. O SIAD foi avaliado utilizando 100 novos tecidos e apresentado à empresa têxtil que forneceu um parecer positivo.

Keywords: *Machine Learning, Modern Optimization, Predictive Analytics, Prescriptive Analytics, Evolutionary Computation, Textile Industry*

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

ABC Artificial Bee Colony.

ABI Adaptive Business Intelligence.

ACO Ant Colony Optimization.

AI Artificial Intelligence.

ANN Artificial Neural Network.

AutoML Automated Machine Learning.

CPS Cyber-Physical System.

CRISP-DM Cross-Industry Standard Process for Data Mining.

DM Data Mining.

DSS Decision Support System.

DT Decision Trees.

EA Evolutionary Algorithms.

EC Evolutionary Computation.

EMO Evolutionary Multi-objective Optimization.

ERP Enterprise Resource Planning.

ETL Extraction Transform Load.

GA Genetic Algorithm.

GBM Gradient Boosting Machines.

GLM Generalized Linear Models.

HV Hypervolume.

IDF Inverse Document Frequency.

IDSS Intelligent Decision Support System.

IoT Internet of Things.

IS Information System.

KDD Knowledge Discovery in Data.

KNN K-Nearest Neighbor.

MAE Mean Absolute Error.

ML Machine Learning.

MO Modern Optimization.

NMAE Normalized Mean Absolute Error.

NSGA-II Non-dominated Sorting Genetic Algorithm II.

PSO Particle Swarm Optimization.

R-NSGA-II Reference Point Based Non-dominated Sorting Genetic Algorithm II.

R&D Research and Development.

REC Regression Error Characteristic.

RF Random Forest.

SE Stacked Ensemble.

SVM Support Vector Machines.

XG XGBoost.

XRT Extremely Randomized Trees.

1

INTRODUCTION

This chapter presents the motivation and objectives of this PhD work. Next, the adopted research methodology is presented, followed by the main contributions of this work, and, finally a description of the structure of this document is presented.

1.1. Motivation

Information Technologies are constantly evolving, creating significant changes in organizations and companies. One of the areas that is undergoing a significant transformation is the manufacturing industry, in what is known as the Industry 4.0 phenomenon. This PhD work is set within this context. It was set within a three-year *Research and Development (R&D)* project that was part of a Portugal 2020 structuring program of the Cluster of Textile Technology and Fashion, which addressed the digitization and dematerialization of textiles.

The mentioned R&D project involves a Textile company that is one of the oldest manufacturers of fabrics in Portugal, producing custom-made fabrics for several different client companies. The current fabric design process is mostly based on the designer experience and intuition often requiring several trial-and-error production experiments. In order to produce a fabric that meets the requirements of the client or to develop a new product, the company takes several steps leading to the production of a fabric prototype and executing several tests to verify if it complies with the client requirements and all quality standards. If this prototype is not approved, the design process must be repeated, which is translated into more time and costs for the company. The fabric design process generates data that is related to several components of the fabric, as well as the quality test results. Hence, this data can be used to develop a less costly and more efficient method that could have a direct influence on both the organization and the customers. In this PhD thesis, which covers the intelligent data analysis aspect of the R&D project, we address this particular issue, by using fabric design textile company data, to develop an *Intelligent Decision Support System (IDSS)*, that assumes an *Adaptive Business Intelligence (ABI)* approach (Michalewicz et al., 2006).

Most state-of-the-art works applied to this specific fabric design process address a single type of intelligent data analysis, either predictive analytics (Yap et al., 2010; Mozafary and Payvandy, 2014; Ahmad, 2016; Eltayib et al., 2016) or prescriptive (Majumdar et al., 2017; Pfrommer et al., 2018), with very few using both analytics (Mitra et al., 2015; Majumdar et al., 2016). Moreover, the majority of the related works are focused on one type of challenge of the fabric design process and the selected data attributes only consider one specific phase of textile development information (e.g., spinning, dyeing), with a small number of variables. It is within this context the motivation for this work derives, in which it is intended to design, implement and validate an IDSS that will use predictive and prescriptive analytics, through *Machine Learning (ML)* and *Modern Optimization (MO)* techniques, in order to reduce the number of trial-and-error fabric production attempts. The goal is to adopt a multi-objective approach, in which the predictive models are used as an oracle, to get a quick and cheap estimate of the true fabric properties, helping in the definition of the correct input fabric design values for a particular set of client target properties (e.g., abrasion, seam slippage, elasticity, and others).

1.2. Objectives

This PhD aims to research the design of an IDSS that can assist in the creation of new textile fabrics. This PhD was partially developed under a funded structuring project, which aimed to embrace a set of a R&D initiatives with a central involvement of Textiles and Clothing companies, named *TexBoost - less Commodities more Specialities*[1]. The overall goal of the R&D project is to present an innovative solution related to the digitization and dematerialization of fabrics that is based on the use of Information Technology. The project consortium involves a real Textile company and *Centro de Computação Gráfica* (CCG). The R&D project component handled by CCG has two main components: i) extraction of data from multiple data sources and its integration into a unified data warehouse, oriented for business analysis; and ii) research and development of an IDSS, to assist in the design of new textile fabrics.

[1] https://www.texboost.pt/en/project

This PhD is focused on the second component, although it will make use of the unified data warehouse, as it will be the IDSS data source.

Under the current typical fabric design approach, assumed by most of the companies in the textile sector, there is an execution of several trial-and-error fabric production attempts to meet several targeted properties, either for functional or structural properties. The development of fabric prototypes involves specialized equipment and personnel, manufacturing lines, and several cycles of adjustments (e.g., involving laboratory quality tests, validation sessions with customers, and production of replicas of the prototype in a larger quantity) before reaching the final prototype (order to be mass produced). Thus, each failed attempt implies additional costs and time to the whole process of new fabric creation.

Therefore, the research question addressed in this PhD can be formulated as: **How to design and implement an IDSS that includes predictive and prescriptive analytics in order to support the design of new textile fabrics?** The proposed IDSS makes use of *Artificial Intelligence (AI)* techniques, namely ML and MO. Aiming to answer this research question, in Chapter 3 we perform several initial predictive and prescriptive exploratory experiments. Then, building upon the acquired knowledge, in Chapter 4 we propose the final IDSS that includes: two predictive goals – the estimation of the final fabric properties (as measured by laboratory tests) and the identification of the final textile composition (e.g., percentage of cotton); and a prescriptive module – that uses the best predictive models and MO methods in order to search for the ideal set of fabric design inputs.

1.3. Research Methodology

Considering that this project main focus is the design and development of an IDSS to solve a specific problem, the adopted methodology is Design Research, specifically Design Science Research Methodology for Information Systems. As argued by Arnott and Pervan (2014), Design Science is a natural methodology for the development of IDSS, with its usage being significantly increased in the last few years. The methodology consists of a set of principles, practices, and procedures required to develop *Information System (IS)* research. This process, presented in Figure 1, includes six steps: problem identification and motivation, the definition of the objectives for a solution, design and development, demonstration, evaluation, and communication (Peffers et al., 2007). This is an iterative process, meaning that until the artifact is considered finished, it is possible to restart from a previous step. We briefly describe how each of these steps were applied in this PhD.

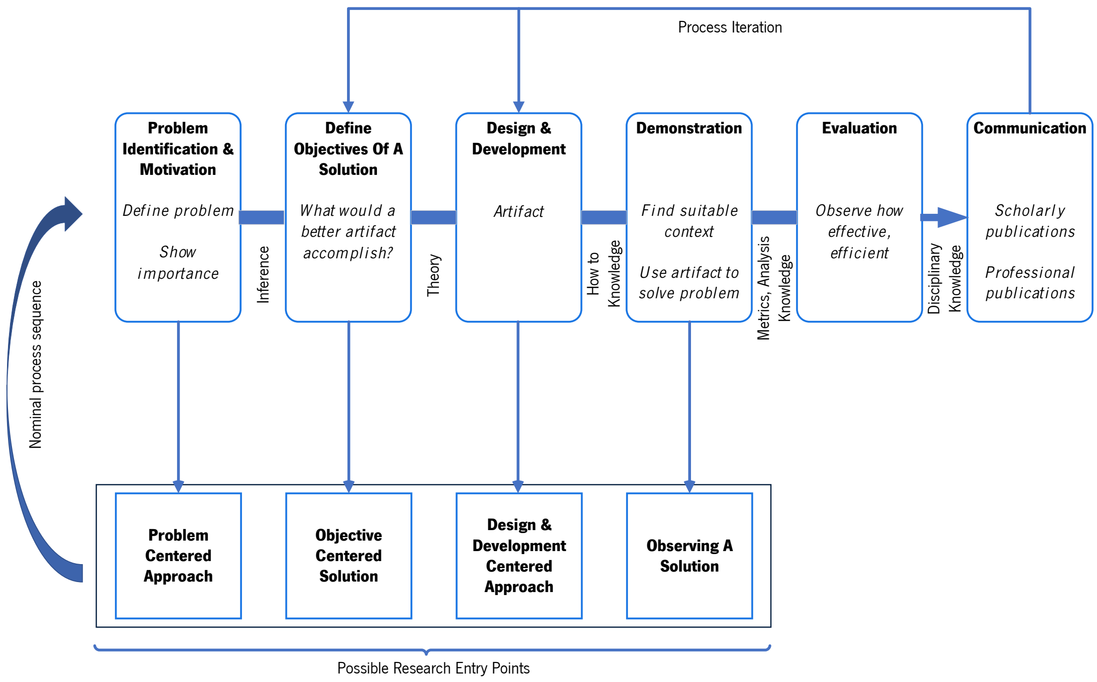

Figure 1: Design science research methodology process model (adapted from Peffers et al. (2007))

The process of creating new fabrics is comprised of several steps, ending with the production of a small prototype of the fabric, in order to verify if it passes all mandatory tests and if it meets the requirements defined by the clients. This prototype can be approved or rejected and if the latter happens, the whole process or parts of it must be repeated, implying costs to the organization. This process is very dependent on human knowledge and experience, meaning that if the individual leaves the organization, the knowledge is lost. Thus, there is a need to implement a system that reduces the number of prototype productions by making use of the respective data generated in the organization and simultaneously allowing the textile company to retain that knowledge.

In the second step of this process, the objectives of the solution must be inferred from the problem definition and knowledge of what is possible and feasible in order to solve the mentioned problem. This solution should then be implemented and evaluated in the next steps, aiming to increase the existing body of knowledge. Regarding this doctoral project, the suggested solution consists on developing a IDSS that will use predictive methods employing ML techniques to estimate several of the final characteristics of a fabric, and prescriptive methods using MO algorithms such as *Evolutionary Computation (EC)* (Cortez, 2021), attempting to optimize a set of fabric construction parameters that will serve as inputs for the predictive models, such that the fabric requirements are met. Ideally, success should be measured by the capability of the developed IDSS to reduce the number of new fabric design attempts, thus reducing time and costs. However, this would require first a deployment of the proposed IDSS in a real environment and then the monitoring of the IDSS performance. Due to temporal restrictions, such validation is out of the scope of this PhD work and thus an alternative IDSS validation is adopted, in which we perform several realistic computational experiments, measuring the performance of the IDSS individual components (predictive and prescriptive).

After concluding the previous steps, the next step is related to the development of the artifact: the IDSS. The development was done gradually and iteratively, by applying different ML approaches and MO techniques, and a combination of both. The different approaches were compared, using several statistical measures, and the best approaches were selected, considering several factors, such as processing time and the selected evaluation measures. For developing this system, the *R* and *Python* programming languages were used, due to their flexibility and success in different *Data Mining (DM)* tasks and allowing to easily implement both ML and MO algorithms.

The following step requires the demonstration of the use of the artifact solving at least one of the instances of the problem. Since this project had several tasks, each task used real-world data provided by the textile company involved in the R&D project and real-world data was also used to evaluate the IDSS.

Throughout the whole development phase, the several components of the IDSS were evaluated under different measures. For the predictive task, the results were assessed using the *Mean Absolute Error (MAE)*, *Normalized Mean Absolute Error (NMAE)* measures, correlation coefficient also known as R^2 and classification accuracy for a given Tolerance T. The MAE measure was selected since is the most natural measure of average error magnitude (Willmott and Matsuura, 2005). The NMAE measure presents a standardized MAE result. The use of NMAE is due to its ease of interpretation as an independent scale and as a percentage of the value scale. The R^2 was adopted as a predictive quality measure since allows to measure how stronger is the correlation between the two variables. The Tolerance measure presents an analysis based on the *Regression Error Characteristic (REC)* curves and it measures the percentage of correctly classified examples when assuming a fixed absolute error Tolerance (T) (Bi and Bennett, 2003). The optimization module was evaluated under different measures such as computational effort (measured in terms of time and generations executed), *Hypervolume (HV)* measure and analysis of the Pareto points that are closer to the ideal solution. The HV measure was selected for its ease of interpretation as it returns a single value that represents the volume of a multi-objective space, allowing one to easily compare and evaluate different approaches (While et al., 2006).

The last step consists in communicating the problem and its importance, the artifact, its utility and novelty, the rigor of its design, and its effectiveness to researchers. This last step is only achieved when the evaluated IDSS provides results considered satisfactory and provides a solution to the problem defined in the first step. In this doctoral project, the results are considered satisfactory if the predictive and prescriptive components of the proposed IDSS provide an interesting statistically proven performance. This PhD project presented the results obtained during the research in three conference papers and submitted an article to an international journal. These articles are detailed in Chapters 3 and 4.

1.4. Contributions

This PhD work was developed within a R&D project and, consequently, had to meet the requirements of the project. The main goal of this thesis is to give a contribution to the body of knowledge in the area of IDSS and ABI with the design and implementation of an IDSS within the challenging domain of the Textile industry. Under this context, a collection of research papers were written during the execution of this PhD project with the goal of reaching the research objectives outlined in the previous section.

Section 3.2 presents the first predictive experiments regarding textile laboratory quality tests. We selected two tests from the total number of 15 that the textile company adopted, and using real-world data collected from the textile company, we performed three iterations of the *Cross-Industry Standard Process for Data Mining (CRISP-DM)* methodology, with different data preprocessing procedures in each iteration. During the modeling phase of the CRISP-DM, we use an *Automated Machine Learning (AutoML)* tool to better focus on feature engineering and data engineering tasks. We concluded that our proposed data preprocessing method was capable of predicting the selected laboratory quality tests with interesting results. This work was published at the following conference:

- Rui Ribeiro, André Pilastri, Carla Moura, Filipe Rodrigues, Rita Rocha & Paulo Cortez (2020). **Predicting the Tear Strength of Woven Fabrics Via Automated Machine Learning: An Application of the CRISP-DM Methodology.** In J. Filipe, M. Smialek, A. Brodsky, & S. Hammoudi (Eds.), Proceedings of the 22nd International Conference on Enterprise Information Systems, ICEIS 2020, Prague, Czech Republic, May 5-7, 2020, Volume 1 (pp. 548–555). SCITEPRESS. https://doi.org/10.5220/0009411205480555

Section 3.3 focus also on the prediction of laboratory quality tests, and in this work, we proposed novel a ML approach to support the creation of new textile fabrics. Using data from the textile company, we performed two CRISP-DM iterations, comparing three input feature representation strategies related with fabric design and finishing processes. In the modeling stage of CRISP-DM, we adopted an AutoML tool. We selected nine textile physical properties to model and concluded that our proposed strategies to represent yarns combined with the addition of finishing features improved the quality of the predictions. This study provided the final set of features and preprocessing methods that were adopted in Chapter 4 (allowing to better set the proposed IDSS) and resulted in the following conference paper:

- Rui Ribeiro, André Pilastri, Carla Moura, Filipe Rodrigues, Rita Rocha, José Morgado & Paulo Cortez (2020). **Predicting Physical Properties of Woven Fabrics via Automated Machine Learning and Textile Design and Finishing Features.** In I. Maglogiannis, L. Iliadis, & E. Pimenidis (Eds.), Artificial Intelligence Applications and Innovations - 16th IFIP WG 12.5 International Conference, AIAI 2020, Neos Marmaras, Greece, June 5-7, 2020, Proceedings, Part II (Vol. 584, pp. 244–255). Springer. https://doi.org/10.1007/978-3-030-49186-4_21

Section 3.4 presents the first experiment regarding the development of a full IDSS (with prescriptive capabilities) for the Textile industry. Using an AutoML approach, we first addressed four predictive tasks that are crucial for estimating production planning indicators. Then, using the historical data provided by a service provider for the Textile industry, we designed a MO model that uses one of the predicted variables (production time) to automatically allocate subcontractors to execute sequential operations associated with a textile production order. The MO model, based on the *Non-dominated Sorting Genetic Algorithm II (NSGA-II)* algorithm, assumed a Pareto approach and it was designed to simultaneously minimize the cost and time to execute a production order. The results obtained by this approach were considered realistic by the company, which signaled that the obtained Pareto front provided a richer set of trade-off solutions, while also being faster to compute when compared with the currently adopted manual subcontractor allocation. This study, executed within a different R&D project, provided practical knowledge about the usage of *Evolutionary Multi-objective Optimization (EMO)* algorithms and contributed to advance the state-of-the-art regarding the implementation of IDSS approaches in the Textile industry, and resulted in the following conference paper:

- Rui Ribeiro, André Pilastri, Hugo Carvalho, Arthur Matta, Pedro José Pereira, Pedro Rocha, Marcelo Alves & Paulo Cortez (2021, November). **An Intelligent Decision Support System for Production Planning in Garments Industry.** In H. Yin, D. Camacho, P. Tiño, R. Allmendinger, A. J. Tallón-Ballesteros, K. Tang, S.-B. Cho, P. Novais, & S. Nascimento (Eds.), Intelligent Data Engineering and Automated Learning - IDEAL 2021 - 22nd International Conference, IDEAL 2021, Manchester, UK, November 25-27, 2021, Proceedings (Vol. 13113, pp. 378–386). Springer. https://doi.org/10.1007/978-3-030-91608-4_37

Following these initial studies, several preliminary experiments were conducted to study the feasibility of a many-objective optimization approach. In the previous prescriptive study, we used NSGA-II with two objectives to minimize, thus we decided to experiment a new approach using Non-dominated Sorting Genetic Algorithm III for six laboratory quality tests, with a total of seven objectives to be minimized simultaneously. These include the six individual predictive errors, for each of the measured physical properties, and the optimized input distance when compared with the learned input space (as detailed in the next paragraph). Also, since some of the features existent in our data are categorical, we studied several distance measures for mixed data (with both numeric and categorical attributes). Although these studies are not reflected in this document, several attempts were performed using different distance measures aiming to select the one with the best results. However, after several weeks of work, the obtained results were not considered satisfactory, both for the explored distance measures and the many-objective optimization approach. Thus, we decided to adopt the *Inverse Document Frequency (IDF)* transformation to the categorical variables and focus on a simpler multi-objective optimization with two goals. The IDF transformation converts one categorical attribute into a single numeric variable. Thus, it produces a final pure numeric dataset that is more easy to work with in terms of computing Euclidean distances.

Chapter 4 presents the proposed IDSS to enhance the design of new fabrics of a textile company. We compared two distinct ML approaches: Single-Target Regression (STR) and Multi-Target Regression for the predictive tasks, and compared two EMO algorithms for the prescriptive task: NSGA-II and *Reference Point Based Non-dominated Sorting Genetic Algorithm II (R-NSGA-II)*. Using historical data provided by the textile company, we designed a MO model that uses four of the predicted variables to automatically define several design features values of a fabric. To guarantee that the MO obtains viable fabrics, we assume a Pareto front optimization of two objectives. The first objective is to minimize the distance of the obtained fabric features to all the fabrics already produced, to verify if the new fabric is within the known input space; and the second objective minimizes the average absolute error of the predicted property values with the desired ones. Both approaches were compared and evaluated using the selected statistical measures and processing time. To complement the obtained results, we also proposed a novel method to predict the final composition of the fabrics suggested by the IDSS, providing the textile designer more valuable information when analyzing the optimized Pareto solutions. The predictive and prescriptive results were shown to the textile company experts, which provided positive feedback and resulted in the submission of an article to an international journal:

- Rui Ribeiro, André Pilastri, Carla Moura, José Morgado & Paulo Cortez. **A data-driven intelligent decision support system that combines predictive and prescriptive analytics for the design of new textile fabrics**. In Neural Computing and Applications, Springer, In Press, ISSN 0941-0643

During the last year of this PhD, we also collaborated with a masters student, in which the knowledge obtained in the course of this doctorate served as support for a masters thesis, which resulted in the following conference paper:

- João Azevedo, Rui Ribeiro, Luís Miguel Matos, Rui Sousa, João Paulo Silva, André Pilastri & Paulo Cortez (2022). **Predicting Yarn Breaks in Textile Fabrics: A Machine Learning Approach**. Procedia Computer Science, 207, 2301–2310. https://doi.org/10.1016/j.procs.2022.09.289

1.5. Thesis Organization

This dissertation is divided into five chapters. This introduction (Chapter 1) presents the motivation and contextualization of this PhD research challenge followed by the research strategy. It also describes the resulting contributions and the document structure. In Chapter 2, the literature review is presented. It starts by presenting the bibliographic search strategy. Then, several relevant PhD concepts are introduced: Intelligent Decision Support Systems, Predictive Analytics, Prescriptive Analytics, Evaluation Measures, Industry 4.0, Textile industry and Predictive and Prescriptive Analytics in the Textile industry. The chapter ends with a description and summary of the core studies that addressed predictive or prescriptive analytics in the Textile industry.

Chapter 3 is related to our initial predictive and prescriptive attempts, presenting the methods, experiments and results produced within this PhD work. Firstly, it describes an initial set of experiments related with an exploratory study of two laboratory quality tests prediction. Then, a second set of experiments is presented. These experiments are also related to laboratory quality tests prediction focusing on feature selection and engineering tasks. Finally, Section 3.4 describes the experiment related to our first implementation of a prescriptive IDSS for the Textile industry, specifically supporting the production planning of garment production process.

In Chapter 4, the core research contributions of this doctoral project are presented, detailing the methods, experiments and results regarding the design and implementation of the proposed IDSS, which resulted in a submission to an international journal. Finally, Chapter 5 presents the main conclusions of this doctoral project and its limitations, concluding the chapter with possible directions for future work.

2

BACKGROUND

2.1. Introduction

This chapter presents the relevant state-of-the-art works related to the main subjects of this PhD project, which is the use of *Modern Optimization (MO)* techniques in the Textile industry. The chapter starts with a description of the adopted bibliographic search strategy, followed by an introduction of the main concepts approached in this PhD thesis. Finally, the chapter details and summarizes the core studies involving the application of predictive and prescriptive models within the Textile industry, with a particular focus on the specific task of fabric engineering.

2.2. Bibliographic Search Strategy

In order to situate this research and identify gaps within the literature, a literature review was executed. There are several types of literature reviews (e.g., traditional, systematic, automatic) (Jesson et al., 2011). For this review the traditional method was selected, using several criteria. The bibliographic searches were conducted in different scientific databases, namely: Google Scholar[1], Scopus[2], Web of Science[3] and ACM Digital Library[4].

The keywords selected for the research were: "Data Mining", "Machine Learning", "Soft Computing", "Decision Support Systems", "Modern Optimization", "Textile Manufacturing", "Fabric Manufacturing", "Textile Design", "Industry 4.0" and "Textile industry". After this first search and when the specific

[1] https://scholar.google.pt/
[2] https://www.scopus.com/
[3] https://apps.webofknowledge.com/
[4] https://dl.acm.org/

scope of the PhD research was fully defined, a second search was performed with new keywords, "Fabric Testing" and "Textile Testing", that combined with the previously selected keywords.

For the bibliographic searches, the period was set within the range of 2012 to 2021, in order to get the most relevant works in the last ten years. As for the type of publication, it should be noted that several articles were retrieved from journals that are directly related with the textile engineering scientific area (e.g., International Journal of Clothing Science and Technology). When possible, a filter was applied to select the best papers, using the Scimago Journal & Country Rank[5] ranking. We only used the number of Google scholar citations as a selection criterion when there were several papers about the same subject and presenting similarities between them. In the last step of the bibliographic search, we performed a manual inspection of several elements of the retrieved scientific documents, namely following this order of reading: titles, abstract and then conclusions. If a document was found relevant, then it would be saved for further full reading, else it would be discarded. Some relevant papers, in terms of the analyzed scientific topics, are not included in this review because they did not present results or were too vague to withdraw valuable insights.

2.3. Intelligent Decision Support Systems

A *Decision Support System (DSS)* is an information technology application that analyses raw data and assists in the execution of managerial decisions. The term DSS was first created by Anthony Gorry and Scott-Morton (1971) when they incorporated the categories of management activities of Anthony (1965) together with the description of the different types of decision proposed by Simon (1977). A DSS is often created using the four model phases as defined by Simon (1977): the first one is the Intelligent Phase, which consists in identifying problems, collecting information and the definition of goals and evaluation criteria; the Design Phase is the following step and it is responsible for determining and analyzing the several courses of action; the Choice Phase consists in selecting the best possible way; and finally the Monitoring Phase, in which it is verified if the choice is the proper one. With the evolution of technology, the definition of DSS also evolved and new types appeared.

Arnott and Pervan (2014) published a study in which they presented the current state and the different types of DSS through an analysis of the relevant literature on the subject, as shown in Figure 2. Since

[5] https://www.scimagojr.com/

this PhD is focused in the use of predictive and prescriptive analytics, it can be naturally framed as an *Intelligent Decision Support System (IDSS)*, which involves a DSS that incorporates *Artificial Intelligence (AI)* techniques (Arnott and Pervan, 2014), such as *Machine Learning (ML)* and MO (also known as Metaheuristics). The IDSS started to incorporate Knowledge-Based Systems in the 1980s. Following the growth and interest in using digital data, after the 1990s, there was a IDSS shift towards the usage of data-driven rather than expert-driven models. In particular, modern IDSS tend to incorporate predictive and prescriptive analytics (in what is often termed as Business Analytics).

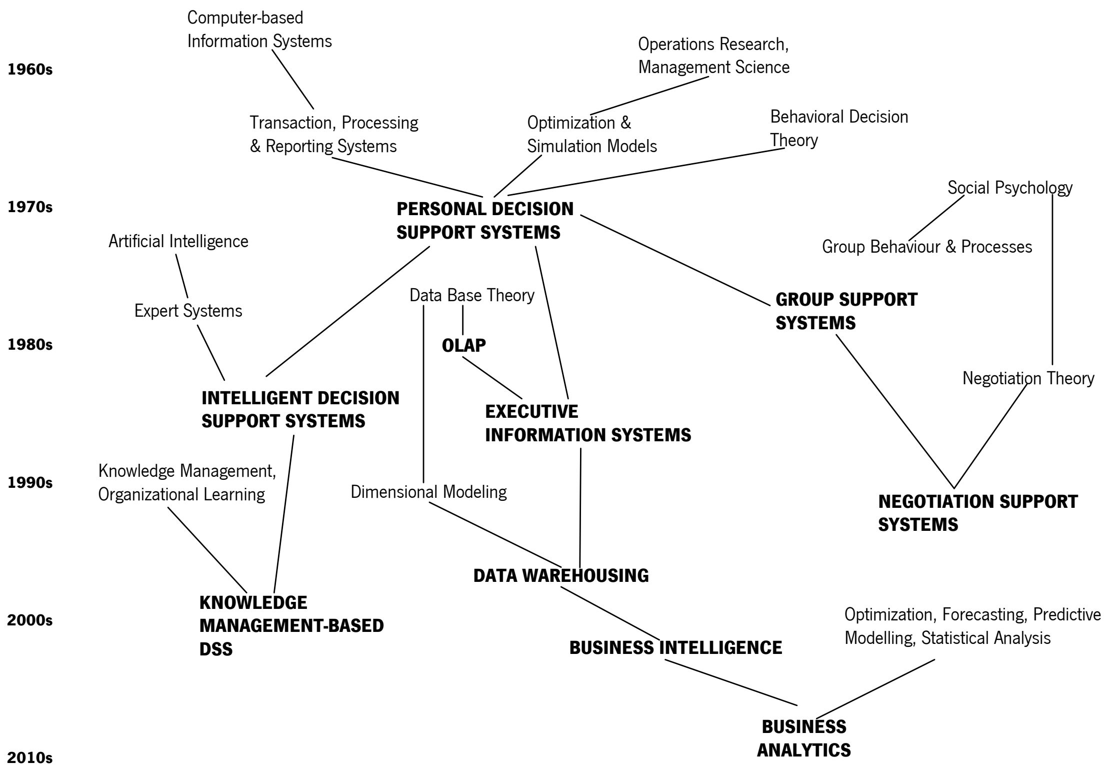

Figure 2: Evolution of decision support systems, adapted from Arnott and Pervan (2014).

The term *Adaptive Business Intelligence (ABI)* was proposed by Michalewicz et al. (2006) and it denotes an IDSS that combines prediction, optimization and adaptability. An ABI system must be able to answer two fundamental questions: "What is likely to happen in the future?" and, "What is the best decision that can be taken at the present moment?". To develop such a system is necessary to first understand the methods and techniques that enable prediction, optimization, and adaptability. This topic would not seem to be anything new at first view, since hundreds of discussions and articles have previously been published on business intelligence, data mining and prediction methods, forecasting methods and optimization techniques. None of them, however, have explicitly provided information on how to integrate these different technologies into a software system that is capable of predicting, optimizing, and adapting. ABI addresses this issue (Michalewicz et al., 2006).

Figure 3 outlines the general concept of the ABI approach. An ABI system includes elements of data mining, predictive modeling, forecasting, optimization, and adaptability, and is used by business managers to improve decision-making. An ABI system incorporates a predictive module which has the objective of creating an output based on a given input. It must also incorporate an optimization module that will recommend a set of answers. Finally, it must contain an adaptive module in which the predictions made can be modified if the environment changes, such that the prediction module can learn and adapt to changes. This module must compare the predictions made with the real data and if detects differences, it will reconfigure the predictive module to reduce those differences (Michalewicz et al., 2006).

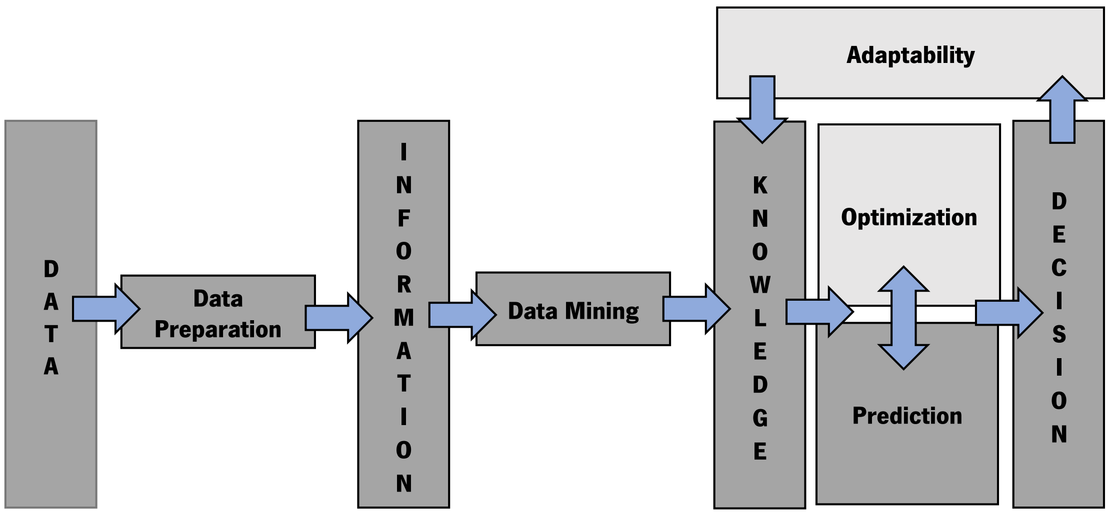

Figure 3: Adaptive business intelligence system structure, adapted from Michalewicz et al. (2006).

In 2011, the famous Gartner Group defined a data analytics maturity model (Eriksson et al., 2020), which includes four main types of analytics that aim to answer four business questions (Figure 4). The maturity model has two dimensions, difficulty (*x*-axis) and value (*y*-axis). As the level of one of the dimensions increases, so does the other. In the first step, descriptive analysis, the related question is "What Happened?", and in this case, the answer can be defined in terms of reports that provide historical insights; the second, "Why did it happen?", can be answered by the use of *Data Mining (DM)* trying to find correlations in the data; the third "What will happen?" can be answered by employing several ML techniques, and finally the most difficult one is "How can we make it happen?", which assumes the use

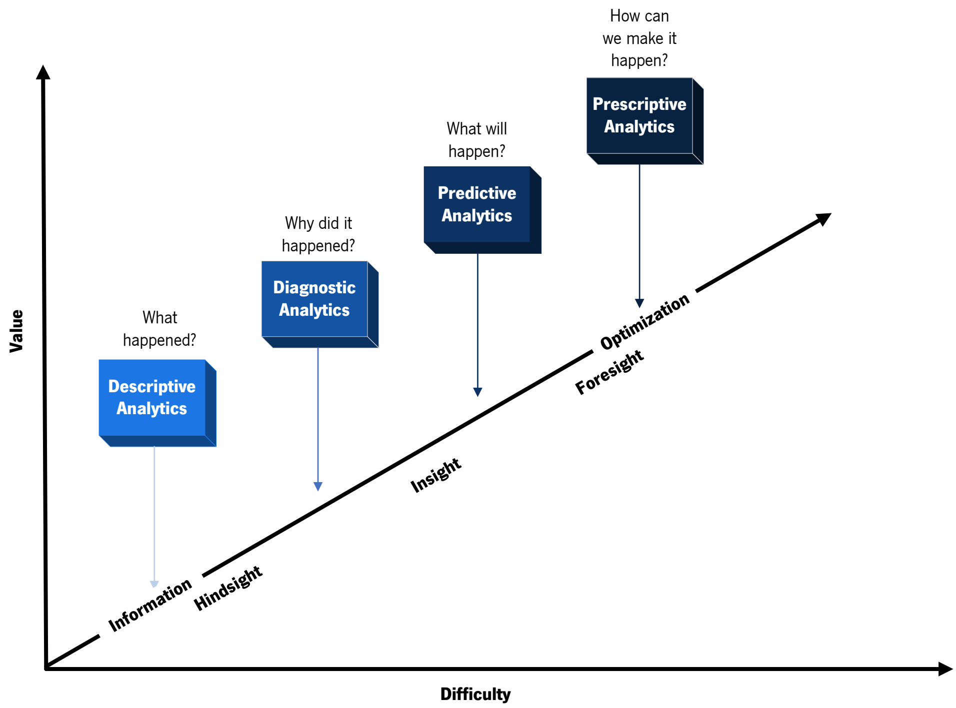

Figure 4: Gartner business analytics maturity model, adapted from Eriksson et al. (2020).

of prescriptive analytics and that can be defined in terms of what-if scenarios, simulation and optimization techniques. The Gartner model is aligned with the ABI approach. In effect, the last two questions are similar to the ones attempting to be solved using ABI, leading to what is known as predictive and prescriptive analytics. In the next sections, we detail several background concepts that are relevant to two core aspects of this PhD work: predictive and prescriptive analytics.

2.4. Predictive Analytics

The daily routine of persons and organizations produces an enormous amount of data every day. In the last decade, the evolution in technology related to processing power, speed and storage, has made the task of data analysis change from a time-consuming task made by people to a more autonomous and computer-driven task. DM is a term that refers to the extraction of implicit, previously unknown, and potentially useful knowledge from raw data (Chakrabarti et al., 2006; Han et al., 2011; North, 2012; Witten et al., 2016). DM is often used as a synonym of other knowledge extraction from data-related terms, such as *Knowledge Discovery in Data (KDD)*, Analytics, Data Science or even ML (Han et al., 2011).

KDD is formally defined as a process of identifying valid novel potentially useful and ultimately understandable patterns in data (Fayyad et al., 1996a). The term DM started appearing at the end of the 1980s (North, 2012) and KDD in 1996 with Fayyad et al. (1996b) stating that the main difference between them is in the process. In this PhD, we considered that DM and KDD are synonyms and that ML is particularly used to obtain predictive data-driven models (North, 2012; Sugiyama, 2015; Witten et al., 2016).

Different types of knowledge can be extracted from data, but sometimes the data is deficient and incomplete, and the knowledge extracted is not useful, with causal relations and patterns that are discovered that are not relevant to the business. In order to increase the success of DM projects, a standard DM methodology was created in 1999, named *Cross-Industry Standard Process for Data Mining (CRISP-DM)* (Wirth and Hipp, 2000). It considers six main stages: Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation and Deployment (Wirth and Hipp, 2000).

DM tasks often assume two main types of goals (Kantardzic, 2011): obtaining descriptive and predictive knowledge. For each goal, there are several algorithms. These algorithms were proposed from different fields such as Statistics, Pattern Recognition, AI, Database Systems and ML (Wu et al., 2008). ML provides the technical basis to DM by using several algorithms to find structural patterns in data (Witten et al., 2016).

ML can be defined as computational methods that use past data to improve a learning goal (e.g., make accurate predictions) (Fulcher, 2008; Mohri et al., 2012). In order to achieve the objectives that are proposed, the ML algorithms need data, and thus they can be classified by the type of data and the objective of learning. In particular, there are three main ML learning types: Supervised Learning, Unsupervised Learning and Reinforcement Learning. This PhD work is focused on Supervised Learning.

2.4.1 *Supervised Learning*

Supervised Learning is used in the majority of ML applications. Essentially, it is a type of learning where the correct values of the target variable are known for the historical records. The algorithms try to learn an implicit function that maps several inputs into a target output. Once the implicit function is learned, by using a training set of input and output examples, then the data-driven model can be used to produce predictions when new inputs are fed into the model (Mohri et al., 2012; Sugiyama, 2015). A real-life example of the use of supervised learning is to detect if an email message is considered spam or not. In this case, given to the spam filter is a set of emails labeled as spam or ham. The ML algorithm (e.g., Naive Bayes) will try to learn the message content differences between spam and ham and use such knowledge to estimate the spam probability for a new email.

Supervised Learning problems can be divided into two main tasks, classification and regression, depending on the type of variable that is going to be studied (Dietterich, 1997). If the variable is discrete (e.g., categorical) then it is often considered a classification task. If the variable is a numeric value (e.g., the price of a car) then it is often considered a regression task. The type of task influences the selection of the ML algorithm and its evaluation measures of predictive performance. This PhD is focused on regression tasks since the target fabric textile properties are numeric.

2.4.2 *Learning Algorithms*

In this subsection, we briefly describe some supervised learning algorithms that have been used in Textile state-of-the-art studies.

Neural Networks

Artificial Neural Network (ANN) are an information processing model that attempts to emulate the processing system of the human brain. An ANN is composed of a group of neurons that perform a simple computation, which is then passed through weight connections to other neurons. The connections have weight values that are changed in an iterative process during the training stage (e.g., via backpropagation) (Yildirim et al., 2018).

Regarding the original Multilayer Perceptron, it only contains forward connections and is grouped in layers: an input layer, one or more hidden layers, and an output layer (Torgo, 2011). The input layer size depends on the set of inputs, also known as features, used to feed the model. The hidden layers contain a user-defined number of neurons, which are responsible for the model computations. In regression tasks, the output layer contains one or more neurons, depending if a single or multi-target regression is modeled. ANN have advantages and disadvantages. They are effective to model numerical smooth relationships when there is a large number of input variables, but ANN often require more computational resources when compared with other ML algorithms, such as linear regression (Turban et al., 2008).

There are several types of ANN architectures. Up to the 2000s, the most popular feedforward ANNs types were the Multilayer Perceptron and Radial Basis Function. There are also Recurrent Neural Networks, which contain feedback or cyclic connections between the neurons (Ojha et al., 2017). Two examples of these architectures are the Hopfield Network and the Boltzmann Machine (Ojha et al., 2017). Figure 5 exemplifies three different types of ANN. In the figure, the red circles denote the input neurons that receive the data, the green neurons represent the hidden layers and the blue circles denote the output neurons.

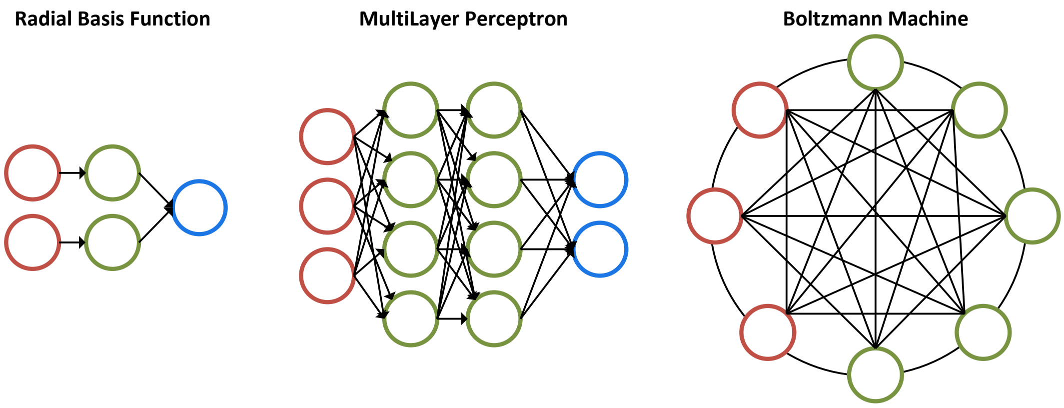

Figure 5: Example of three different neural network types.

Recently, after the 2000s, there has been a regrowing interest in the field of ANN, due to the proposal of more complex deep learning architectures. These modern deep ANN have performed remarkably well in several competitions (e.g., computer vision, natural language processing). Examples of popular deep ANN architectures include (Goodfellow et al., 2016a): Deep Feedforward Networks, Convolutional Neural Networks and Long Short-Term Memory. Such deep ANN tend to achieve particularly good predictive results when trained with very large datasets.

Support Vector Machines

Support Vector Machines (SVM) is a supervised learning model based on the theory of statistical learning (Schölkopf and Smola, 2002; Yildirim et al., 2018), presenting the concept of decision planes and structural risk minimization. A SVM constructs a hyperplane in a high-dimensional space, converting the data for prediction, and choosing the maximum margin separating the hyperplane with a defined kernel function to map the training data, into a higher-dimensional feature space (Yildirim et al., 2018). The maximum margin hyperplane gives the greatest separation between classes that are linearly separable, classifying all training instances correctly.

Figure 6 exemplifies how SVM can be applied to regression by using a ϵ-insensitive tube, set within the imaginary feature space (after the kernel transformation). In this case, the SVM algorithm only considers as support vectors the data points that are outside the ϵ-insensitive tube, thus optimizing a hyperplane that minimizes the ϵ-loss function.

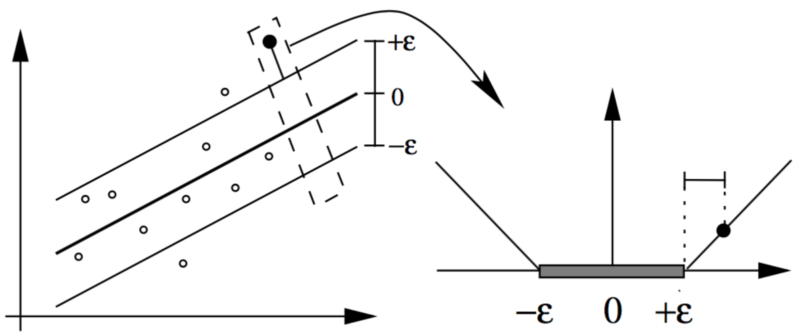

Figure 6: Example of linear SVM regression and the ϵ-insensitive loss function, adapted from Smola and Schölkopf (2004).

Decision Trees

Decision Trees (DT) consists of a supervised learning algorithm that can be used in regression or classification tasks (Breiman et al., 1984). DT models are found to be very useful for classification tasks since the models are often easy to interpret by humans and the learning algorithm is relatively inexpensive when compared with other ML algorithms (e.g., ANN or SVM) (Du and Zhan, 2002). A decision tree consists of several components: nodes, representing questions or conditions, with the top one being the root node; edges, representing the options to questions being possible to have more than 2 edges coming from a node, and leafs, being the end actions of the tree.

In Figure 7, is presented a simple decision tree, created to predict the class of a variable. Starting at the root $(x > 5)$, perform the question, follow the edge corresponding to the outcome chosen and if a leaf (red rectangles) is then found, stop and predict the outcome associated with the leaf. If it is a node that is found (blue rectangle), return to perform the question again and proceed to take the steps described above until a leaf is found. When applied to regression tasks, DT have numeric values

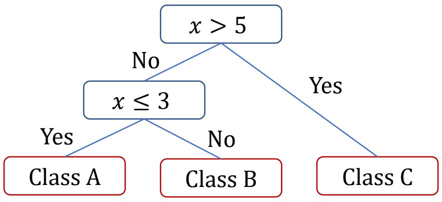

Figure 7: Example of a decision tree.

instead of class labels at the leafs. And the algorithm (e.g., Classification And Regression Tree) is often set to minimize the squared error.

Random Forest

Random Forest (RF) is a supervised learning algorithm that can be used for classification or regression tasks (Breiman, 2001). RF is an ensemble method that typically combines a large set of tree predictors, such that each tree depends on a random sample of features and training examples (bagging). The distinct trees are then combined to issue a single prediction (e.g., the average response is often used in regression).

Figure 8 presents a small example of a RF, in which data is randomly separated to each DT. Then, the RF model computes the aggregated prediction as the mean of all the individual DT estimates to predict.

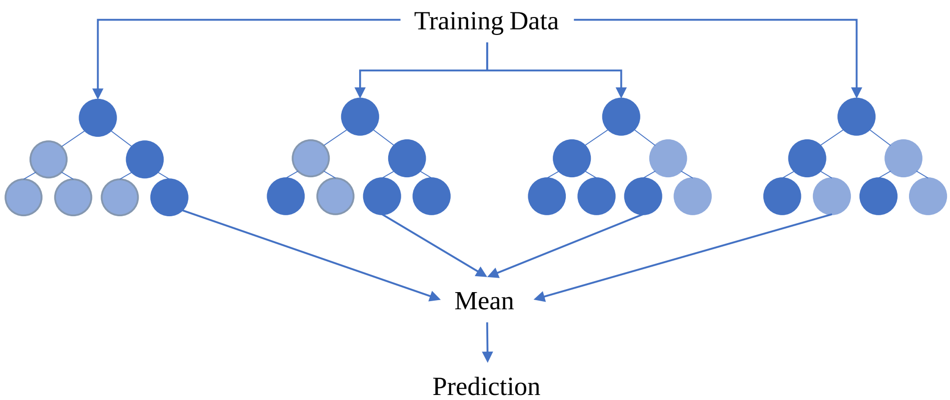

Figure 8: Example of a random forest, adapted from Ali et al. (2012).

K-Nearest Neighbors

The *K-Nearest Neighbor (KNN)* is a learning classification algorithm that determines a class label or numeric value by considering a combination function computed over the closest K (a user-defined constant) neighbors of the analyzed input (Yildirim et al., 2018). The closeness is defined by using a similarity measure (e.g., Euclidean distance) (Bhatia and Vandana, 2010). This algorithm needs to memorize all training samples and assumes that predictions should be similar to previously known cases.

In the example of Figure 9, the blue and orange circles correspond to the training data, while the red star represents new data with an unknown class. Using $k = 6$ neighbors, the algorithm detects that the most probable output label for the red star is the blue circle class.

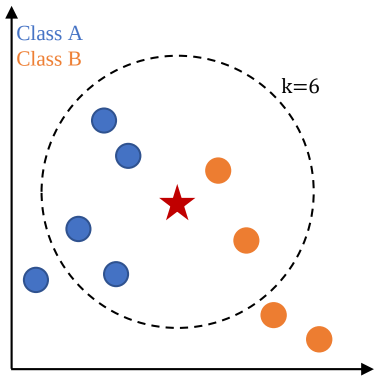

Figure 9: Example of a k-nearest neighbor.

When used for classification, the KNN often uses the majority voting function to output a single class label. As for regression, weight-averaging combination functions are mostly used. The KNN model is very simple to implement and it can work well when the dataset is relatively small. On the other hand, the KNN does not “create” a model, it requires storing all training examples and this can be very costly when working with big data.

2.5. Prescriptive Analytics

Optimization is a discipline within Applied Mathematics or Operational Research that deals with optimization models, their mathematical properties (optimization theory), and the development and implementation of algorithms (numerical analysis and algorithmic design). An optimization problem or mathematical program seeks to minimize or maximize a function (often real-valued) over a set of (decision) variables subject to constraints.

Using a mathematical optimization model requires several phases: specifying and formulating the problem, constructing a suitable mathematical model and deriving a solution from it, testing and modifying the model, and finally implementing the model solution in the real problem situation.

Optimization is a core area of the Operations Research field. The term Operations research was coined in World War II by the British military when scientists were asked to analyze and support the creation of solutions to several military problems, in connection with the deployment of radar, the configuration of convoys, and others. The term was created to distinguish research on operations from technical research (Derigs, 2009). Optimization attempts to obtain the best possible solution to the problem/model under given circumstances and it consists in finding the conditions that give the maximum or minimum value of a function (Derigs, 2009; Cortez, 2021). Optimization as a tool can be used in several different areas such as Finance, the Manufacturing industry, Airlines, Agriculture and Science.

Several complex real-world tasks contain restraints, exceptions, difficulties (e.g., nonlinear costs) and dynamics (e.g., changing environments) that are often difficult to handle by classical Operational Research methods. An alternative approach is provided by what is termed as MO (Michalewicz et al., 2006; Cortez, 2021) or Metaheuristics (Lones, 2011). This type of optimization algorithms are general purpose solvers, meaning that they can be applied to a wide range of problems that are often more complex since minimal domain knowledge is needed (Cortez, 2021). MO methods do not ensure that the optimal solution will be found, but they often achieve high-quality solutions with less computational effort.

When applying MO, two important issues need to be specified (Michalewicz and Fogel, 2013; Cortez, 2021): the representation of the solution for each problem, which defines the search space; and the evaluation function that measures the quality of the solution and that will be maximized or minimized. As for the way the search space is searched, there are three main approaches: blind, local or population-based search.

2.5.1 *Blind Search*

This section is based on the book of Cortez (2021), which contains a full chapter about blind search. Blind search, often called uniformed search, performs a search in which previously found solutions are not used to guide future searches. When using a full-blind search, all the possible solutions in the search space will be searched. Since the full search space is used, it will always find the optimal solution. The problem is that this search is not feasible for many real-world problems since the search space is too large or it is continuous. Full blind search can be easily encoded by using recursive programming. Considering the search space as a tree, the search can be depth-first, starting at the tree root and going through each branch as far as possible before backtracking, or breadth-first, starting also on the root, but searches all succeeding nodes first and then the next succeeding nodes, and so on until it stops. Since full-blind search is often too computationally expensive, there are other more reasonable variants, such as grid search, which limits the search space or Monte Carlo, which performs a finite set of random searches.

2.5.2 *Local Search*

Local search, also called single-state search, consists in searching the neighborhood of a possible solution. As stated before, searching the whole space is often not feasible. Thus, an alternative is to use the concept of neighborhood, consisting in searching for the next solution in a subset of the search space close to the previously analyzed solution. Due to its nature, local search can often get stuck at a local optimum. One way to solve this issue is to perform several local searches, each with a different initial solution (Michalewicz et al., 2006).

Local optimization techniques need to set a trade-off between the size of the neighborhood and the size of the search (Michalewicz et al., 2006). If the neighborhood size is small, the algorithm may be able to search all the possible solutions rapidly but the probability of stopping at the local optimum increases. If the neighborhood is too large, being stuck in a local optimum is less probable, but it could become impossible to compute due to the large number of solutions that need to be tested (Webster and Bernhard, 2003).

The starting solution can be generated using previous knowledge, such as problem domain heuristics, or can be set randomly, often by performing several restarts (Cortez, 2021). There are several examples of local optimization algorithms, such as Hill Climbing, Simulated Annealing and Tabu Search. Hill Climbing emulates climbing up a hill if a maximization goal is assumed. Each better solution is accepted as the current one until a local optimum is found.

Simulated Annealing is a variation of Hill Climbing and it is inspired by the metallurgy process, which consists of heating a metal and making its temperature decrease slowly in a controlled manner (Lones, 2011; Dhaenens and Jourdan, 2016). When the temperature is high, there is a higher probability to accept lower-quality solutions, behaving more like a Monte Carlo search. As the method iterates and the temperature decreases, only the best solutions tend to be accepted, similarly to what occurs with the Hill Climbing algorithm. Tabu Search is another variation of the standard Hill Climbing, using the concept of memory for storing the most recent solutions, which cannot be used (thus being "tabu") when selecting a new solution (Cortez, 2021).

2.5.3 *Population-Based Search*

Local search uses a single search point in every iteration of the algorithm. An alternative approach is to use a pool of candidate competing solutions (population). Such type of search is termed Population-Based search and it often requires more computational effort, when compared to single-state search methods, but they tend to work better as global optimization methods (Michalewicz et al., 2006).

Population-Based methods tend to explore more distinct regions of the search space and consequently, more diversity can be reached in terms of setting new solutions (Cortez, 2021). Most of these population methods are inspired by natural phenomena, such as genetics, natural selection or collective behavior of animals (Lones, 2011; Cortez, 2021). The popularity of bio-inspired Metaheuristics, especially Swarm Intelligence and *Evolutionary Algorithms (EA)*, has increased rapidly over the last two decades (Piotrowski et al., 2017).

Evolutionary Computation

Evolutionary Computation (EC) is inspired by the mechanism of natural evolution such as reproduction, mutation, recombination, and selection (Vikhar, 2016). These algorithms share a similar structure with a biological terminology associated (Lones, 2011), as is presented in the components of an EC (Vikhar, 2016; Cortez, 2021):

- **Individual** – a candidate solution;
- **Population** – group of individuals;
- **Genotype, genome or chromosome** – individuals data structure;
- **Gene** – position of a genotype, genome, chromosome;
- **Allele** – the value for a gene;
- **Fitness** – evaluation function;
- **Phenotype** – how individual operates during evaluation;
- **Breeding** – creation of new solutions;
- **Crossover** – generation of children through the combination of parents;
- **Mutation** – performing slight changes to individuals;
- **Child and parent** – child is the adjusted copy of a candidate solution (its parent);
- **Selection** – process of selecting individuals according to their fitness;
- **Generation** – one cycle of evaluation, breeding and population reassembly.

EC works on a common principle of simulated evolution of individuals using processes as reproduction, mutation, and selection (Vikhar, 2016). They can be differentiated on the basis of how they represent a solution and how the new solutions are created (Cortez, 2021):

- *Genetic Algorithm (GA)* – the most used of EC type, is based on natural selection principles, such as selection, crossover and mutation and often uses binary encoding. When numeric values are used, they are often termed EA;
- Genetic Programming – is used for evolving programs or mathematical functions, with the solutions mostly represented in terms of trees;

- Differential Evolution – similar to EA with the main difference being how mutation and crossover operators work, using mostly arithmetic operators to generate new solutions;
- Estimation of Distribution Algorithms – combines EC with ML and Statistics, and it works by estimating a probability distribution of promising solutions, using such distribution to create new individuals;
- Grammatical Evolution – similar to Genetic Programming, it evolves a variable-length string based on grammar rules (O'Neill and Ryan, 2001);
- Evolutionary Strategies – very similar to EA but it only uses mutation operators.

Swarm Intelligence

Swarm Intelligence is inspired by the collective behavior of insects and other groups of animals (bats, fishes, birds). It assumes a population of agents with direct or indirect interactions between them and the environment, that influences future behavior, making the whole swarm work as a whole, although each agent is independent (Cortez, 2021).

There are several algorithms based on this premise, being one of the most famous the *Particle Swarm Optimization (PSO)*. PSO is inspired by the behavior of animals like birds while they are searching for food. It is defined by the evolution of a population of particles in a search space (Lones, 2011). Each of the particles has a velocity, location and memory associated with it. When the optimization process starts, the velocity of the particle is changed, depending on the quality of the particle solution and the best-known position of neighborhood particles.

2.5.4 *Multi-Objective Optimization*

There are real-world problems where more than one objective needs to be optimized simultaneously. Taking as an example the Textile industry, it is often needed to reduce the quantity of waste produced, and, at the same time, increase the number of textile meters produced. This is an example of a multi-objective task, which is can be handled using three different approaches (Cortez, 2021): weighted-formula, lexicographic and Pareto front.

The weighted-formula approach is the simplest of the three methods. It is also easy to implement, consisting in assigning a weight to each goal, and then optimizing a single measure. Taking the example from Cortez (2021):

$$Q = w_1 \times g_1 + w_2 \times g_2 + ... + w_n \times g_n$$

where Q is the quality variable to optimize, $g_1, g_2, ...g_n$ are the distinct goals and $w_1, w_2, ...w_n$ denotes the assigned weights. This approach, although simple, has the disadvantage of being difficult to define the ideal weights. Moreover, when optimizing Q it will only yield a single solution.

The lexicographic approach gives a different priority to different goals, ordering the objectives by priority and optimizing first the objective with higher priority, then the others. When comparing two solutions, if the first is significantly better than the second, the former is chosen. Else, further objectives are compared, using the second highest objective and repeating this process until the best solution is found (Cortez, 2021).

Finally, the Pareto approach is a truly multi-objective approach, since it simultaneously optimizes all target variables. In this approach, instead of optimizing a single point, the method optimizes a Pareto front, which contains the set of non-dominated solutions (Lones, 2011). A non-dominated solution has a better value for one of the objectives and is at least, as good for the other objectives. After the optimization, the user can analyze the Pareto front to choose which trade-off solution is the best. This type of approach does not need ad-hoc weights or objective prioritization. Since multiple points are optimized, it is natural to use an EA to generate Pareto optimal solutions, such as executed by the *Non-dominated Sorting Genetic Algorithm II (NSGA-II)* algorithm (Cortez, 2021).

2.6. Evaluation Measures

A crucial aspect in ML is the need to compare and improve different models. In order to achieve that, a selection of functions that measure the quality of model predictions is necessary. There are several criteria to evaluate the performance of supervised learning, with different criteria being appropriate for different settings (Caruana and Niculescu-Mizil, 2004). For regression tasks, one of the most common measures is *Mean Absolute Error (MAE)*. The MAE computes the mean absolute error between the actual and the forecast values, i.e., the absolute difference between the actual and the forecast values is calculated for each of the values, and then the mean of these values is averaged and used to check how far the forecasts deviate from the actual value. It is calculated as follows:

$$MAE = \frac{\sum_{j \in \mathcal{T}} |y_{i,j} - \hat{y}_{i,j}|}{\#\mathcal{T}} \quad (1)$$

where $\mathcal{T}$ denotes the test set with a cardinality of $\#\mathcal{T}$, $y_{i,j}$ and $\hat{y}_{i,j}$ represent the desired and predicted value for output target i and test example j.

The *Normalized Mean Absolute Error (NMAE)*, presents a standardized MAE result showing the error as a percentage of the response range and is calculated using the following formula,

$$NMAE = \frac{MAE}{\max{(y_i)} - \min{(y_i)}} \quad (2)$$

The use of the NMAE is due to its ease of interpretation as an independent scale and as a percentage of the value scale. Each of these measures has a negative orientation, i.e. the closer the value is to zero the better the model will be.

Another measure often used in multiple linear regression (Harel, 2009) is the Coefficient of Determination, also known as R^2, and it measures how much the dependent variable is determined by the independent variables, in the proportion of variance (Chicco et al., 2021), and is calculated by the following formula:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(x_i - y_i)^2}{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad (3)$$

in which x_i represents the predicted value and y_i represent the real values and $\hat{y}_i$ is the mean of all the values.

Finally, the other evaluation measure used across this doctoral project is the classification accuracy for a given absolute Tolerance T. The value is based on the *Regression Error Characteristic (REC)* curve analysis and it measures the percentage of correctly classified examples when assuming a fixed absolute error tolerance (Bi and Bennett, 2003). This percentage of error tolerance is computed by considering the full range of the true values.

Regarding *Evolutionary Multi-objective Optimization (EMO)* algorithms, the *Hypervolume (HV)* measure is frequently used as a performance indicator, and as indicator-based for EMO algorithms to guide research (Shang et al., 2021). The HV of a specific set of solutions is the total size of the space that is dominated by one (or more) of the solutions in it. The HV of a set is measured by defining an anti-optimal point in space that will be used as a reference point and calculating the distance of all Pareto solutions to that point obtaining a single value that represents the volume of the objective space (While et al., 2006). The higher the HV value, the better is considered the Pareto curve optimization.

2.7. Automated Machine Learning

Over the last years, there has been a growth in ML research and applications, in particular, Deep Learning techniques have enabled significant advancements in several application fields, such as computer vision and voice processing. However, the performance of many ML systems is very sensitive to diverse design choices, presenting a significant barrier for new users with every application requiring the same procedure to be executed again (Hutter et al., 2019). And even ML specialists frequently have to go through laborious periods of trial-and-error before they identify the best modelling options for a certain dataset often using heuristics to exploit the wide dimensional space of parameters (He et al., 2018).

Most of the *Automated Machine Learning (AutoML)* frameworks focus on addressing the problem of how to automatically and choose a learning algorithm and set its hyperparameters to optimize empirical performance for a particular dataset, identified as a combined algorithm selection and hyperparameter optimization problem (Thornton et al., 2012). Several methodologies and frameworks have been developed recently to address this problem focusing on the model/hyperparameter tuning process but some approaches are now addressing the problem differently by ensembling multiple models and stacking them in multiple layers (Erickson et al., 2020).

More recently, AutoML has emerged as a form to reduce the time and effort on several repetitive tasks that are executed in ML pipelines, such as data preprocessing, feature engineering, model selection and hyperparameter optimization (Truong et al., 2019). According to Feurer et al. (2015), AutoML systems need to automatically choose a suitable algorithm and provide preprocessing stages for an available dataset, and also define their respective hyperparameter.

When analyzing several AutoML tools, Truong et al. (2019) concluded that most of the studied AutoML tools follow a common three-stage pipeline with these stages being optimized in an iterative way to obtain the best results. The first stage has two main functionalities: Data Preprocessing and Feature Engineering; the second has three: Model Selection, Hyperparameter Optimization, and Architecture Search and the third contains Model Interpretation and Prediction Analysis, with some tools providing more detailed result representation through model dashboards, feature importance and other visualization methods.

2.8. Cross-Industry Standard Process for Data Mining

In this doctoral project, the Design Science Research Methodology for Information Systems was used for the development of the IDSS, and at the same time, the CRISP-DM methodology was used to guide the DM tasks associated with predictive models. The CRISP-DM methodology is described in terms of a hierarchical process model, consisting of several tasks described at four levels of abstraction: phase, generic task, specialized task, and process instance (Wirth and Hipp, 2000). It provides an overview of the life cycle of a DM project, containing the different phases of a project, the respective tasks, and the relationships between these tasks. This life cycle is an iterative process, with a flexible sequence of phases, where the outcome of a phase will determine the next step to take, as shown in Figure 10.

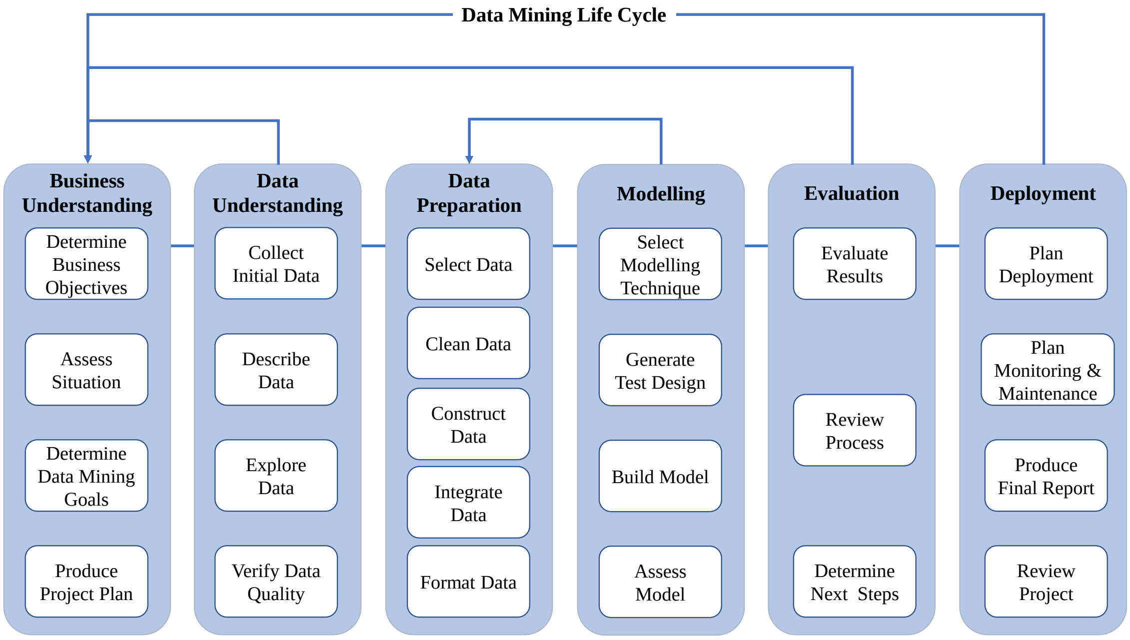

Figure 10: CRISP-DM phases, adapted from Wirth and Hipp (2000).

The methodology was developed by Chapman et al. (2000) and is widely used to improve the success of DM projects, being composed of six phases:

- Business Understanding - this is the first phase of the CRISP-DM and focuses on understanding the project objectives and requirements from a business perspective. This knowledge is then converted into a DM problem definition and a preliminary plan is designed to achieve the defined objectives.

- Data Understanding - this phase starts with an initial data collection and proceeds with several activities in order to get familiar with the data, identify possible data quality problems, obtain insights into the data or detect interesting subsets that will allow forming hypotheses for hidden information.
- Data Preparation - The data preparation phase comprehends all the activities in which the initial raw data is used for the construction of the final set of data that will be used by the modeling tool. These tasks can be performed multiple times and not in any particular or prescribed order. These tasks can include different processes such as attribute selection and data transformation and cleaning to feed the modeling tool.
- Modeling - In this phase, several modeling techniques are selected and applied, and their parameters are fine-tuned to optimal values. Since there are several techniques for the same DM problem type, is necessary to consider the specific requirements of modeling techniques, and it may be necessary to perform the data preparation phase again.
- Evaluation - At this stage, the project is expected to have a model that presents a high quality from a data analysis perspective. In this phase, the model is thoroughly evaluated and all the previously executed steps are reviewed, to guarantee it properly achieves the business objectives, and at the end of this phase, a decision on the use of the DM results should be reached.
- Deployment - This is the final phase and is the only step not belonging to a cycle. This phase varies in complexity as the project can be closed with a simple report generation or it can be necessary to implement a suite of complex models.

2.9. Industry 4.0

The fourth industrial revolution is occurring at the moment and it is characterized not just by the technical realization but by the capacity of the companies to meet current and future challenges (Drath and Horch, 2014). Industry 4.0 can be defined as an industrial vision of having “people and things connected anytime, anyplace, with anything and anyone, ideally using any path/network and any service”

(Vermesan et al., 2011). The term Industry 4.0 first appeared in Germany, originating from a project of the German government, and this notion of Industry 4.0 shares diverse common aspects with developments in other European countries where it has been labeled differently, such as Smart Factories, Smart Industry, Advanced Manufacturing or Industrial Internet of Things (Tjahjono et al., 2017). There are three key components of Industry 4.0: *Internet of Things (IoT)*, *Cyber-Physical System (CPS)* and Smart Factories (Bui and Jr., 2016).

IoT is defined as a network of many different devices that contain several different things as electronics, software, and that are connected allowing these devices to interact and exchange data over the internet. Applying this to the industrial area, IoT refers to the use of sensors, machines, instruments, and similar devices connected with computer manufacturing applications and other applications used, collecting data, exchanging it and analyzing it to improve all the different processes from the creation of a product to the supply chain.

CPS are the integration of computation and physical processes, where embedded computers and networks monitor and control the physical processes and vice-versa. CPS evolved since they appeared, passing through three different stages: the first stage was the inclusion of identification technologies, such as RFID, allowing a unique identification; the second one was the inclusion of sensors with limited functions in the CPS; and finally, in the last evolution, CPS were equipped with multiple sensors to store and collect data (Bui and Jr., 2016).

Smart Factories integrate those two ideas (IoT and CPS) in their activities, being one of the key factors in Industry 4.0. These are factories that depending on the context, assist people and machines in the execution of their daily tasks, with the help of different systems working in the background, that receive information both from the physical and virtual world (Lucke et al., 2008). Thus, Industry 4.0 is about industrial data and communication between different devices to optimize a process in a real-time situation.

2.10. Textile Industry

The Textile and Clothing industry is one of the largest industrial sectors in the world (Shishoo, 2012). The Textile industry involves different aspects such as research, design, development, production and distribution of yarn, cloth and clothing, and it uses raw materials that can be of natural origin or synthetic.

Initially, fabrics and clothing were homemade by individuals and with the purpose of personal use. Then some individuals started to resale the materials or clothing but on a very small scale. With the invention of the flying shuttle, an important step was taken in the direction of automatizing the process of weaving, this invention made it possible that one person operating the shuttle could produce more than the persons that were necessary before (Curley, 2019). With the first industrial revolution, the transition from hand-made to machine-made products, the Textile industry become a dominant industry. Now, it is considered a very complex, global industry that generates great income around the world. In 2019, the Textile and Clothing industry in the EU-28 employed 1.5 million people and generated a turnover of 162 billion EUR.

The Textile industry is comprised of different processes, but since the scope of this PhD is related to fabric manufacturing and more specifically to the design process of new fabrics, a brief description of the main processes involved is detailed here. Before presenting the topic of Fabric manufacturing, is necessary to differentiate two terms that are associated with the topic, and that sometimes are misleading (Mills, 2018). The first one is Textile, which originally was named woven fabric and now is applied generally to any one of the following: staple fibers and filaments able to be converted into woven, knit, or braided fabrics, or yarns made from natural or manufactured fibers. The second one is Fabric which can be defined as a planar textile structure produced by interlacing yarns, fibers, or filaments. The creation of fabric, independently from its final purpose, contains several phases. The process of fiber to finished fabric is depicted in Figure 11.

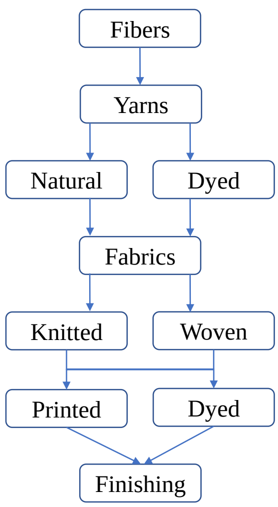

Figure 11: Flow diagram of woven fabric manufacturing.

The process starts with the raw material which can be a polymer or a material from a natural source (e.g., cotton, flax). This material passes to a process called fiber spinning (Wadje, 2009), after this if the fibers are to produce a non-woven fabric-like felt, a series of processes are applied to combine them into a cohesive fabric-like material. Fibers that are intended to produce another type of fabric are spun into yarn (an assembly of substantial length and relatively small cross-section of fibers and or filaments with or without twist (Wadje, 2009)). This process is called yarn spinning.

In the next stage, yarn can be used in its natural color or it can be dyed. To form a fabric, several methods can be used (e.g., weaving, braiding, knitting) and depending on the method used, the type of fabric will be considered knitted or woven. After this, the fabrics will pass to processes of printing or dyeing and finally, they undergo the finishing processes and are ready to be delivered to the client.

In several parts of the fabric manufacturing process, many problems can occur and the resulting product has to pass several tests to detect if the fiber or the fabric has defects. Defects are flaws in the surface of a fabric, that result from something that went wrong in the manufacturing process (Ngan et al., 2011). Those defects are problematic as they can cause delays that affect the whole production process.

The process of creating of a new fabric, either by a request from a customer or a need of the organization, starts with the definition of several components that will define the characteristics of the final product. The developer will look at the requirements of the fabric and will search for something similar that was already made as a starting point. Then she/he will try to shape that product in order to meet the requirements by changing several characteristics, such as the type and number of fibers, the pick count, the thread count and warp count, etc. Then, the process presented in Figure 11 starts. After finishing this process, it is needed to produce a small prototype of fabric, in order to verify if it passes all mandatory tests and if it meets the requirements defined initially. The prototype is then analyzed and several results are possible: the prototype passed all the tests and met the requirements and so its production may start; if the prototype did not meet the requirements, thus the developer must start again the fabric design and change some of the characteristics, or the fabric did not pass the standard tests and it must be verified if the flaw is in the conception phase or in the production phase. This process can be repeated several times until all the requirements are made, resulting in additional time and costs. It should be noted that the loom that manufactures the fabric prototype needs to produce several meters of fabric, in each design attempt.

Throughout this whole process, a large amount of data is created and stored, such as the properties of each yarn (e.g., color, thickness), the configuration of each machine used in the creation process (e.g., spinning, weaving) (Mozafary and Payvandy, 2014). This data can be used to discover previously unknown but potentially useful knowledge (Yildirim et al., 2018), presenting the research opportunity that this PhD addresses.

2.11. Predictive and Prescriptive Analytics in the Textile Industry

This section presents the core state of the art related to the use of predictive and prescriptive approaches in the Textile domain. As previously stated, the Textile industry is undergoing several transformations related to the adoption of Information Technology.

2.11.1 *Predictive Analytics*

Defects are abnormalities in the fabric, fibers or yarns that hinders the final appearance of the fabric or its purpose. Defects can be detected in any part of the process, as described in Figure 11. This is an issue that is often modelled using prediction techniques. For example, different types of ANN, (e.g., Multilayer Perceptrons, Probabilistic Neural Networks) were used to predict and classify defects: in fabric (Hasnat et al., 2017; Kuo and Juang, 2016); in a specific type of fabric like Lycra Spandex (Su and Lu, 2011); and in garment (Yuen et al., 2009). Mohanty and Bag (2017) used Association Rules to detect and classify fabric defects, using image mining. The same ML algorithm was used by Lee et al. (2013) to predict garment defects. Other data analysis techniques adopted for defect prediction were Principal Component Analysis (Eldessouki et al., 2014) and SVM (Ghosh et al., 2011).

Yarn is a generic term for a continuous strand of textile fibers, filaments, or material in a form suitable for knitting, weaving, braiding, or otherwise intertwining in order to form a textile fabric (Mills, 2018). Yarns can be of different types such as cashmere, cotton, silk and wool with the latter being one of the most common types. They can be categorized by weight and have different properties, and they are the basis of a fabric and help to determine its composition. With yarn having so much influence on the final product, and with so many distinct characteristics, it is normal that it was one of the first areas where ML was applied. In particular, Ahmad (2016) used ANN in order to predict the strength of carded cotton yarns, using different characteristics of fibers, such as length, fineness and tenacity. The same method was also used by Admuthe et al. (2009). Mozafary and Payvandy (2014) used a ANN in combination with K-means clustering to group the yarns in different classes, obtaining better results than the normal pure ANN prediction method. SVM was used to predict yarn quality in Abakar and Yua (2014), comparing favorably with a ANN. Abakar and Yua (2014) also used SVM to predict the tenacity of yarn. The authors tested two different approaches: one using a SVM and another where a GA was first used to select the features that the SVM would adopt. This hybrid approach presented better results. Furferi and Carfagni (2010) used ANN to predict the color that would result from a blend of yarns. To solve a similar task, Pan et al. (2011) used a GA.

The textile studies that used data analytics have approached objectively the challenges related to the physics of a fabric, and its subjective properties, which require human feedback. An example of a subjective texture is the feeling of human touch, which can vary from person to person. Pfrommer et al. (2018) presented a study where they attempted to predict an objective property, the shear angle of a textile composite drape. The authors used deep ANN and they slightly improved the values of a similar study. An example of a subjective textile property study was conducted by Xue et al. (2017). The authors used a GA to model the relations between tactile properties and total preference for textile products, as measured by a group of human experts.

Bergmann et al. (2017) and Jetchev et al. (2016) used Generative Adversarial Networks, which is a special type of deep learning ANN that can generate new input patterns based on historical data, to analyze several images in order to predict visual patterns on fabric and produce images with the pattern. Similar work was done by Bao et al. (2009), in which the authors proposed methods of textile image segmentation by applying a K-means clustering algorithm that was used for textile Computer-aided design.

ANN is a popular method used by several authors to predict different characteristics either on fabric, seam or even a final product. For example, Kalkanci et al. (2017) attempted to estimate dimensional measure properties of shirts and they detected that dimensional changes experienced in fabrics during garment manufacturing were considered a problem. Boubaker and Khedher (2015) predicted the consumption of sewing thread used in jean trousers with a regression coefficient of 0.973. In another study, Yildiz et al. (2013) attempted to predict the sewing thread together with elongation, comparing two ANN types. Furferi et al. (2012) proposed a ANN predictive model of a particular coating process for forecasting the final characteristics of a coated fabric. In the same year, Pattanayak et al. (2011) used

ANN to predict the five parameters of cotton woven fabrics drape. The goal was to create a profile that could be used to classify the different falling behaviors of cloth. El-Ghezal Jeguirim et al. (2011) also selected a ANN to model the relationship between manufacturing parameters, especially the finishing treatments and instrumental tactile properties. Taieb et al. (2018) also predicted fabric drapability with a ANN based on surface mechanical properties at low stress, particularly fabric formability to predict obtaining a correlation between the formability on the weft direction and drapability of 0.892 and between the formability on the warp direction and drapability of 0.886. For a different characteristic, Matusiak (2015) designed several ANN to predict the air permeability of woven fabrics obtaining a correlation coefficient that varied from 0.877 to 0.9848.

Predicting the fabric parameters that originate more defects or the ones that do not present any problem is crucial for the Textile industry. Semnani and Vadood (2010) approached such prediction based on image processing. The authors prepared eight categories of different yarn counts and types, including cotton ring spun yarn of 20 and 56Tex, cotton rotor spun yarn of 20 and 56Tex, filament yarn of 16 and 33Tex and acrylic yarn of 20 and 80Tex. The data included images with four different samples and three different kinds of fabric for each category. The data was preprocessed and fed into a ANN. A GA was used to optimize the ANN hyperparameters, which included the number of neurons in each layer and the maximum number of training epochs. The GA used crossover and mutation operators, with a probability of 0.6 and 0.4, respectively. The optimization also adopted elitism, where the best solution was always selected for the next generation, and it was stopped after 20 generations. The optimized model obtained predictions with a correlation measure of 0.997. Then, the authors asked several observers to estimate the appearance of the samples, using a scale that ranged from 0 to 100, and a high correlation (of 0.972) was computed when comparing the observers responses and the ANN outputs.

Fabric testing has a crucial role in gauging product quality, ensuring regulatory compliance and evaluating the performance of textile materials providing information on the structural, chemical and performance properties of fabrics. It refers to several procedures for assessing yarn and fabric characteristics (e.g., abrasion resistance, seam slippage, pilling) (Hu, 2008).

The tear strength is usually a measure of the force (tensile stress) required to propagate a tear and is often used to give a direct assessment of the serviceability of the fabric (Teli et al., 2008). Tear strength can be tested in both warp and weft directions and it is considered one of the most important performance attributes of woven textiles (Malik et al., 2011). In Kotb (2009), linear regression models were used to predict the fabric tearing force based on 9 identified input features to select which ones have more influence in fabric tearing, concluding that tearing force is largely affected by the type and number of weft yarns, weft density, ground structure, and ground yarns, while the shape of the pile and the change in pile designation have minor effects. In another study, linear regression was also used to predict the fabric tear strength in warp and weft direction for woven wool fabrics, using as input data from 234 woven fabric samples and the characterization of the yarns. The authors obtained a Pearson correlation between the actual and the predicted strength for warp and weft of 0.976 and 0.975, respectively (Malik et al., 2011). The same linear regression model was used in (Eltayib et al., 2016) to predict the relationship between fabric tear strength and other independent variables, such as yarn tensile strength, yarn count and fabric linear density. They used nine samples of fabrics with different linear densities and yarns and devised two equations to predict the tensile strength in the warp and weft direction. Zeydan (2010) compared four different methods: ANN, Taguchi design of experiment, Multiple Regression and a hybrid Artificial Neural Network-Genetic Algorithm modelling methodologies, to predict fabric tensile strength of woven fabrics. The author used seven different parameters related to the properties of yarns and the density of warp and weft and verified that the hybrid approach obtained the lowest MAE values.

Another important mechanical property of woven fabrics is elastic modulus, Kularatne et al. (2020) compared a Linear Regression, RF and ANN approaches using weave factor, pick density, and yarn count as input variables, with RF presenting better results with a low MAE and a Coefficient of Determination value of almost one. Air permeability is one property of textile materials that influences air flow through textile material and Baghdadi et al. (2016) used a ANN approach to predict the air permeability behavior of stretch plain knitted fabrics that were previously treated with different finishing processes and different sequences obtaining a Coefficient of Determination of 0.99 and Mean Relative Absolute Error of 6%.

Pilling is a surface defect that consists of circular accumulations of entangled fiber attached to the surface of the fabric, which affect the appearance and the handle of the fabric. Beltran et al. (2006) used an ANN with 17 inputs related to the yarn and the fabric to predict the value of the pilling test. The authors used three input feature selection techniques in combination to reduce the number of input features from 17 to 12 and then to 10 in each iteration. ANN were also used by Rejali et al. (2014) to predict the pilling performance of weft-knitted fabrics produced from wool\acrylic blended yarns. After several experiments to optimize the number of neurons in the hidden layer, the maximum and minimum errors of prediction were 4.18% and 0.24% with an average error for the number of pills of 1.92%. For the same test, Yap et al. (2010) used SVM. The authors reduced the number of attributes from the original data and transformed the pilling ratings by omitting outlier ratings of more than 0.5 grade, and rounding off to the nearest half grade. They employed the text classification method where the SVM learns from the existing data through training, and subsequent classifier models are built for prediction purposes. The Radial Basis Function kernel achieved the highest performance of 90%.

Bijoya and SB (2005) compared three modeling methodologies: mathematical, empirical and ANN based on radial basis function to predict two fabric properties: initial tensile modulus and bending rigidity. They verified that the later approach produced an error of 10.2%, as compared to 20.4% and 20.53% for warp tensile modulus and 8.63%, 12.33%, 13.65% for weft tensile modulus, in the case of fabric bending rigidity in warp and weft the prediction errors were of 8.77%, 9.22%, 10.74% and 25.35%, 18.7% and 16.9%, respectively for hybrid network, empirical model and mathematical models, concluding that the ANN based on radial basis function model was able to generalize the functional relationships between data very well. The same approach was used to predict initial tensile modulus by Hadizadeh et al. (2009), but in this case, with different inputs from the previous work by Bijoya and SB (2005), with high correlation coefficients of the fabric modulus for all cases in the warp and weft directions (between 0.90 and 0.99).

Ogulata et al. (2006) used ANN and Linear Regression Models to predict the results from elongation test and residual extension in bi-stretch fabrics in warp and weft directions. The authors used different input variables with both ANN and linear regression models and verified that when predicting the elongation values of the fabric, either one of the two models could be used. However, for residual extension, the prediction power of the two models shows a difference according to fabric direction: in the warp direction, the performance of ANN seems to be better than the regression model; in contrast, this is reversed in the weft direction.

The bursting strength of a fabric is extremely important. The fabric should have sufficient strength against forces acting upon it during dying, finishing, and general use. Ertugrul and Ucar (2000) utilized two different methodologies: a multi-layer feed-forward neural network and an adaptive network-based fuzzy inference system (a combination of a radial basis neural network and the Sugeno-Takagi fuzzy system) and verified that yarn strength, elongation, and fabric weight are the most important attributes to predict bursting strength. The obtained results were considered promising with a Sum of Squared Errors = 0.03. ANN were also used by Chen et al. (2009), but in this case to predict the shear stiffness of worsted fabrics, utilizing human knowledge on the shear stiffness and data sensitivity criterion based on a distance method to select the inputs for the ANN, with an absolute error value of nearly 12% and an average error less than 1%. ANN were also used by Murrells et al. (2009), but in this case to predict the spirality of fully relaxed single jersey fabrics. Spirality is related to bias distortion that a fabric may suffer after use. After performing several experiments to select the optimal number of neurons in the hidden layer, the authors achieved relatively good results when comparing the predictions and actual measured values with a correlation coefficient of 0.976 in out-of-sample testing. Also for modeling spirality but for single jersey cotton knit fabric, Shahid et al. (2015) used a fuzzy logic approach, taking the knitting stitch length and yarn count as input variables and obtaining a correlation coefficient of 0.991 and a Mean Relative Error of 2.41%.

Fan and Hunter (1998) used a backpropagation ANN with one single hidden layer with 30 inputs based on fiber, yarn, and fabric constructional parameters to predict nine fabric properties (e.g., abrasion, seam slippage, thickness). The authors made several transformations to the discrete and nominal inputs transforming them into -1 and 1 values. The authors compared thickness, warp bending rigidity, weft bending rigidity, and shear rigidity predicted values with the actual values, and obtained an interesting number of correct predictions, although the predictive error can be large in certain cases, due to variations or differences in fabric manufacturing, particularly dyeing and finishing.

2.11.2 *Prescriptive Analytics*

Engineered fabrics are textile materials manufactured mainly for technical and functional performances and to support the design of these fabrics, Das et al. (2014) optimized several weave parameters such as count, crimp and thread spacing of warp and weft yarns to design woven fabrics with desired quality and low manufacturing cost, this optimization problem has been solved using PSO algorithm to determine the optimum values of the different parameters for the production of light, medium and heavyweight cotton fabrics by their Grams per Square Meter weight.

The process of yarn production also known as weaving is one of the most important processes regarding fabric manufacturing. A GA was used in this process to optimize weaving machine settings by Gloy et al. (2015), effectively reducing the warp tension without impact on the mechanical properties of the fabric and reducing the occurrence of fabric defects.

The same algorithm was utilized to optimize the production planning of the weaving process aiming to reduce the overall production time (Ferro et al., 2021), providing information that will support the decision-making about batch sizing and production scheduling activities. A NSGA-II was compared with *Artificial Bee Colony (ABC)*, *Ant Colony Optimization (ACO)* and PSO to optimize several yarn characteristics (Chakraborty and Diyaley, 2018) with the latter providing the best results with respect to the objective value, consistency of the solutions and convergence speed.

After this process, the yarn can be dyed, with Junior et al. (2015) comparing two algorithms, GA and Memetic Algorithm, to optimize the scheduling of yarn dyeing, with the latter approach obtaining better results within less time when comparing with GA. Also for the same process, Elahi et al. (2022) implemented a Multi-Objective Group Counseling Optimizer II algorithm to optimize the planning, freshwater consumption and wastewater treatment costs, that when compared with manual scheduling presented a variation up to 35%.

To optimize the tardiness cost of dyeing machines scheduling and simultaneously minimize the utilization rate of dyeing vats utilized in the dyeing process, Zhang et al. (2017) proposed a Multi-Objective Artificial Bee Colony algorithm that provided better results when compared with a generic multi-objective scheduling algorithm in terms of both solution quality and computational time robustness.

Concerning the fabric design stage, and with the intent of obtaining cotton knitted fabrics with the desired level of UV protection, NSGA-II was used to simultaneously optimize the air permeability and thermal conductivity values with four input parameters: loop length, carriage speed, yarn input tension and yarn count Majumdar et al. (2017). For different purposes, an Interactive Genetic Algorithm was used to optimize patterns with two or three colors of yarn-dyed plaid fabric (Zhang et al., 2020) to provide design references to the fabric designer.

Garment manufacturing is considered one of the last stages in the Textile process, with several approaches used in different areas such as: optimization of marker planning that was addressed using a GA (Xu et al., 2020) approach, and the authors validated the proposed system in a real industrial environment, with the results showing that the system improves the garment fit for a target population with a limited increase of cutting cost. For the same challenge, Tsao et al. (2021) proposed a Hybrid Genetic Algorithm-Simulated Annealing approach that saved almost 28% of fabric length. Also for garment manufacturing, Jaouachi and Khedher (2021) compared three approaches ACO, ABC and PSO to optimize the consumed amount of sewing thread required to produce a pair of jeans, with the ACO and PSO approaches providing the lowest consumption values.

Several studies are focused on the optimization of a production plan or part of it, with a multi-objective optimization using a parallel evolution and scenario generation-based Multi-Objective Evolutionary Stochastic Optimization approach used to optimize the production plan for printing and dyeing areas (Zhang et al., 2021), also in the dyeing process of fabrics, a Multi-Subpopulation Genetic Algorithm with heuristics embedded was developed to improve scheduling of the dyeing orders (Huynh and Chien, 2018); and in three different areas, Liyanage et al. (2020) used a Genetic Programming approach for the sewing, quality assurance and finishing areas, with the three areas being individually optimized. Also related to production but in this case, regarding all the production stages of a garment manufacturer, Lorente-Leyva et al. (2019) optimized the master production scheduling for the production of several types of t-shirts and uniforms with a GA approach.

2.11.3 *Predictive and Prescriptive Analytics*

One of the main processes in the Textile domain is yarn manufacturing, where Ghosh et al. (2013) used NSGA-II to maximize cotton yarn strength at minimum raw material quality, using six fiber properties as inputs to a ANN to predict the yarn strength and was capable of finding optimal solutions for the production of stronger cotton yarns at minimum costs.

Another challenge present in the Textile industry is the design of fabrics with certain characteristics. To address this challenge, a hybrid Artificial Neural Network Genetic Algorithm was implemented by Mitra et al. (2015), to optimize a fabric thermal resistance, using the drape coefficient and air permeability as constraints, selecting the ends per inch, picks per inch, warp count and weft count as inputs of three ANN models that will be utilized by the GA; the same approach was used to create a fabric with optimized values for ultraviolet protection factor, air permeability and moisture vapor transmission rate using the proportion of polyester in the fabric, warp count, weft count, end density and pick density in fabric as inputs (Majumdar et al., 2016).

Intending to develop a system to utilize successful experiences and help beginners of garment pattern design, Hu (2009) created a hybrid system based on ANN and Immune Co-Evolutionary Algorithm. The authors had five body parts as parameters, including the measures of the waist, hip, crotch, thighs and knees from a tester and then they attributed a score to calculate average fit. Hip line, pants length, total inside seam angle, up of back waist, front crotch or groin thickness and back width at the hip line were also measured. They used a ANN to create an approximate mapping between sizes and fit. The trained ANN error was used as a fitness function for the algorithm optimization. The proposed approach was then used to predict the fit of the garment and search for optimal sizes. Although the authors affirm that the proposed research was feasible and valuable for the textile domain, the study was more argumentative and it did not perform any experiments with real-world data.

The selection of appropriate spinning parameters was addressed by Lv et al. (2011). The motivation was the need to estimate yarn quality from existing spinning data in advance, attempting to discover the relationship between the spinning variables and the yarn properties. The authors used several parameters related to the characteristics of fiber (mean fiber diameter, diameter distribution, hauteur, etc.) as input features. The two most important quality characteristics of the resulting yarn, namely the elongation at break and break force, were used as target output variables of a SVM model. To select the best values for the sparsity parameter, the penalty term and the width of the Gaussian kernel parameter, a GA was selected. The authors compared the SVM results with a ANN, with the SVM performing better for both output targets (break force and elongation break), presenting a correlation coefficient of 0.99 for the first output and 0.87 for the second one. However, it should be noted that the study used a small data sample size, with just 26 examples. The authors argue that such a small dataset reflects the real-world production of the analyzed textile company and that the SVM model was capable to provide accurate predictions under such data size restrictions.

2.12. Summary

This chapter presented the necessary background for this PhD work, including the research topics, the involved concepts and state-of-the-art studies applied to the Textile domain. After the introduction of the main theoretical concepts, the studies regarding the use of predictive and prescriptive analytics (or both) in the Textile domain were presented. The purpose is to understand what ML and MO techniques current state-of-the-art works applied to similar works as the one of this PhD, and to identify research opportunities.

Table 1 provides a summary of the state-of-the-art of these studies, focusing on the fabric design process and textile quality tests. The table is chronologically ordered and it includes five columns: **Study** – the bibliographic reference; **TS** – the textile stage where the study was performed; **Objective**

– the main goal of the study; **PM** – the predictive methods used; **OM** – the optimization methods used; and **Analytics** – type of data analytics used (predictive, prescriptive or both).

Table 1: Summary of the relevant state-of-the-art studies.

Study	TA[a]	Objective	PM[b]	OM[c]	Analytics
Lv et al. (2011)	FM	Prediction of spinning quality	SVM	–	Predictive
Malik et al. (2011)	FD	Prediction of tear strength	LR	–	Predictive
Pattanayak et al. (2011)	FD	Prediction of five parameters for cotton woven fabrics drape	ANN	–	Predictive
Ghosh et al. (2013)	FM	Optimize quality and price of yarns	ANN	NSGA-II	Predictive, Prescriptive
Abakar and Yua (2014)	FM	Prediction of the tenacity of yarn	SVM	–	Predictive
Das et al. (2014)	FD	Optimize grams/m^2 value	–	PSO	Prescriptive
Mozafary et al. (2014)	FM	Prediction of yarn quality	ANN, K-means	–	Predictive
Rejali et al. (2014)	FD	Prediction of pilling	ANN	–	Predictive
Mitra et al. (2015)	FD	Optimize drape coefficient, air permeability,thermal resistance	ANN	GA	Predictive, Prescriptive
Matusiak (2015)	FD	Prediction of air permeability	ANN	–	Predictive
Shahid et al. (2015)	FD	Prediction of spirality	FL	–	Predictive
Ahmad (2016)	FM	Prediction of strength of yarns	ANN	–	Predictive
Baghdadi et al. (2016)	FD	Prediction of air permeability	ANN	–	Predictive
Eltayib et al. (2016)	FD	Prediction of fabric tear strength	LR	–	Predictive
Majumdar et al. (2016)	FD	Optimize UV protection factor, air permeability, moisture vapor rate	ANN	GA	Predictive, Prescriptive
Hasnat et al. (2017)	FM	Predict fabric defects	ANN	–	Predictive
Majumdar et al. (2017)	FD	Optimize air permeability, thermal conductivity	–	NSGA-II	Prescriptive
Pfrommer et al. (2018)	FD	Optimize the textile draping process	DNN	–	Prescriptive
Taieb et al. (2018)	FD	Prediction of fabric drapability	ANN	–	Predictive
Kularatne et al. (2020)	FD	Prediction of elastic modulus	ANN, LR,RF	–	Predictive

The following are the keywords used in the table:

[a] *FD* - Fabric Design; *FM* - Fabric Manufacturing; *GM* - Garment Manufacturing.

[b] *ANN* - Artificial Neural Network; *DNN* - Deep Neural Network; *FL* - Fuzzy Logic; *LR* - Linear Regression; *RF* - Random Forest; *SVM* - Support Vector Machine.

[c] *GA* - Genetic Algorithm; *NSGA-II* - Non-dominated Sorting Genetic Algorithm II; *PSO* - Particle Swarm Optimization.

The table confirms that most studies only address predictive analytics. In effect, prescriptive analytics were approached in five studies, with three of those studies utilizing ML techniques combined with MO methods (Ghosh et al., 2013; Mitra et al., 2015; Majumdar et al., 2016).

Several works address the prediction of textile laboratory quality tests, mostly using a ANN algorithm, with only three studies using different techniques such as Linear Regression in the works of Malik et al.

(2011) and Kularatne et al. (2020), and Fuzzy Logic in the work of Shahid et al. (2015). However, all of these studies are focused in a single laboratory test, and the remaining predictive studies are related to processes within the manufacturing of fabrics. Also, within our knowledge, there are no studies that have modeled the final textile composition (e.g., % of cotton, % of elastane) based on the fabric construction features or other textile-related features.

Regarding the prescriptive studies, the majority of the analyzed studies use EA and are focused on the fabric design process. We also verified that in what concerns the use of Evolutionary Multi-Objective Optimization methods, the research is very scarce, with most of the works using a single objective approach. Moreover, the state-of-the-art studies approach some of the textile design and manufacturing steps, as described in Figure 11, but not the entire process, with no study using the finishing operations as input feature or output task.

To the best of our knowledge, there are no studies that support the whole process of textile design of new fabrics based on data analytics. In particular, that use predictive models to estimate several textile properties of the fabric and then use prescriptive methods, attempting to optimize the input values to fed the predictive model such that the desired client properties are reached. Thus, this PhD work aims to fill this research gap, researching on the design of an IDSS that incorporate such analytics, using data from a real textile company. In particular, we intend to use data from all the different stages in the fabric design process, and in using Evolutionary Multi-Objective Optimization methods for the prescriptive tasks.

3

INITIAL PREDICTIVE AND PRESCRIPTIVE ATTEMPTS

This chapter presents the main methods, experiments and results obtained throughout our initial predictive and prescriptive iterations and is divided into five sections. Section 3.1 describes the research context of the experiments. Sections 3.2 and 3.3 present the predictive experiments. Then Section 3.4 presents the first prescriptive experiment. Finally, in Section 3.5 the conclusions of these experiments are withdrawn.

3.1. Research Context

This PhD was inserted in a *Research and Development (R&D)* project related to Industry 4.0 transformation in the Textile industry which partly coincided with the first three years of this doctoral work. The textile company involved in the project expressed the need for an *Intelligent Decision Support System (IDSS)* that could reduce the number of iterations that currently are executed to design a new textile fabric. This IDSS will provide support to the textile designer by using predictive analytics to estimate several fabric properties (e.g., elasticity, bias distortion) and then prescriptive analytics to optimize the fabric construction inputs that feed the predictive models.

In order to implement the IDSS, we first developed a method to predict the values of two laboratory quality tests using a *Automated Machine Learning (AutoML)* approach (see Section 3.2). This initial exploratory study provided initial insights about which features and data preprocessing methods present better results when using textile data and also proved that is possible to predict the outcome of laboratory quality tests with interesting results.

Then, a second comparison study (see Section 3.3) was performed. In this study, we increase the number of analyzed final physical properties, modeling nine laboratory tests. Also, we analyzed a new data transformation method for categorical variables. In particular, the second research study involved two iterations, one to compare the yarn feature representation obtained in the previous study with another yarn feature representation, and the second to compare the inclusion of a newer set of input features that other studies considered a challenge. The study achieved interesting predictive results for the nine fabric properties, and confirmed that was possible to use *Machine Learning (ML)* methods to predict laboratory quality tests of fabrics. As a result of this research, a final set of input features was selected, which includes a proposed representation form for the yarns used in the fabric and the finishing operations that were adopted to finalize the textile product.

We further note that both the first and second predictive studies used the textile composition values (e.g., % of cotton) as inputs of the predictive models. These inputs were included because they were available in the analyzed datasets and, when the studies were conducted, it was our knowledge that such inputs could be easily defined by the textile designer. Moreover, the Textile company suggested that the inclusion of the composition values could potentially improve the predictive results. However, on a later stage of this research it become clear that the setting of the correct textile composition values is a nontrivial task that is dependent on the types of yarns used in the warp and weft elements of the fabric. Thus, in Chapter 4 we corrected this issue by first removing the textile composition values from the inputs of the physical property prediction models and then defining a secondary prediction goal, the estimation of the textile composition based on the same set of inputs adopted by the physical property prediction models.

After the conclusion of the two initial predictive studies, the R&D project ended. Another R&D project, also related to Industry 4.0 transformation in the Textile industry, termed Connect@Fashion, included as one of its requisites the development of an IDSS that could predict and optimize a production planning with the capability of adjusting dynamically over time as new data is generated. Since the main objective of this newer R&D project was similar to the one proposed in this PhD work, we utilized the same main methodology and approach that was defined for this PhD. The Connect@Fashion project had a duration of 12 months and it was developed in collaboration with INFOS, a Portuguese software company that works with several Textile industry clients. The company developed an *Enterprise Resource Planning (ERP)* that supports the production of garments. Our main goal was to develop an IDSS based on the *Adaptive Business Intelligence (ABI)* concept to be integrated into the INFOS ERP system, allowing it to automatically design garment manufacturing subcontractor plans regardless of the size of the company and the complexity of the production order. We then started the implementation of a similar IDSS to the one proposed in this PhD work. The purpose was to study and gain expertise on the application of *Evolutionary Multi-objective Optimization (EMO)* approaches in this challenging area. Section 3.4 presents the results of this initial attempt of developing an IDSS for the Textile industry. Finally, we would like to note that whenever possible, in this chapter the original text of the published papers was maintained, with minor changes to some specific textile terms to maintain coherence across all document. We further note that since these works were published while the unified data warehouse was being implemented, there are slight changes regarding the volume of data used in each study.

3.2. Prediction of Two Laboratory Quality Tests[1]

3.2.1 *Introduction*

The Textile and Clothing industry is one of the largest industrial sectors in the world. However, the Textile market is highly competitive and there is pressure to improve production processes and reduce costs. Under this context, this industry can be enhanced by adopting the Industry 4.0 concept, which assumes the digitalization of the productive processes (e.g., digital sensors with connectivity capabilities) and a stronger usage of Information Technology.

In order to create the final textile product, the raw materials undergo a series of processes, where fibers are combined into yarns and the combination of these yarns creates a fabric, which receives a series of treatments, creating the final product that is delivered to costumers. During this procedure, a large amount of data is created and stored, such as the properties of each yarn (e.g., color, thickness), the configuration of each machine used in the creation process (e.g., spinning) and the results of the fabric quality tests. All these data can be processed by *Data Mining (DM)* and ML methods, allowing the discovery of valuable knowledge in order to improve the textile manufacturing process.

This set of experiments presents an implementation of the *Cross-Industry Standard Process for Data Mining (CRISP-DM)*) methodology, based on AutoML, to predict the results of tear strength test (warp

[1]This section integrates the content of the research paper "Predicting the Tear Strength of Woven Fabrics Via Automated Machine Learning: An Application of the CRISP-DM Methodology", published in the Proceedings of the 22nd International Conference on Enterprise Information Systems (ICEIS 2020) (Ribeiro et al., 2020). **Acknowledgments** This work was carried out within the project "TexBoost: less Commodities more Specialities" reference POCI-01-0247-FEDER-024523, co-funded by *Fundo Europeu de Desenvolvimento Regional* (FEDER), through Portugal 2020 (P2020).

and weft directions) on fabrics. The data were collected from a Portuguese Textile company, aiming to reduce the number of attempts required to produce a fabric.

3.2.2 *Materials and Methods*

In this initial work, we use data collected by a Portuguese Textile company, aiming to predict the tear strength test, at both warp and weft directions, of fabrics. This Portuguese Textile company creates and produces fabrics for fashion and clothing collections from diverse customers. The current fabric design is based on the designer experience and intuition and several trial-and-error fabric sample production experiments. When designing new fabrics, in order to meet the requirements of the client, the company produces several small sample attempts. In each attempt, several laboratory tests are used to verify if the fabric complies with quality goals. If this sample is not approved, the design process must be repeated, which is translated into more time and costs for the company. The fabric design process generates data that is related to the several components of the fabric, as well as the quality test results.

Within our knowledge, this is the first Textile industry study that employed an AutoML procedure, which automatically tested five families of flexible regression algorithms during the Modeling stage of CRISP-DM. Such an automatic selection of the best ML method allowed us to perform different CRISP-DM iterations more quickly (described in Section 3.2.2 to Section 3.2.4), after obtaining feedback from the textile company and aiming to explore different data and feature engineering approaches.

Computational Environment

All executed experiments were conducted in two different open-source computational environments: the R statistical tool and its `rminer` package, which facilitates the use of DM techniques ML result analysis (Cortez, 2010); and `H2O`, which implements an easy to use AutoML algorithm (Landry and Bartz, 2022). The AutoML was configured to automatically select the regression model and its hyperparameters based on the best *Mean Absolute Error (MAE)* over a validation set, using a 10-fold cross-validation that is applied over the whole training data. A total of five different regression families were automatically compared by the AutoML. These include three individual base learners, namely *Generalized Linear Models (GLM)*, *Gradient Boosting Machines (GBM)* and Distributed Random Forests, and two stacking ensembles, one using all trained models (Stacking All) and other using just the best model per ML base algorithm (Stacking Best).

First CRISP-DM Iteration

In this iteration, we performed the first five phases of the CRISP-DM, Business Understanding to Evaluation, aiming to predict the two fabric tear strength targets (warp and weft).

Business Understanding

The Textile company expressed the need to reduce the number of attempts that were necessary to produce a fabric sample. The two fabric tear strength numeric attributes (warp and weft directions) were also identified as relevant prediction targets, thus setting two regression tasks. We also selected the computational tools (R and `H2O`), as detailed in Section 3.2.2.

Data Understanding

First, we analyzed the Textile company two main fabric data sources: the ERP, which included the 88,653 fabric main data records, and the laboratory testing database, which contained the fabric quality tests performed between February 2012 to March 2019. After merging the two data sources, the resulting dataset had 12,088 examples for the warp test and 12,143 for weft. Table 2 summarizes the initial set of input attributes, as suggested by the Textile company. Most attributes are numeric and the exceptions are the type of fabric and yarn code. The last 5 rows are related to yarn attributes. We note that each fabric can include several types of yarns, which is a relevant issue that is handled in Section 3.2.2. Figure 12 shows the box plot distribution of the fabric weft and warp tear strength.

Data Preparation

A Data Warehouse system was implemented, in which an *Extraction Transform Load (ETL)* process was used to merge the two sources of data and preprocessing some data records. The preprocessing included the removal of fabric records with missing components (e.g., with not registered yarns). Also, in some cases it was detected that the same fabric had different quality test values, related with repeated tests conducted at different fabric production stages. In order to have a single test value per fabric, the

Table 2: List of input attributes used for regression.

Name	Description (data type)	Min.	Max.	Average
T_cm	Number of finished threads per centimeter (numeric)	18	1,321	115.60
P_cm	Number of finished picks per centimeter (numeric)	7	510	88.87
weight/m^2	Weight (in grams) per square meter (numeric)	22	1,690	241.70
finished width	Width in centimeters (numeric)	90	168	140.00
weave design	Weave pattern of the fabric (nominal with 21 levels)	-	-	-
reed width	Width of the reed in centimeters (numeric)	30	242	188.80
denting	Number of the reed dents per centimeter (numeric)	0	252	126.80
ends/dent	Number of yarns per dent (numeric)	0	88	2.30
n_picks	Number of picks on loom per centimeter (numeric)	0	81	16.50
weft code	Identification code of the weft (nominal with 6,883 levels)	-	-	-
warp code	Identification code of the warp (nominal with 5,353 levels)	-	-	-
warp total ends	Total number of threads on the warp (numeric)	477	21,858	6,950.00
yarn code	Identification code of the yarn (nominal with 11,020 levels)	-	-	-
n_folds	Number of single yarns twisted (numeric)	1	12	1.60
yarn count	Mass per unit length of the yarn	2	268	47.70
yarn usage	If the yarn is used in warp or weft (binary)	-	-	-
yarn repetitions	Number of yarn repetitions in warp or weft	1	8	1.42

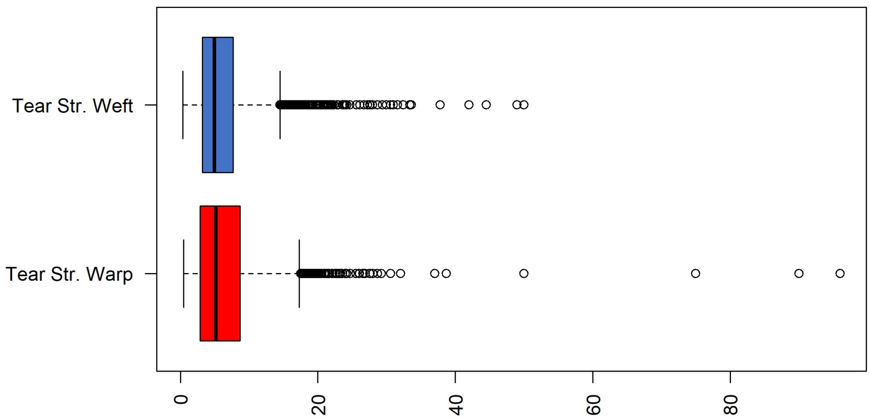

Figure 12: Box plot of the test targets used in first CRISP-DM iteration.

distinct test values for the same fabric were averaged. The resulting preprocessed data included 8,453 observations for the warp test and 8,423 examples for the weft tear strength.

Each fabric can include several types of yarns. In this work, we propose a novel input combination of features in which we include the sequence of all possible yarns (up to 9 in our dataset), for both warp and weft. Since each yarn is represented by 5 features (Table 2), the regression models are fed with $12+9\times5\times2$ (warp and weft)$=102$ input variables. A zero padding (i.e., addition of zero values to missing elements) was performed on all fabrics that had less than 18 yarn codes. Finally, before feeding the data to the ML algorithms, the numeric input attributes were standardized to a zero mean and one standard deviation, while the nominal variables were transformed using the one-hot binary encoding, which sets one binary variable per possible level.

Modeling

To evaluate the models, an external holdout split was executed, in which the data was randomly divided into training (75%) and test (25%) data. The quality of the predictions was measured using (Witten et al., 2016; Cortez, 2021): MAE, Adjusted R2 (Adj. R2) and classification Tolerance. For MAE, the lower the values, the better are the predictions. Regarding Adj. R2 and Tolerance, higher values indicate better predictions. Adj. R2 is often used in multiple linear regression and it ranges from 0 to 1. The Tolerance value is based on the *Regression Error Characteristic (REC)* analysis and it measures the percentage of correctly classified examples when assuming a fixed absolute error tolerance (Bi and Bennett, 2003). In this work, three tolerance values were set: 5%, 10% and 20%. We note that the percentage of error tolerance is computed by considering the range of the true values.

Using only training data, the AutoML procedure was applied, as described in Section 3.2.2. Figure 13 show the REC curves and respective MAE values, computed using validation data, for the best five ML algorithms that were obtained when using the internal 10-fold procedure for the warp and weft tear strength predictions.

The REC curve shows the error tolerance on the *x*-axis versus the percentage of correctly predicted points within the tolerance on the *y*-axis. In all AutoML experiments conducted in this study, and for both targets, the selected ML algorithm was a stacked ensemble that used all trained models (Stacking All).

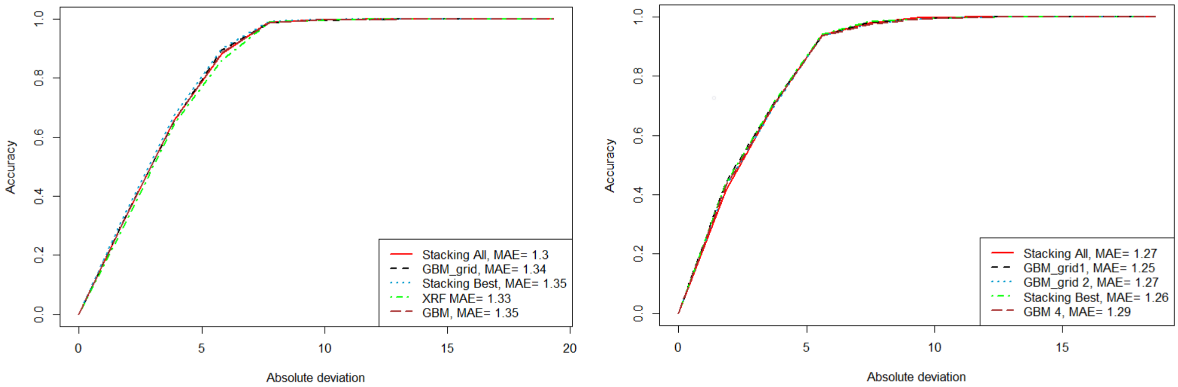

Figure 13: AutoML validation REC curves for the warp (left) and weft (right) tear strengths).

Evaluation

The obtained test set predictions are shown in Figure 14, in terms of the predicted (y-axis) versus real (x-axis) values. The plots show an interesting initial fit, with most points being close to the perfect prediction (the red diagonal line), although there are high errors, particularly when the real target values increase. This behavior alerted the Textile company experts for the need to discard outliers, which was addressed in the second CRISP-DM iteration.

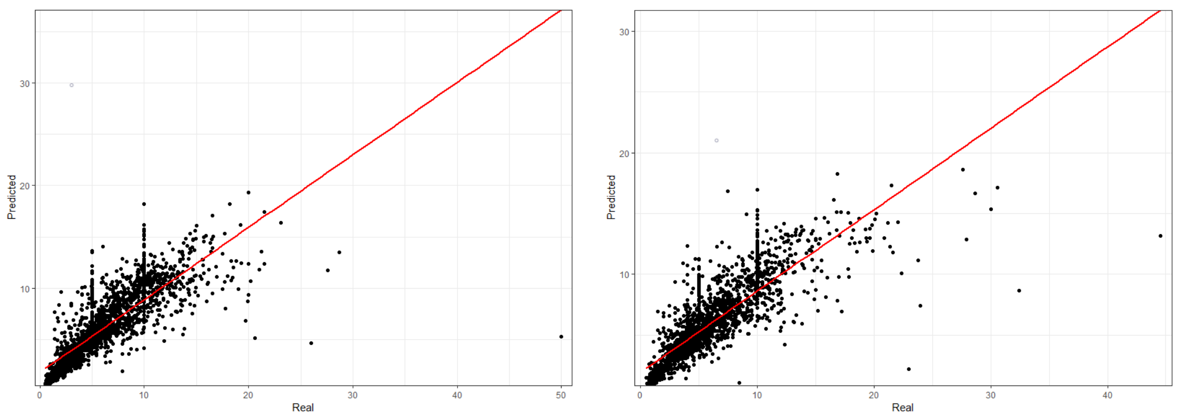

Figure 14: Regression scatter plot for the first CRISP-DM iteration warp (left) and weft (right) tear strength predictions.

3.2.3 *Second CRISP-DM Iteration*

In order to improve the previous results, a new iteration of CRISP-DM was defined. During a new Business Understanding phase execution, the Textile company provided a list of business normal ranges for the tear strength values. Thus, all test values that were inferior to 0.4 and superior to 25 were discarded in a new Data Preprocessing stage, since these tests were considered outliers (e.g., related with special uncommon military fabrics). The resulting dataset included 8,431 observations for the warp shear strength target and 8,399 examples for the weft one. Figure 15 presents the distribution of the two analyzed targets.

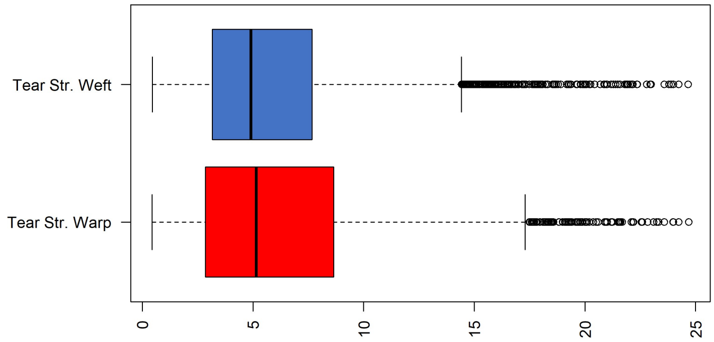

Figure 15: Box plot of the test targets used in second CRISP-DM iteration.

We then executed the Modeling and Evaluation phases, similarly to what is detailed in the previous Subsection. When the predictive results (presented in Section 3.2.5) were shown to the textile company, they were considered more satisfactory. However, the textile experts suggested a new CRISP-DM iteration, which would test the impact of using the overall composition of the fabric as a useful and extra input element.

3.2.4 *Third CRISP-DM Iteration*

In the third CRISP-DM iteration, we tested if the final composition of the fabric (e.g., the overall percentage of cotton and polyester), as an extra input feature, could improve the results from the second iteration. During a new Data Understanding phase, we collected the final composition attribute which contained 1,164 distinct levels, was treated as nominal, being thus preprocessed using the one-hot transform. The remaining CRISP-DM iteration were executed similarly to the second CRISP-DM iteration (e.g., with outlier removal), except that the predictive models used a total of 103 input variables (and not 102).

3.2.5 *Results*

Table 3 presents the overall predictive results for the test data (25%) and the three CRISP-DM iterations. For comparison purposes, we also tested a baseline method that is equivalent to the first CRISP-DM iteration except that it uses the classical multiple linear regression model, as implemented in the `rminer` R package (Cortez, 2010). In all three CRISP-DM iterations, and as previously explained, the AutoML selected model was the ensemble that included all searched AutoML models (Stacking All).

Table 3: Overall predictive results for the test data (best values in **bold**.)

Test	Iteration	Target Interval	Regression Measures				
			MAE	Tol. 5%	Tol. 10%	Tol. 20%	Adj. R^2
Tear warp	Baseline	[0.52,50.00]	2.06	2%	4%	7%	0.44
	1	[0.52,50.00]	1.30	6%	12%	23%	0.68
	2	[0.44,24.00]	**0.70**	**8**%	**16**%	**29**%	**0.92**
	3	[0.44,24.00]	1.20	6%	14%	23%	0.75
Tear weft	Baseline	[0.50,44.52]	1.92	2%	4%	8%	0.49
	1	[0.50,44.52]	1.27	5%	**12**%	20%	0.69
	2	[0.56,24.25]	1.18	5%	11%	22%	0.71
	3	[0.56,24.25]	**1.16**	**6**%	**12**%	**21**%	**0.72**

The analysis of the tear strength warp results shows an improvement from the first to second CRISP-DM iteration but not from the second to the third one. In effect, the best predictive results (for all regression measures) were achieved during the second CRISP-DM iteration, showing that outlier removal is beneficial when predicting the warp test, although there is no gain in including the final fabric composition as an input variable. Regarding the tear strength weft, the results confirm the progress of the CRISP-DM iterations, where each iteration resulted in a lower MAE value. Also, the Adj. R^2 values improved in a similar way. Thus, the best prediction results were obtained in the third iteration, which also corresponds to the best classification Tolerance for all 5%, 10% and 20% values. This confirms that removing outliers and using the final fabric composition is valuable for improving the weft quality predictions.

As for the baseline results, they are clearly worst when compared to the AutoML method and for both prediction goals, confirming that the regression tasks are nonlinear. The MAE differences are higher when comparing the linear method with the AutoML results for the first CRISP-DM iteration than when comparing different AutoML CRISP-DM iterations (e.g., the differences are 0.65, 0.08 and 0.02 for the weft test), which clearly backs the AutoML as an interesting modeling method.

To complement this analysis, Figure 16 plots the REC curves for the predictive models from Table 3. The plots include also the Normalized Regression Error Characteristic value for each curve (the higher, the better). The REC curves confirm the best performance of the third CRISP-DM iteration model for

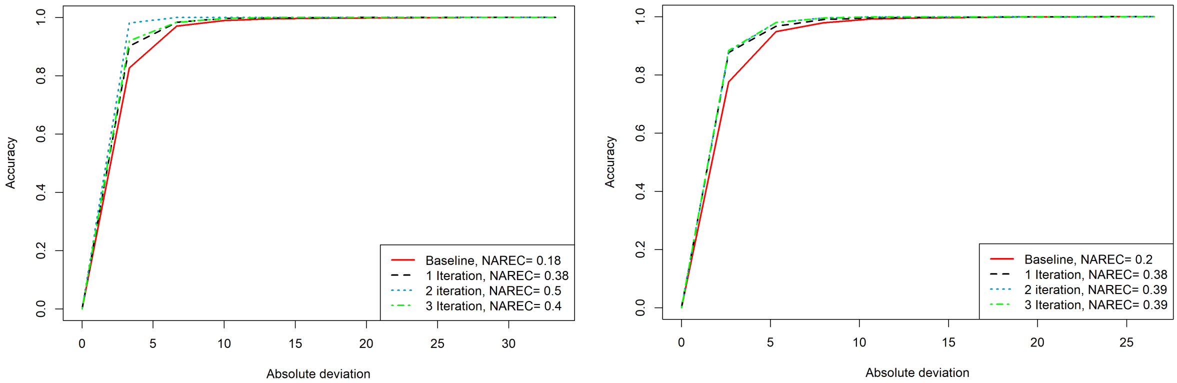

Figure 16: REC curves for the warp (left) and weft (right) tear strength prediction models.

tear weft and second CRISP-DM iteration model for tear warp. The quality of the best model predictions

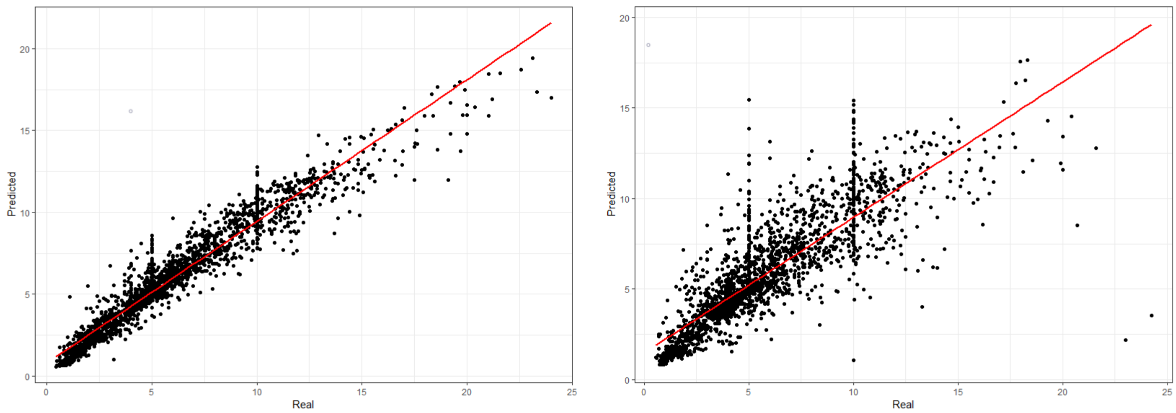

Figure 17: Regression scatter plot of best models for the warp (left) and weft (right) tear strength predictions.

can be visualized in the left of Figure 17. The regression scatter plots show that the predictions are closer to the real values when compared with the scatter plots of the first CRISP-DM iteration (Figure 17). A high quality regression was achieved for the warp tear strength prediction (left of Figure 17). These regression results were then shown to the textile company experts, which provided very positive feedback and considered them as valuable. In effect, the best predictive models are already incorporated into a prototype tool, which includes a friendly dashboard that will be soon integrated with the textile production information system.

3.3. Prediction of Nine Laboratory Quality Tests[2]

3.3.1 *Introduction*

In this second set of experiments, we implement a CRISP-DM project for the prediction of the final fabric physical properties, as measured by nine laboratory quality tests (e.g., abrasion, pilling). The goal is to use a ML model as an "oracle", providing estimates of the fabric real physical properties for several input design options, thus aiding the textile design experts and reducing the number of fabric production attempts. To better focus on input feature selection and transformation, an AutoML procedure was utilized during the modeling stage of CRISP-DM, allowing to automatically select and tune the hyperparameters of the predictive ML models. In particular, we focus on input variables that can be set during the textile design phase, namely based on fabric design (e.g., composition, amount of finished threads) and finishing (e.g., washing, drying, singeing) features. In total, we executed two major CRISP-DM iterations, in which we explored different input feature engineering strategies.

3.3.2 *First CRISP-DM Iteration*

Business Understanding

The creation of a new woven fabric starts with the definition of desired characteristics. The fabric developer uses their experience and intuition, taking into account the textile requirements and starts to analyze the most similar fabrics already produced. Then, several design elements are initially set, such as the type and number of fibers and the pick count. Some of these design elements involve a single value per fabric (e.g., number of picks), while others involve a variable number of choices (e.g., which and how many yarns to use). Next, a physical sample is produced using several materials (e.g., yarns) and machines (e.g., loom). The final production stage includes a variable sequence of finishing operations (e.g., washing, drying). Then, the produced fabric is tested via laboratory instruments, allowing to infer the physical properties and check if it meets the desired characteristics. If the fabric

[2]This section integrates the content of the research paper "Predicting Physical Properties of Woven Fabrics via Automated Machine Learning and Textile Design and Finishing Features", published in the Artificial Intelligence Applications and Innovations - 16th IFIP WG 12.5 International Conference, (AIAI 2020) (Ribeiro et al., 2020). **Acknowledgments** This work was carried out within the project "TexBoost: less Commodities more Specialities" reference POCI-01-0247-FEDER-024523, co-funded by *Fundo Europeu de Desenvolvimento Regional* (FEDER), through Portugal 2020 (P2020).

does not comply with the quality standards or client requirements, then the whole fabric creation process is repeated. In practice, several iterations are executed until a quality fabric is achieved, which results in additional production time and costs. The analyzed Textile company expressed the need to get a fast and cheap estimate of the true fabric physical properties by adopting a ML approach. The goal is to use the predictive ML models as "oracles", quickly checking some fabric design and finishing alternative choices, thus reducing the number of attempts necessary to produce a woven fabric.

In total, the company identified nine target properties: abrasion, seam slippage (warp and weft directions), elasticity (warp and weft directions), pilling, dimensional stability to steam (warp and weft directions) and bias distortion. All these nine properties are measured using numeric values. In this work, each property is measured as a separate regression task.

Data Understanding and Preparation

The data was collected from two main data sources: the company ERP, with fabric production records, and the laboratory testing database, with fabric quality tests performed between February 2012 to March 2019. The ERP data included 90,034 examples with 2,391 features per row. Using a manual analysis and domain expert knowledge, the ERP features were filtered into a total of 805 potentially relevant attributes. The laboratory dataset had 149,388 examples with the results for the nine selected physical tests. To aggregate all data, a Data Warehouse system was implemented, using an ETL process to merge and preprocess the two data sources. During the ETL process, some records were discarded since they had missing features (e.g., no yarns or no composition values).

When analyzing the obtained historical data, we identified that a small fraction of laboratory database entries (around 1%) included slightly different physical test values for the same fabric. After consulting the laboratory analysts, it become clear that the differences were due to the execution of laboratory tests at different fabric finishing procedures (e.g., before or after drying). Since the laboratory database did not include when such tests were executed, we opted to compute average values, in order to get a single number per fabric and test.

The initial set of input features explored in this CRISP-DM iteration is presented in Table 4. Figure 18 exemplifies how some of these features are related with the textile fabric.

The first 11 rows of the table are related with a fixed set of design attributes that are defined for all fabrics. Each fabric is composed of two main elements warp and weft, each including a variable mixture of yarns, from 1 (minimum) to a maximum of 21 (in our database). Moreover, each yarn has four main

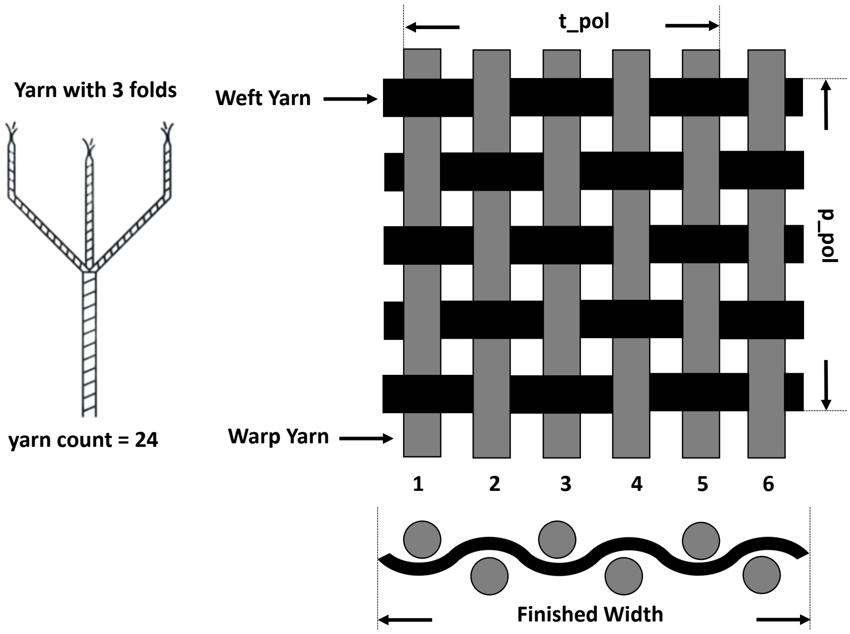

Figure 18: Visualization of some woven fabric features.

characterizing features plus the number of its repetitions in the warp or weft (these features are shown in the last five rows of Table 4). Thus, the proper preprocessing of yarn data, to feed the regression models, is a non-trivial issue. In this work, we propose the following yarn representation. For each fabric, we use a sequence that has a maximum of max_y yarns for warp and then another sequence of max_y yarns for weft. Each sequence is thus composed by the elements $< y_1, ..., y_{max_y} >$, where y_i denotes the i-th yarn representation data. In this work, we adopt the threshold of max_y=6 yarns per warp and weft. This value allows the representation of 99.7% of the fabrics without any information loss, while using a larger threshold would increase the sparseness of the input space, increasing the complexity of the predictive models. When a fabric does not have 6 yarns, we use a zero padding to fill the "empty" yarn values, which is a popular text preprocessing technique that adds null values (e.g., 0) to non-existent features. Finally, we explore two yarn alternative representations: **A**, use of the code (unique value) and its number of repetitions, where y_i is set as the tuple $(\text{code}_i, \text{repetitions}_i)$; and **B**, use of all yarn characterizing elements except the code, where y_i is set as $(\text{composition}_i, \text{folds}_i, \text{count}_i, \text{repetitions}_i)$. In total, the **A** representation assumes 35 input variables (11+2×6×2), while the **B** encoding results in 59 input features (11+2×6×5).

Before feeding the ML algorithms, all the numeric inputs were standardized to a zero mean and one standard deviation. As for the nominal variables, several of them contain a high cardinality. For

Table 4: Initial list of fabric design input features.

Attribute	Description (data type)	Range
composition	Composition of the fabric (nominal code)	660 levels
t_pol	Number of finished threads per centimeter (numeric)	[18,1321]
p_pol	Number of finished picks per centimeter (numeric)	[7,510]
weight/m^2	Weight (in grams) per square meter (numeric)	[22,1690]
finished width	Width in centimeters (numeric)	[122,168]
weave design	Weave pattern of the fabric (nominal code)	20 levels
reed width	Width of the reed in centimeters (numeric)	[30,242]
denting	Number of the reed dents per centimeter (numeric)	[5,252]
ends/dent	Number of yarns per dent (numeric)	{0,1,...,8}
n_picks	Number of picks on loom per centimeter (numeric)	[0,3450]
warp total ends	Total number of threads on the warp (numeric)	[1026,6862]
yarn code	Identification code of the yarn (nominal code)	6,265 levels
yarn composition	Composition of the yarn (nominal code)	88 levels
n_folds	Number of single yarns twisted (numeric)	{0,1,...,12}
yarn count	Mass per unit length of the yarn (numeric)	[0,268]
yarn repetitions	Number of yarn repetitions in warp or weft (numeric)	{0,1,...,8}

instance, the analyzed database includes 6,265 distinct types of yarns. A popular nominal to numeric transform is the one-hot encoding, which assigns one boolean value per nominal level. However, this transformation would highly increase the input space, resulting in a very sparse representation that would prejudice the learning of the regression models, also enlarging the computational memory and effort. To handle this issue, in this work we transform all nominal attributes with the *Inverse Document Frequency (IDF)* function:

$$IDF(l) = ln(n/n_l) \quad (4)$$

where n is the total number of examples in the training set and n_l is the number of examples that contain the level l in the analyzed attribute (Campos et al., 2016). The advantage of this transform is that is encodes a nominal attribute into a single numeric value, with the most frequent levels being set near the zero (but with a larger “space” between them), and the less frequent ones being more close to each other and near a $IDF(l)$ maximum value.

Table 5 presents the nine output targets. The last column (**Range**) shows the admissible range values for each target, as defined by the textile company. All examples outside such range were considered outliers (e.g., uncommon military fabrics) and thus removed from the dataset. Since different quality tests can be assigned to different fabrics (depending on the client requirements), a variable number of examples is presented for each output (column **Examples**).

Table 5: List of output target variables.

Test	**Examples**	**Range**
Abrasion	456	[5000,30000]
Seam Slippage (warp)	10,605	[1,20]
Seam Slippage (weft)	10,279	[1,20]
Elasticity (warp)	7,901	[5,55]
Elasticity (weft)	12,698	[5,70]
Dimensional Stability to Steam (warp)	8,773	[-4,2]
Dimensional Stability to Steam (weft)	8,871	[-4,2]
Bias Distortion	15,141	[1,14]
Pilling	11,912	[1,4.5]

Modeling

The experiments were conducted in a personal computer using two different computing environments: the R statistical tool and its `rminer` package for data manipulation and evaluation of ML algorithms (Cortez, 2021), and `H2O` software which implements a AutoML procedure (Cook, 2016). As previously discussed, during the first iteration of CRISP-DM we explored the issue of yarn representation, thus two main strategies as compared: **A** and **B**. During this modeling stage, to find the best ML algorithm we adopt an AutoML procedure.

The AutoML was configured to automatically select the regression model and its hyperparameters based on the best MAE, using a internal 5-fold cross-validation applied over the training data. We adopted the same AutoML configuration executed in (Ferreira et al., 2020). The computational experiments were executed on a desktop computer and each ML algorithm was trained using a maximum running time of 3,600 seconds. After selecting the best ML algorithm, its best set of hyperparameters are fixed and the ML algorithm is retrained with all training data. A total of 6 different regression families are searched by the AutoML tool: *Random Forest (RF)*, *Extremely Randomized Trees (XRT)*, GLM, GBM, *XGBoost (XG)* and *Stacked Ensemble (SE)*. RF is an ensemble method that typically combines a large set of tree predictors, such that each tree depends on a random sample of features and training examples (Breiman, 2001). XRT is another tree ensemble that consists of randomizing both attribute and cut-point choices when splitting a tree node (Geurts et al., 2006). GLM estimates regression models for outcomes following exponential distributions (e.g., Gaussian, Poisson, gamma) (Nelder and Wedderburn, 1972). GBM performs an ensemble of weak successive decision trees, sequentially building regression trees for all data features (Natekin and Knoll, 2013). XG is another popular boosting decision tree algorithm (Chen and Guestrin, 2016). Finally, the SE combines the predictions of the previous individual ML algorithms by using a second-level ML algorithm (Breiman, 1996). The `H2O` tool sets RF and XRT with their default hyperparameters, performs a grid search to set the hyperparameters for GLM (1 hyperparameter), GBM (9 hyperparameters) and XG (10 hyperparameters), and uses GLM as the second-level learner for SE.

Evaluation

An external 3-fold cross-validation was executed to evaluate the regression models. Several measures were selected to measure the quality of the predictions: MAE, *Normalized Mean Absolute Error (NMAE)*, Adjusted R2 (Adj.R^2) and classification accuracy for a given tolerance T (Acc@T). Regarding MAE and NMAE, the lower the values, the better are the predictions. The NMAE measure normalizes the NMAE by the range of the output target on the test set, thus it provides a percentage that is easy to interpret and is scale independent. In the case of Adj.R^2 and Acc@T (from 0 to 1), higher values indicate better predictions. The Acc@T value is based on the REC curves and it measures the percentage of correctly classified examples when assuming a fixed absolute error tolerance (T) (Bi and Bennett, 2003). In this work we use $T \in \{5\%, 10\%, 20\%\}$. We note that the percentage of error tolerance is computed by considering the range of the target values. The first CRISP-DM iteration results are discussed in Section 3.3.4.

3.3.3 *Second CRISP-DM Iteration*

After showing the Section 3.3.4 results to the textile experts, it was decided to perform a second CRISP-DM iteration to check the utility of finishing features. During a new business understanding phase, it become clear that the finishing process should influence the final fabric properties. The finishing consists of a predefined sequence of operations that are applied to a fabric with the goal to increase the attractiveness or serviceability of the textile product (Hall, 2000).

In a new data understanding and preparation stages, we identified that the company had a total of 61 different types of finishing operations. Moreover, the sequence of finishing operations can be different for each fabric and it can include repetitions of the operations (e.g., several wash and dry cycles). In the analyzed database, the number of executed finishing operations ranged from 1 (minimum) to 39 (maximum), with an average of 6.82. Table 6 presents the top ten most used types of finishing and the respective number of usages (column **Examples**). In the table, we added the special value "Others" to represent a merge of distinct finishing operations for which there was no description data.

Table 6: Ten most used fabric finishing operations.

Rank	**Finishing**	**Examples**	**Rank**	**Finishing**	**Examples**
1	Dry	145,008	6	Finish_Fixate	53,717
2	Wash	122,513	7	Dyeing	44,162
3	Sanforization	73,401	8	Others	31,704
4	Finish	67,774	9	Shear_Right	21,600
5	Singeing	58,875	10	Decatizing	20,685

Similarly to the yarn encoding strategy, in this work, we will assume a sequence with a maximum of max_f finishing operations to represent the finishing process: $< f_1, ..., f_{max_f} >$, where f_i denotes the i-th finishing operation. In this work, we set max_f=10 as a reasonable value that represents around 85% of all fabrics without information loss, helping to reduce the number of inputs that are fed into the ML models. To encode each finishing operation (nominal attribute) we adopt the same IDF transform (Equation (4)). In the modeling phase, the best previous input encoding (**A**) is compared with the new encoding **C** that merges all **A** inputs with the finishing features, resulting in 45 (35+10) input variables. The evaluation phase was executed similarly to the first CRISP-DM iteration.

3.3.4 *Results*

In all experiments performed the AutoML always selected the GBM or SE algorithms. GBM provided the best overall results (lowest NMAE averaged over the external 3 cross-validation iterations), while for some targets and specific folds (e.g., Bias Distortion and third fold experiment), the selected model was SE. To compare the feature strategy results, we always assume the best algorithm (GBM or SE) per external fold validation, denoting this as the AutoML model.

Table 7 summarizes the predictive performance results, in terms of the 3-fold average NMAE values for the best AutoML model that were obtained during the first and second CRISP-DM iterations.

Table 7: AutoML predictive results (average NMAE test set values in %; best results per CRISP-DM iteration in **bold)**.

	First Iteration		**Second Iteration**	
Target	**A**	**B**	**A**	**C**
Abrasion	**4.81**	5.17	**4.81**	4.93
Seam Slippage (warp)	**4.54**	5.00	4.54	**4.43**
Seam Slippage (weft)	3.09	**2.62**	3.09	**2.56**
Elasticity (warp)	**2.94**	3.42	2.94	**2.59**
Elasticity (weft)	**2.39**	2.87	2.39	**2.15**
Dimensional Stability to Steam (warp)	**6.60**	7.52	6.60	**6.17**
Dimensional Stability to Steam (weft)	**4.27**	6.58	4.27	**4.12**
Bias Distortion	**4.16**	4.33	4.16	**3.79**
Pilling	**6.80**	8.14	6.80	**6.70**
Average	**4.41**	5.07	4.41	**4.16**

For the first iteration, it becomes clear that **A** is the best yarn representation strategy. It provides the lowest NMAE results for eight of the nine fabric targets and it also obtains the lowest average value over all output tasks (difference of 0.66 percentage points when compared with **B**). Moreover, **A** has the additional advantage of producing fewer inputs (35 and not 59), leading to predictive models that require less computational memory and fitting effort. Following these results, we adopted the **A** encoding to represent the yarns. In the second iteration, the usage of fabric finishing features (**C**) improves the prediction results for eight of the nine targets. Overall, **C** provides the lowest average NMAE, with a 0.25 percentage point improvement when compared with **A**.

Table 8 complements the results by showing the other predictive measures for **A** and **C** (represented in column **Str.**). In general, when **C** obtains the lowest MAE error, it also outperforms the **A** strategy for the other measures (Acc@T and Adj. R^2).

For demonstrative purposes, Figure 19 shows the AutoML elasticity (warp) predictions (x-axis) versus the target values for a particular external 3-fold iteration. The plot includes the tolerance ranges of the $T = 5\%$ and $T = 10\%$, showing that an interesting percentage of the values are correctly predicted within those ranges (e.g., 43% of accuracy for $T = 10\%$).

Table 8: AutoML predictive results (other regression measures; best results in **bold)**.

Test	Str.	MAE	Acc@5%	Acc@10%	Acc@20%	Adj. R^2
		Regression Measures				
Abrasion	A	**924.03**	**1%**	**1%**	**2%**	**0.76**
	C	948.60	0%	0%	**1%**	0.75
Seam Slippage (warp)	A	0.83	24%	35%	47%	0.79
	C	**0.81**	**28%**	**42%**	**54%**	**0.80**
Seam Slippage (weft)	A	0.55	28%	52%	65%	0.83
	C	**0.46**	**61%**	**67%**	**73%**	**0.84**
Elasticity (warp)	A	1.46	11%	17%	24%	**0.92**
	C	**1.29**	**26%**	**34%**	**42%**	**0.92**
Elasticity (weft)	A	1.25	10%	17%	25%	**0.92**
	C	**1.12**	**13%**	**20%**	**30%**	**0.92**
Dimensional Stability to Steam (warp)	A	0.40	17%	31%	50%	0.63
	C	**0.37**	**30%**	**40%**	**53%**	**0.65**
Dimensional Stability to Steam (weft)	A	**0.25**	**55%**	61%	67%	0.73
	C	**0.25**	**55%**	**62%**	**68%**	**0.75**
Bias Distortion	A	0.50	13%	25%	40%	0.53
	C	**0.46**	**14%**	**26%**	**44%**	**0.59**
Pilling	A	0.24	**35%**	**48%**	**63%**	0.76
	C	**0.23**	25%	37%	54%	**0.78**

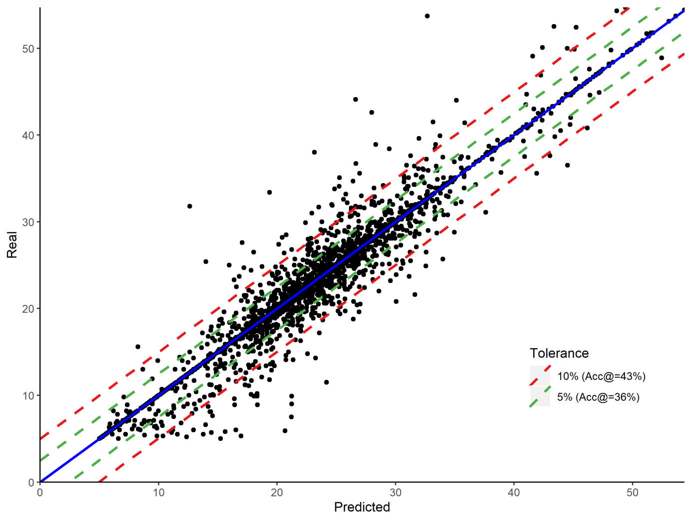

Figure 19: Predicted versus real elasticity (warp) values.

3.4. Intelligent Decision Support System for Production Planning[3]

3.4.1 *Introduction*

Currently, there is pressure in industries to increase efficiency (e.g., reduce operating costs and time) in order to compete in their markets. One way is to adopt an IDSS, which incorporates *Artificial Intelligence (AI)* techniques to provide actionable knowledge from raw data. In this work, we assume an IDSS for the Garments industry and that is based in the concept of ABI, which combines ML, to predict relevant decision context variables, with *Modern Optimization (MO)*, to search for the best decision choices (according to one or more objectives). In this work, we follow such innovative ABI combination by using an AutoML to first predict four important garment subcontractor decision variables. Then, we adopt historical data and one of the predicted variables (production time) to feed an EMO that searches for the best subcontractor allocation plan, simultaneously minimizing the total allocation cost and time.

3.4.2 *Materials and Methods*

Garment Data

The data was provided by INFOS, which is a Portuguese software company that works with several Textile industry clients. The company developed an ERP that supports the production of garments. The goal of this research is to develop an IDSS based on the ABI concept and that will be integrated into the INFOS ERP system, allowing it to automatically design garment subcontractor plans regardless of the size of the company and the complexity of the production order. The subcontractor selection is a nontrivial task, since is a large range of textile operations, each involving costs and delivery dates. We collected all company garment related records, including purchase and manufacturing orders, from 2016 to 2020. The data was then divided into three major groups: purchase of raw material, manufacturing and subcontractor. Next, we implemented an ETL process to select and clean the data (e.g., removal

[3] This section integrates the content of the research paper "An Intelligent Decision Support System for Production Planning in Garments Industry.", published in the International Conference on Intelligent Data Engineering and Automated Learning, (IDEAL 2021) (Ribeiro et al., 2021). **Acknowledgments** This work was carried out within the project "Connect@Fashion" reference POCI-01-0247-FEDER-045296, co-funded by *Fundo Europeu de Desenvolvimento Regional* (FEDER), through Portugal 2020 (P2020).

of missing features and records with wrong dates). All data processing procedures (including the ABI system) were implemented in the Python language by the authors.

Table 9 describes the input features (**Attribute**), their description (**Description**), data **Type**, number of **Levels** and **Domain** values separated by objective (four predictive targets and one optimization task). The final set of input features was obtained after several iterations of predictive task executions. The datasets for the predictive (regression) tasks include: Lead Time – 3,315 records; Production Time – 25,449 examples; Production Waste – 24,425 instances; and Delivery Delays – 6,016 records. Finally, the optimization objective (Production Plan) contains 5,500 records related with subcontractors.

Table 9: Description of the input features by objective.

Objective	Attribute	Description	Type	Levels	Domain
Lead Time	Supp_cod	Supplier identification	Integer	102	[14, 2265]
	Date_purch	Date of purchase order	Date	873	-
	Rmat_cod	Raw material code	String	188	-
	Qty	Quantity to buy	Integer	876	[1, 11650]
Production Time	Subc_cod	Subcontractor identification	Integer	275	[0, 9999]
	Mat_cod	Final product code	String	846	-
	Oper_desc	Textile operation	String	93	-
	Qty	Quantity to produce	Float	9356	[1, 97512]
Production Waste	Mat_cod	Final product code	String	864	-
	Qty	Quantity to produce	Float	4646	[1, 13448.3]
	Subc_cod	Subcontractor identification	Integer	41	[8, 9996]
	Rmat_cod	Raw material code	String	1386	-
Delivery Delays	Plan_endate	Planned date to end production	Date	940	-
	Mat_cod	Code of the final product	String	205	-
	Qty	Quantity to produce	Integer	3156	[3, 75838]
Production Plan	Oper_desc	Textile operation	String	94	-
	Avgp_cost	Average cost of textile operation	Float	47	[0.01, 1.75]
	Price	Cost of textile operation for given product per subcontractor	Float	247	[0.01, 11.60]
	Subc_cod	Subcontractor identification	Integer	293	[24, 2254]
	Mat_cod	Code of the final product	String	169	-
	Capacity	Subcontractor production capacity by textile operation	Integer	14	[100, 3000]

Regarding the target output target variables for the predictive tasks, we detected that the company does not have records of them, being necessary to calculate them: *Ldtime* was obtained by subtracting the receiving date of a order from the placement order date and if resulting value was negative that row was discarded; for *Prod_days* we created a function that subtracts the production finish date from the planned production start date and outputs the number of working days between the two dates and the if the number of days was negative that row was discarded; in the case of *Waste_ratio*, we first subtracted the produced quantity from the quantity to produce and if the resulting value was positive it was changed to zero, afterwards we divided the absolute result by the quantity to produce, multiplying the final result by 100; finally for *Delay_days* we created a function that subtracts the scheduled delivery date finish date from the delivery date and outputs the number of working days between the two dates and the if the number of days was negative it was changed to zero.

Table 10: Description of the output target variables.

Target	Description	Type	Levels	Domain
Ldtime	Days between delivery date and purchase date	Integer	94	[0, 59]
Prod_days	Working days to produce a certain quantity	Integer	47	[1, 63]
Waste_ratio	Percentage of wasted material	Integer	247	[0, 100]
Delay_days	Working days between scheduled delivery date and delivery date	Integer	293	[0, 64]

Table 10 describes the four output target variables with their description (**Description**), data type (**Type**), number of levels (**Levels**) and domain values (**Domain**).

In terms of preprocessing, since the String variables had a high cardinally, we employed a Label Encoder, in order to transform each level into a distinct numeric value. This option provided better results when compared with the known One-Hot encoding, which created a very high number of input features. As for the Date features, we adopted the proleptic Gregorian ordinal of a date, allowing to provide a simpler numeric value. Then, all numeric inputs were normalized by using a z-score standardization.

Intelligent Decision Support System

The proposed IDSS contains three main modules (Figure 20): data extraction and processing, prediction and optimization. The first module is responsible for receiving the garment data, selecting the features for each objective and then creating the necessary input for prediction. The prediction module receives the data separated by predictive task splitting it into training and test sets (data separation, according to the adopted cross-validation method). Then, it trains the predictive models (model training), evaluating the models performance (model evaluation), selecting and storing the best prediction model (model selection). Then, the user inserts the data related to the lead time, using the respective model to predict the number of days that will take to receive the raw materials and can define a starting and end date for production. Finally, the optimization module receives the subcontractors data (Table 9), filtered by the product to manufacture and the textile operations to execute, the quantity to produce and maximum allowed dates (all provided by the user). Then, the MO algorithm uses this data and also one of the predicted indicators (production time) to search for the best subcontractor quantity allocation, aiming to reduce the total costs and time.

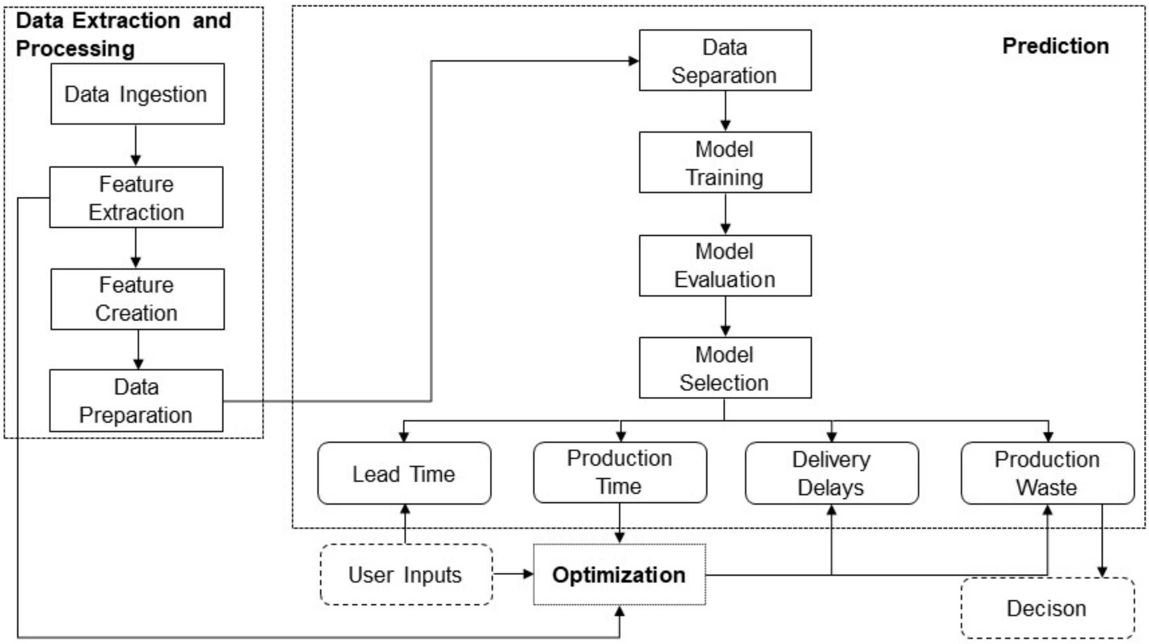

Figure 20: Flow diagram describing the behavior of the IDSS.

To reduce the modeling effort during the development of the prediction module, we adopted the `H2O` AutoML tool that provided good results in recent AutoML benchmark study (Ferreira et al., 2021). The AutoML was configured to automatically select the best regression model and its hyperparameters based on the best MAE, using an internal 10-fold cross-validation applied over the training data. Five different ML algorithms were searched by the tool: RF, XRT, GLM, GBM and two Stacked Ensembles, one with the best model of each family and other with all trained models. An external 10-fold cross-validation was executed to evaluate the ML models and the quality of the regression was accessed by using the MAE and NMAE measures. The lower the values, the better are the predictions. The NMAE measure normalizes the MAE by the range of the output target on the test set, thus it provides a percentage that is easy to interpret and that is scale independent.

A production order can be defined as a composition of tasks that are executed sequentially. Each task can be represented by a set of candidate subcontractors offering similar services, where each service can have a different value in price and quantity per subcontractor. The subcontractor allocation is defined as a multi-objective task (i.e., reduce both cost and time), thus we employ a Pareto approach via an EMO algorithm, namely *Non-dominated Sorting Genetic Algorithm II (NSGA-II)* (Cortez, 2021), as implemented in the `pymoo` Python module (Blank and Deb, 2020). NSGA-II is a multi-objective optimization algorithm with three distinctive features: a fast non-dominated sorting approach, fast crowded distance estimation procedure and usage of a simple crowded comparison operator (Deb et al., 2002). When compared with other hypervolume based algorithms (e.g, SMS-EMOA), the NSGA-II algorithm tends to obtain competitive results when only two or three objectives are optimized (Chiandussi et al., 2012). The algorithm returns a population of non-dominated solutions, each representing a different subcontractor allocation and that is associated with a distinct cost-time trade-off. The full subcontractor optimization can be defined in terms of *x* textile sequential operations that need to be executed. For each operation, there are *y* candidates (subcontractors) with different price and capacity parameters. Each solution is naturally represented as a sequence of q_i integer values ($0 \leq q_i \leq q_{max}$), denoting the quantity assigned for each subcontractor i, where q_{max} denotes the total required quantity for operation x, and $i \in \{1, ..., M\}$ and M represent the number of available subcontractors for operation x. We repair solutions by ignoring any excess of subcontractor allocation (the first allocated subcontractor is served first) or by randomly distributing the deficit allocation to any of the available subcontractors. Each solution is evaluated in terms of total production plan cost and allocation time. To compute these two goals, the EMO algorithm uses the production time prediction (as shown in Figure 20). Once the Pareto curve is optimized and for user selected trade-offs, we then compute the prediction indicators of the remaining targets (e.g., production waste), such that the user can further inspect the quality of the obtained solutions. In order to obtain a single measure per Pareto curve, we selected the Hypervolume (HV) measure, which represents the volume of the objective space when assuming a “worst” reference point (Campos Ciro et al., 2016). The higher the HV value, the better is the Pareto curve optimization.

3.4.3 *Experiments and Results*

The average of the external 10-fold iteration predictive results (in terms of **MAE** and **NMAE**) are presented in Table 11. The table also presents the best ML **Model**. In general, low regression errors were achieved, with the NMAE values ranging from 3.6% to 9.2%. We particularly note that the best NMAE values were obtained for the target that is directly used by the NSGA-II MO (*Prod_days* produces an average NMAE error of just 3.6%). The selected ML algorithm was a stacked ensemble for three of the targets, while the GBM obtained the best results for the production waste prediction.

Table 11: AutoML predictive results for each predicted target.

Target	**Model**	**MAE**	**NMAE**
Ldtime	Stacked Ensemble (All Models)	3.31	9.20%
Prod_days	Stacked Ensemble (All Models)	1.60	3.63%
Waste_ratio	Gradient Boosting Machine	4.24	4.24%
Delay_days	Stacked Ensemble (Best of each family)	3.57	5.71%

For the optimization experiments, we analyzed a production order of 10,000 units of a product that requires three textile operations (cutting, tailoring and packaging) using one raw material. Using historical data, we then selected all the subcontractors that could execute these operations along with the respective cost and production capacity to create a subcontract allocation case study to utilize in the experiments. In total, the case study includes 26 subcontractors (which corresponds to the number of searched integers by the NSGA-II algorithm): cutting - 4 candidates, tailoring – 8 candidates and packaging – 14 candidates (4+8+14=26). To compute the cost and time associated with each solution, we use four attributes from Table 9 (*Subc_cod, Capacity, Price* and *Oper_desc*) and also the predicted *Prod_days* variable (see Table 10). We assumed some reasonable assumptions (defined by the INFOS company): one subcontractor cannot execute two or more tasks simultaneously, the subcontractor is always available and there is no shortage of raw materials.

The two objective functions that need to be minimized are the Total Cost (TC) and Total Production Time (TPT). The TC function is the sum of the multiplication of the assigned quantity to an individual by the price of operation for that individual operation. As for TPT, the function is the sum of the maximum days required by each sequential operation (cutting, tailoring and packaging). Since subcontractors can work simultaneously in the same operation (e.g., cutting), we consider the slowest operator time (measured in terms of the number of days). Solutions that split the q_i quantities by different operators for an operation will thus contribute for a lower TPT value. The lower bound is always zero and the upper bound was set to the quantity to be produced. When needed, a repair procedure is used to convert an unfeasible solution to a feasible one, see Section 3.4.2.

The NSGA-II algorithm was configured with a check procedure that eliminates duplicates, making sure that the mating produces offspring that are different from themselves and the existing population regarding their design space values. A grid search was used to set the NSGA-II hyperparameters (e.g., the population size was ranged within {50,100,150,...,500}), assuming the *Hypervolume (HV)* measure as the selection criterion and a reference point of (30 days, 20,000 EUR). The best obtained values correspond to a normalized HV (when each objective is divided by the respective reference point value) of 0.71, which requires 157 seconds of execution time on an Intel Xeon processor. The selected NSGA-II setup includes: population size of 100, two-point crossover with 90%, polynomial mutation probability of 20% and a total of 200 generations.

The left of Figure 21 shows the Pareto front obtained after 200 generations when considering our case study. The Pareto front contains 100 solutions, with the TPT ranging from 12 to 30 working days and TC ranging from 18,000 to 20,000 EUR. The right of Figure 21 shows the evolution of the NSGA-II algorithm, in terms of the full HV measure (y−axis) through the executed 200 generations.

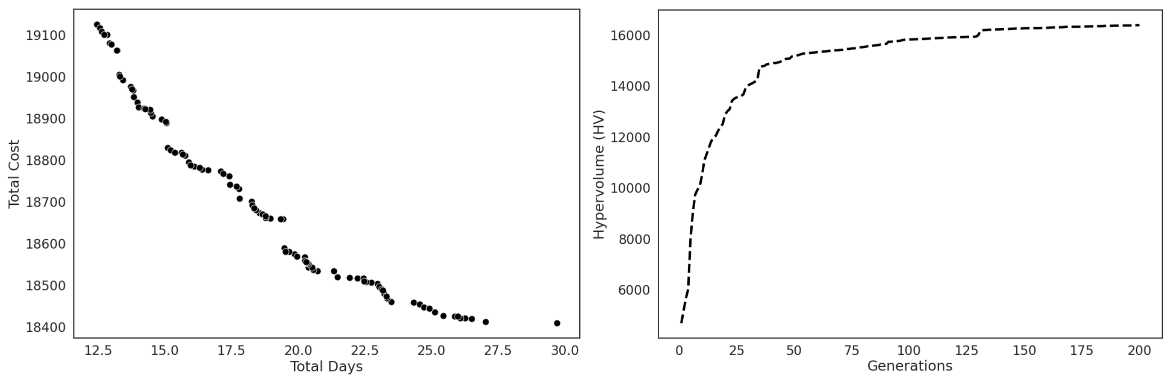

Figure 21: Optimized Pareto front (left) and NSGA-II HV generation evolution (right).

The graph shows a substantial improvement that is obtained by NSGA-II. In effect, in the first generation, the HV measure is 4,700 (normalized value of 0.2). After 200 generations, the value increased to 16,391 (normalized value of 0.71), which corresponds to an improvement of 51 percentage points when considering the normalized HV scale. The results were shown to the INFOS company, which provided very positive feedback. In particular, the obtained TPT and TC ranges were considered realistic. Moreover, the company signaled that the obtained Pareto front provides a more richer set of trade-off solutions, while also being faster to compute when compared with the currently adopted manual subcontractor allocation.

3.5. Conclusions

The Fourth Industrial revolution presented a new type of solutions to the Textile industry, by introducing customized and flexible mass-production technologies that collect and store high volumes of data. These Information Technologies (IT) solutions represent a great opportunity to increase the competitiveness of the Textile industry, where companies must find a way to differentiate themselves from their competitors.

In this sector, a differentiating aspect is an ability to address the needs of a market in constant evolution due to fashion trends. This creates a need to constantly design new fabrics within a relatively small time frame, in order to meet the needs of the market. To design a new fabric is necessary to first define a prototype design, with several construction components that will affect the physical properties and aesthetics of the final product. This a time-consuming and expensive that involves several trial-and-error fabric prototype production cycles until the desired properties and aesthetics are obtained.

In this chapter, we address this challenging task by showing the initial research attempts that were performed, aiming to obtain the desired predictive and prescriptive analytics. For the first predictive experiments, we collected data from a Portuguese textile company which included an initial database of thousands of fabric records from 2012 to 2019 that was merged with the laboratory test data. The experimentation assumed a DM approach guided by the CRISP-DM methodology, aiming to predict the results of fabric tear strength tests in warp and weft directions. A total of three CRISP-DM iterations were executed, aiming to explore distinct data preprocessing operations (e.g., outlier removal, the inclusion of additional inputs) for these tasks, adopting an AutoML tool during the Modeling stage of CRISP-DM. Interesting results were achieved at the second CRISP-DM iteration for the warp tear strength, which involved outlier removal while the best weft test results were obtained at the third CRISP-DM iteration, which included the final fabric composition as an extra input feature.

In the second predictive study, we implemented two iterations of the CRISP-DM) methodology. Each iteration focused on a feature engineering task, aiming to check the value of input fabric yarn and finishing feature representations. Similar to the first work, we used an AutoML tool during the modeling stage of CRISP-DM, to select the best among six state-of-the-art ML algorithms. The best results were achieved by an input set of features that includes a fixed sequence with a simple yarn code representation and another fixed sequence with fabric finishing operations. Overall, interesting predictive results were achieved for nine targeted fabric properties, with an average NMAE error that ranges from 2% to 7%.

When comparing these initial predictive works, although the results for the full 15 tests are not reflected in this document, it became clear that the second work presented the best approach, providing lower NMAE values for all tests. Moreover, the approach proposed in the second work has the additional advantage of producing fewer inputs (45 and not 59), leading to predictive models that require less computational memory and fitting effort. Also, the experiments depicted in the second work demonstrated that the usage of fabric finishing features improves the prediction results for most of the tests. Following the results obtained in the second predictive work, we adopted the simple yarn encoding for yarn representation and included the fixed sequence of fabric finishing operations for future studies. The results obtained in this work were shown to the textile company, which considered them valuable to reduce the number of fabric creation attempts, thus having the potential to save production time and costs. In future work, we intend to apply a similar approach in the prediction of other fabric quality tests, such as residual extension and traction.

Finally, in the last work presented in this chapter, we detailed our first research attempt regarding the design and implementation of an IDSS for the Textile industry. In this work, we propose an IDSS that creates a textile production plan to allocate subcontractors. The IDSS is based on the ABI concept that combines predictive (via ML) with prescriptive (via MO) analytics in order to provide actionable knowledge from raw data. The IDSS was designed to work with real-world data from a Portuguese software company (INFOS) that works with diverse textile clients. Firstly, an AutoML tool was adopted to automatically select the best ML model among five algorithms when targeting four relevant allocation decision context variables. Interesting results were achieved by the prediction models (error that ranges from 3.6% to 9.2%). Then, we designed a MO model that uses one of the predicted variables (production time) and historical data to automatically allocate subcontractors to execute sequential operations associated with a textile order. The MO model, based on the NSGA-II algorithm, assumes a Pareto approach and it was designed to simultaneously minimize the cost and time to execute the order. To demonstrate the MO, we selected a case study that includes four operations and 26 potential textile subcontractors. The obtained results were shown to the INFOS company, which considered them very positive. In future work, we intend to augment the IDSS by incorporating more problem-domain constraints, such as incorporating updated data about the current availability of subcontractors. Furthermore, we wish to deploy the designed IDSS into the INFOS ERP system, in order to get more valuable feedback from real environment usage.

4

PROPOSED INTELLIGENT DECISION SUPPORT SYSTEM[1]

4.1. Research Context

After obtaining the results regarding the three initial research works detailed in Chapter 3, the PhD effort was redirected to its original core research question, involving the design and development of an *Intelligent Decision Support System (IDSS)* that makes use of predictive and prescriptive analytics in order to assist in the process of creating new textile fabrics. While the contents of this chapter can be read as a single work unit (and thus it was submitted to an international scientific journal), several of its elements benefit from the previous works. In particular, the proposed IDSS fabric physical property prediction models assume all inputs proposed in Section 3.3 except for the textile composition values, which are now considered a new regression target. Thus, the proposed IDSS assumes two predictive goals (physical property and textile composition). Moreover, the adopted prescriptive models use the same *Modern Optimization (MO)* optimization approach that was explored in the third work presented in Chapter 3, namely the usage of the *Non-dominated Sorting Genetic Algorithm II (NSGA-II)* algorithm and analysis of obtained results in terms of the Pareto front and *Hypervolume (HV)* measure.

Nevertheless, it is also relevant to highlight that the proposed IDSS, presented in this chapter, also contains innovative elements, such as the exploration of Multi-Target Regression (MTR) approaches, the prediction of the textile composition and the definition of the two objectives that were adopted for the prescriptive modeling.

[1] This chapter integrates the content of the research paper "A data-driven intelligent decision support system that combines predictive and prescriptive analytics for the design of new textile fabrics", in Neural Computing and Applications, Springer, In Press, ISSN 0941-0643. **Acknowledgments** This work was carried out within the project "TexBoost: less Commodities more Specialities" reference POCI-01-0247-FEDER-024523, co-funded by *Fundo Europeu de Desenvolvimento Regional* (FEDER), through Portugal 2020 (P2020).

4.2. Introduction

In the Textile and Clothing industry, there is a frequent need to design new fabrics in order to meet the fashion market trends. The creation of a new fabric begins with the definition of a prototype design, which includes several construction components that affect the physical properties and aesthetics of the textile product. Initially, the textile designer often uses her/his experience and intuition to select fabrics that have been previously manufactured and that are similar to the desired product specifications. Aiming to reach these specifications, she/he then reshapes the selected fabric by altering one or more of the construction elements (e.g., the type and number of yarns used). Next, the prototype goes into a production stage, in order to check if the desired physical properties are met. If not, then a new prototype design iteration is attempted, in which the designer sets a new fabric configuration that is then produced and tested. Typically, success is reached only after a larger number of fabric prototype developments.

The development of the fabric prototypes involves specialized equipment and personnel, manufacturing lines and several cycles of adjustments before reaching the final prototype (the order to be mass-produced), making it a time-consuming and expensive process. Each time a prototype is produced, laboratory quality tests are required, to assure regulatory compliance of the physical properties (Hu, 2008). In some cases, there are also validation sessions with customers, which might require the production of replicas of the prototype in a larger quantity.

This research is set within a *Research and Development (R&D)* project involving a Portuguese textile company that is being transformed under the Industry 4.0 context. The company expressed the need for an IDSS that could enhance the design of new fabrics, aiming to reduce the number of fabric development attempts, which would highly reduce costs and fabric development time. An IDSS is a decision support system that incorporates *Artificial Intelligence (AI)* methods, such as *Machine Learning (ML)* and metaheuristics, to improve decision making (Arnott and Pervan, 2014). To solve the textile company need, we propose a novel data-driven IDSS that is based on the *Adaptive Business Intelligence (ABI)* (Michalewicz et al., 2006) concept, which assumes a combination of predictive models, based on ML, with prescriptive analytics, based on metaheuristics.

In particular, the proposed IDSS targets two nontrivial predictive goals: first ML goal – the estimation of four relevant fabric physical properties (bias distortion, warp and weft elasticity, and pilling); and second ML goal – the prediction of the final textile composition (e.g., % of cotton and % of polyester).

The physical property ML model can be used as a substitute for the fabric production and laboratory testing, thus reducing time and costs. As for the second ML goal (within our knowledge, first attempted here), it allows to automatically compute the final textile composition from the same set of inputs, which is a relevant information to be shown to the textile clients and customers. Both predictive goals are modeled as regression tasks that depend on several design inputs (e.g., the type of yarns). To solve these goals, historical data, including around 8.6 thousand of fabric creation records, are used to adapt and compare two ML main approaches: STR, based on an *Automated Machine Learning (AutoML)* tool (Ferreira et al., 2021); and Multi-Target Regression (MTR) (Arashloo and Kittler, 2022), based on a deep Multilayer Perceptron *Artificial Neural Network (ANN)* (Goodfellow et al., 2016b).

As for the IDSS prescriptive analytics, they use the first ML goal best prediction models. The aim is to search for the model inputs that match the desired textile properties, thus minimizing the ML predictive error. However, when adopting only this objective, the optimization methods often select inputs that are distant from the learned input space, performing a prediction that is more an extrapolation than an interpolation, thus less reliable. To solve this issue, we also minimize the input vector distance when compared with the ML training dataset. Both objectives are simultaneously optimized by adapting two *Evolutionary Multi-objective Optimization (EMO)* methods (Cortez, 2021): NSGA-II and *Reference Point Based Non-dominated Sorting Genetic Algorithm II (R-NSGA-II)*. The EMO methods are compared by using 100 additional records of designed fabrics (not used for the training and testing of the predictive ML models). The EMO result is a Pareto front of fabric designs, each related with a particular predictive error and input distance trade-off. For the optimized solutions, the IDSS also presents the final textile composition values by using the second ML goal predictions.

4.3. Problem Statement

This work was carried out with the collaboration of a Portuguese textile company that produces high-quality fabrics, based on natural, synthetic, artificial and recycled fibers, focusing in polyester, viscose and elastane blends. The company presents a vertical production system comprised of the following areas: Research and Development, Spinning, Dyeing, Twisting, Weaving and Finishing, with a production capacity of 700,000 meters per month. Currently, the development of new fabrics consists of several trial-and-error cycles that are executed until the client requirements are met. This process is heavily sustained in the knowledge and intuition of the textile designer. When a designer leaves the company, all this knowledge is lost.

Figure 22 presents the process flow for the development of a new fabric. It starts with the selection of a previously developed fabric that is similar to new fabric physical property requirements. Based on the designer experience, she/he will then alter several parameters of the selected fabric, aiming to obtain the desired characteristics of the new fabric. Next, a fabric physical prototype is produced and inspected for its quality by running several laboratory tests. If the prototype passes the tests, it is presented to the client. Often, this is not the case and thus a new cycle is executed, where the designer attempts a new fabric design based on the previous prototype tests. Typically, a larger number of design cycles is needed (e.g., 20) until the fabric is ready for mass production, with each cycle requiring costs and time.

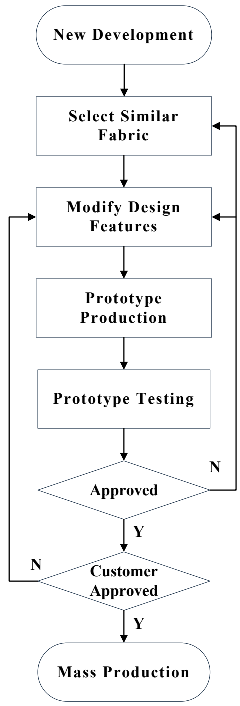

Figure 22: Flowchart for the development process of a new fabric (Y and N denote Yes and No, respectively).

Firstly, there is a high number of fabric design features combinations, thus the search space is quite large. Secondly, there are interactions between the features, where a desired physical property change might only occur if there are simultaneous alterations in multiple construction elements. Thirdly, producing changes that improve one physical property can prejudice other properties. For example, increasing the fabric elasticity can also increase the fabric bias distortion. Fourthly, the final composition of the fabric is a highly relevant information for textile clients and customers (e.g., % of cotton). Yet, it is not trivial to compute the precise final composition for a new fabric, since it can assume different types of yarns (each with a particular thickness) that can be repeated in different ways in the weft and warp elements of the fabric. In effect, the textile composition is often measured by using the produced fabric prototypes, assuming a manual counting of the number of weft and warp yarns for each fiber type contained in a fabric square of 2.54 centimeters.

4.4. Material and Methods

4.4.1 *Fabric Data*

The data was provided by a Portuguese textile company that creates and produces fabrics for fashion and clothing collections. We collected data from two main data sources: the *Enterprise Resource Planning (ERP)*, which included all the data related to fabrics, and the laboratory testing database, which contained the fabric quality tests performed between 2012 to 2019. We implemented then an *Extraction Transform Load (ETL)* process to merge the different data sources and clean the data. The ERP data included 34,998 fabric examples with 2,391 features per row. Using manual analysis and domain expert knowledge, the ERP features were filtered into a total of 805 potential relevant attributes. Since the number of potential attributes was high, we performed several feature selection iterations ending with 68 features. The final set of selected attributes is associated with a total of 8,650 records and it was obtained by executing several ML process iterations that involved the company designers, as described in our previous work (Ribeiro et al., 2020). In this work, 100 randomly selected fabrics are used to execute the prescriptive experiments. The remainder 8,550 examples are used to perform the predictive experiments (e.g., external 10-fold cross-validation).

Table 12 summarizes the adopted set of fabric attributes in terms of their **Attribute** name, **Description** and **Range** of the domain values. The first 13 rows of Table 12 are related to a fixed set of design attributes whose values can be changed by the fabric designer. They are common to all fabrics and therefore are used to create the 44 inputs of our predictive and prescriptive models (as explained below). The remainder rows of Table 12 are used as the targets of the predictive models. The physical properties are measured by laboratory tests that were executed on the final fabric prototypes (accepted for

Table 12: List of fabric design features.

Attribute	Description (data type)	Range
finished width	Width in centimeters (numeric)	[112, 155]
weave design	Weave pattern of the fabric (nominal code)	15 levels
reed width	Width of the reed in centimeters (numeric)	[157, 242]
denting	Number of the reed dents per centimeter (numeric)	[50, 252]
ends/dent	Number of yarns per dent (numeric)	{2, 3, ..., 7}
n_picks	Number of picks on loom per centimeter (numeric)	[13, 445]
warp total ends	Total number of threads on the warp (numeric)	[2448, 17970]
yarn repetitions	Number of yarn repetitions in warp or weft (numeric)	{0, 1, ..., 4}
yarn code	Identification code of the yarn (nominal code)	1,730 levels
op	Finishing operations (nominal code)	48 levels
t_pol	Number of finished threads per centimeter (numeric)	[0, 323]
p_pol	Number of finished picks per centimeter (numeric)	[0, 377]
weight/m^2	Weight (in grams) per square meter (numeric)	[0, 579]
Bias Distortion	Quality test result (numeric)	[1, 12.36]
Elasticity (warp)	Quality test result (numeric)	[5, 55]
Elasticity (weft)	Quality test result (numeric)	[5, 58.1]
Pilling	Quality test result (numeric)	[1, 4.5]
CO	Percentage of cotton in a fabric (numeric)	[0; 0.99]
CV	Percentage of viscose in a fabric (numeric)	[0; 0.54]
EL	Percentage of elastane in a fabric (numeric)	[0.01; 0.13]
PA	Percentage of nylon in a fabric (numeric)	[0; 0.83]
PES	Percentage of polyester in a fabric (numeric)	[0; 0.93]
WO	Percentage of pure wool in a fabric (numeric)	[0; 0.50]

mass production). It should be noted that each fabric can have one or more distinct tests. For fabrics that had the same test repeated with a slightly different value, we opted to compute the average values in order to get a single number per fabric and test. The analyzed dataset contained 15 different tests but only a tiny portion of the fabrics (27) had the full 15 test values. In this work, we model the most frequently measured physical properties, which correspond to the four tests shown in Table 12. Finally, the last 6 rows of Table 12 denote the final textile composition attributes, representing the percentage of the 6 types of most common fibers that are used in the yarns. Each textile composition type includes a combination of the fiber percentages, where the sum equals to 100%. For instance, the most popular composition type includes 63% of PES, 27% of CV, 7% of CO and 3% of EL. In the analyzed dataset, there are a total of 95 different types of textile compositions that include the six types of fibers, corresponding to 93% of fabrics produced by the company. In this work, the textile composition prediction is also modeled as a regression task due to two main reasons. Firstly, it avoids dealing with a large multi-class

(95) classification task that would produce poor results for the least represented classes. Secondly, it allows the reuse of the regression algorithms already developed for the physical property goal.

Figure 23 exemplifies how some of the design features are related with the textile fabric and machine settings. Structurally, a fabric is made of two primary components: warp and weft.

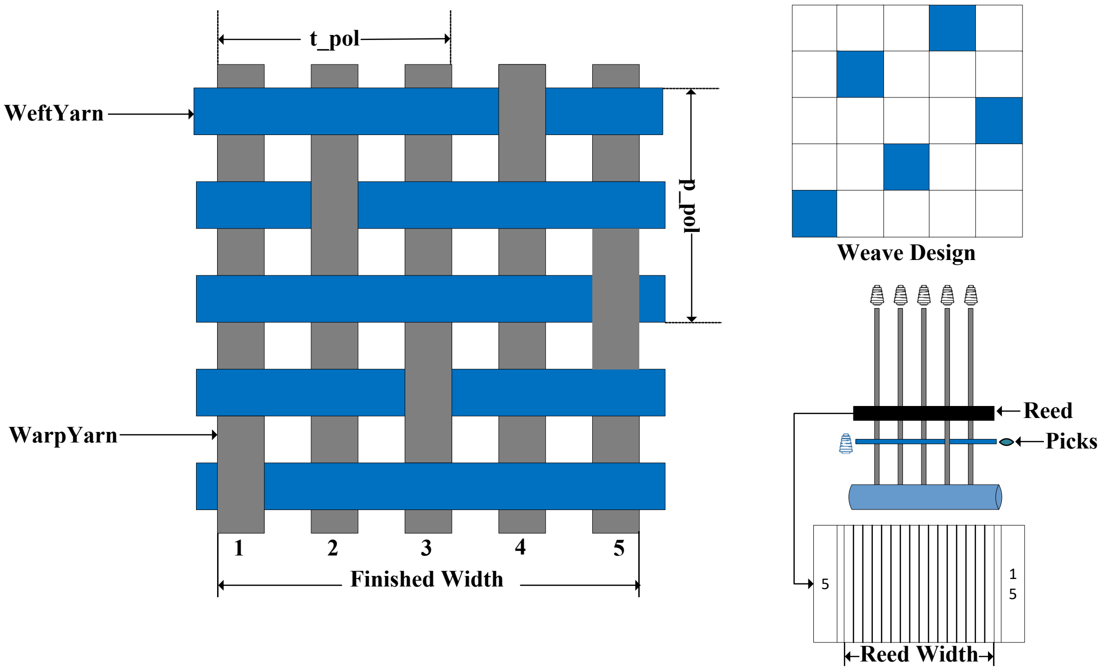

Figure 23: Visualization of some woven fabric and machine features.

The left of Figure 23 presents a view of a fabric that contains interlaced yarns related with the warp (blue color) and weft (grey color) components used in the weaving. In the example, the fabric has one type of yarn for the warp and another for the weft, both repeated five times. Since the warp is the set of yarns that are placed on the loom, it contains a minimum number of warp yarns to support the tension applied during the weaving process. A higher number of yarns per centimeter implies a better fabric quality.

To obtain the number of finished threads per centimeter (**t_pol**), the designer must define how many warp yarns will be put in a centimeter of the fabric, and for the **p_pol** attribute, she/he defines how many weft yarns will be per centimeter. To obtain the **finished width**, the distance between the first and last warp yarn of the fabric is measured.

The top right of Figure 23 presents the **weave design**, where the spaces between the yarns are eliminated and only the squares where warp yarns are over the weft yarns are shown. This distribution defines the type of the fabric, which in this case is a twill.

The bottom left image of Figure 23 presents a representation of a loom machine, where the weft yarns (grey color) are inserted in a loom, going through the reed with a specific width, that will press the warp (blue color) that is inserted in between the warps yarn threads, according to the structure implemented in the weave design. The reed has two main characteristics: the number of the reed dents per centimeter (**denting**, in this case 5) and the **reed width** (in this case 15), which is the distance from the first reed dent to the last one. When designing the fabric, it is necessary to define how many of the weft yarns will pass in each dent (attribute **ends/dent**).

The warp and weft elements of the fabric can contain a variable combination of yarns, ranging from 1 to 21 in our dataset. Moreover, each fabric is processed with a varying sequence of finishing operations, ranging from 1 to 47 in our data. In the analyzed data, the most commonly used yarns assume combinations of the six fibers types that appear Table 12 (e.g., cotton, viscose, elastane), while the prevalent finishing operations are drying, finishing and dyeing. In order to represent these repeated elements, in previous work (Ribeiro et al., 2020), we performed several feature engineering experiments, resulting in the final representation adopted in this work. For each fabric, the representation assumes a sequence with a maximum of $max_y = 6$ yarns for warp and then another similar sequence of $max_y = 6$ yarns for weft, allowing to encode 99.7% of the fabrics without any information loss. Each yarn is represented by two elements: a numeric unique code and the number of times the yarn appears in that specific element (warp or weft). When one element does not have up to $max_y = 6$ yarns, a zero padding is used to fill the empty yarn values. A similar approach was adopted to represent the finishing operations, defined in terms of a sequence of $max_{op} = 10$ operations, which represents 85% of the analyzed fabrics. Also, if a fabric is not processed with 10 operations, then a zero padding is used to fill the empty values. Thus, a total of 44 fabric design inputs are adopted by the predictive models, corresponding to: 6×2 values to denote the warp and weft yarns (total of 24 inputs), 10 values to code the finishing operations, and the other 10 attributes from Table 12 (e.g., finished width, weave design).

In terms of data preprocessing, the three categorical attributes in Table 12 (**weave design**, **yarn code** and **op**) were first transformed into numeric inputs. Then, all numeric inputs were standardized to a zero mean and one standard deviation (Z-score transformation). These transformations are needed for three main reasons. Firstly, several of the explored ML algorithms (e.g., ANN) only work with numeric inputs. Secondly, the ML is often more efficient when all inputs are standardized (Hastie et al., 2009). Thirdly, the computation of distance measures when comparing two sets of inputs is much simpler when all inputs are numeric and standardized (e.g., usage of the Euclidean distance).

Given that some of the categorical attributes present a high cardinality (e.g., the yarn code contains 1,730 distinct levels), we opted to transform all nominal attributes by using the *Inverse Document Frequency (IDF)* function (Campos et al., 2016):

$$IDF(x_i) = ln(\frac{n}{f_l}) \tag{5}$$

where x_i denotes a data attribute, *n* is the total number of instances and f_l is the number of occurrences (frequency) of the level $l \in x_i$ in the analyzed data. The IDF mapping is computed using only training data and then stored, in order to enable a future encoding for unseen data. If a new level appears on the unseen data, it is replaced by the highest *ln(n)* value present on the training data, which represents the most infrequent IDF value.

The advantage of the IDF method is that encodes a nominal attribute into a single numeric value, where the levels with a higher frequency are set near 0 (but with a larger difference between them), while the less frequent levels are coded close to each other and near a $IDF(x_i)$ maximum value. Thus, more frequent levels are more easily distinguished by the ML algorithms. And while the one-hot encoding is a popular categorical transform, assigning one Boolean value per nominal level, its usage would highly increase the input space, resulting in a very sparse representation that would enlarge the computational memory and effort required by the ML algorithms (Matos et al., 2022). Moreover, given that IDF produces a single numeric value for each categorical input, the computation of input distance measures is much straightforward than when adopting the one-hot method.

4.4.2 *Intelligent Decision Support System*

The proposed IDSS consists of three main modules (Figure 24): data extraction and processing, predictive and prescriptive. All modules were implemented by using the Python programming language. The first module is responsible for extracting the relevant textile data, converting it into the adopted fabric design representation (with 44 inputs), leading to a stored fabric design database.

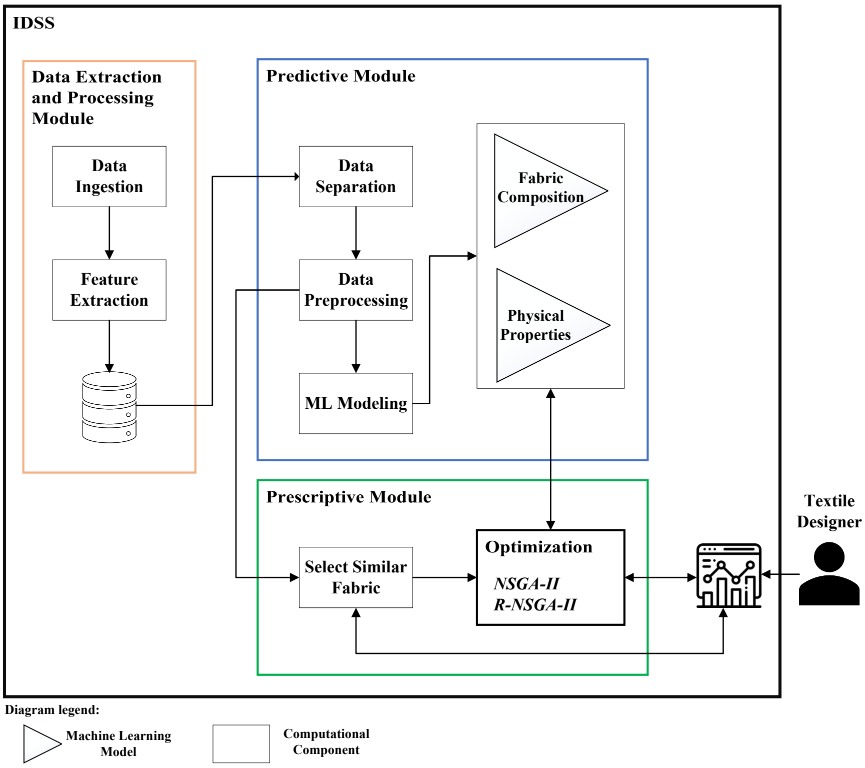

Figure 24: Flow diagram describing the components of the proposed IDSS.

The prediction module receives the data, which is then divided into training and test sets, according to the adopted cross-validation method. Then, the features are preprocessed (IDF and standardization). Next, a ML algorithm is trained and evaluated (using validation data), allowing to perform a model selection. The best ML per task is stored, allowing a later computation of fabric physical properties (first ML goal) and textile composition (second ML goal). More ML modeling details are provided in Section 4.4.3.

The prescriptive module performs two main operations. It can search for previously manufactured fabrics that are similar to some desired product specifications. This search is executed by iterating all data examples, aiming to minimize the Euclidean distance between the desired values and the historical data, assuming a standardized multidimensional space. It can also execute a EMO search for the best set of design inputs, returning a Pareto front of solutions. Section 4.4.4 presents further EMO details.

The proposed IDSS includes also a friendly dashboard that sets the interaction with the textile designer. This interaction can assume several possibilities. First, the designer can inspect the historical data and search for the most similar fabrics, when assuming a desired set of physical properties. Second, the selected fabrics can be changed in some of their components, with the IDSS presenting the predicted physical properties and fabric composition. Third, a set of selected fabrics can be fed into the EMO search, thus using a seeding procedure to generate the initial population (instead of a purely random process). Fourth, the Pareto front of solutions returned by the EMO can be further inspected, allowing to select one or more interesting candidate designs for prototype production. Fifth, once more prototypes are produced and tested, the predictive modules can be retrained using the recently acquired data. Sixth, if none of the suggested EMO designs are accepted by the designer, then the search can be rerun by reshaping some of the previously obtained Pareto solutions and then seeding them into the next EMO population. All these interaction scenarios were found interesting by the analyzed textile company. However, in order to obtain an objective evaluation of the EMO results (independent of a specific designer), in this work we assume a pure automated IDSS usage, with no designer interaction. First, we evaluate the predictive models (Section 4.4.3). After selecting the best ML approaches, we then evaluate the EMO methods (Section 4.4.4).

4.4.3 *Predictive Modeling*

In this work, the two goals (physical properties and textile composition) contain multiple regression outputs for the same number of inputs. In particular, there are four targeted physical properties and six types of fiber percentages (Table 12). Under this context, performing a MTR can be an interesting modeling approach, since it requires a single ML model to simultaneously model several equally important targets (Arashloo and Kittler, 2022). Thus, less effort is required to design and maintain a single MTR model when compared with several STR ones. In this work, we adopt a deep multilayer ANN, which is a natural ML model for MTR, since it can directly model several regression targets without requiring any learning algorithm changes by simply assigning a distinct output node for each target. Another interesting ML modeling possibility is to adopt an AutoML tool, which automates the search for the best ML algorithm and its associated set of hyperparameters. Thus, it alleviates the ML design, allowing future model updates to new data without human effort. In this work, we adopt the `H2O` AutoML tool (LeDell and Poirier, 2020), which achieved good results in a recent AutoML benchmark study (Ferreira et al., 2021). Since the `H2O` AutoML only performs a STR, the `H2O` tool is set to generate a distinct ML model for each output target (e.g., four distinct models will be searched for the first ML goal).

For the MTR experiments, we implemented a ANN using the Keras Python module (Chollet, 2021). Let $(I, L_1, ..., L_h; O)$ denote a vector with the layer sizes, where I is the input layer size, h is the number of hidden layers, and O is the output layer. Some preliminary experiments were held to set the ANN structure using older textile data and triangular-shaped multilayer perceptrons, in which each subsequent layer size is smaller (Matos et al., 2019). As the result of these experiments, the MTR ANN was set to use the ReLU activation function under two types of structures: first ML goal (physical properties) – $(44, 28, 20, 12, 4)$; and second ML goal (textile composition) – $(44, 18, 12, 6)$. The popular ADAM optimizer was used to adjust the ANN weights during the training phase (Kingma and Ba, 2015), assuming the *Mean Absolute Error (MAE)* loss function, an early stopping (with 10% of the training data being used as the validation set) and a maximum of 1,000 epochs.

Regarding the STR experiments, the adopted AutoML tool was configured to automatically select the optimal regression model and its hyperparameters for each validation fold data by minimizing the MAE measure, using an internal 10-fold cross-validation applied over the training data. The AutoML was run with its default configuration values, including a maximum execution duration of 1 hour. A total of six different regression approaches are searched by the AutoML tool: *Random Forest (RF)*, *Extremely Randomized Trees (XRT)*, *Generalized Linear Models (GLM)*, *Gradient Boosting Machines (GBM)*, *XGBoost (XG)* and *Stacked Ensemble (SE)*. The H2O tool utilizes a grid search to set the hyperparameters for GLM (1 hyperparameter), GBM (9 hyperparameters), and XG (10 hyperparameters), while RF and XRT are configured using their default hyperparameters. As for the SE, the tool uses GLM as the second-level learner and compares three distinct ensemble methods: one with the best model of each individual approach (e.g., the best XG model), one with the best 100 models and one with all trained models.

To evaluate the ML models for both regression tasks, an external 10-fold cross-validation was implemented, and regression quality was assessed by using the MAE and the *Normalized Mean Absolute Error (NMAE)* metrics. The NMAE measure presents a standardized MAE result, thus scale independent, showing the error as a percentage of the response range. The MAE and NMAE errors are calculated as (Oliveira et al., 2017):

$$
\begin{aligned}
MAE &= \frac{\sum_{j \in \mathcal{T}} |y_{i,j} - \hat{y}_{i,j}|}{\#\mathcal{T}} \\
NMAE &= \frac{MAE}{\max(y_i) - \min(y_i)}
\end{aligned}
\tag{6}
$$

where $\mathcal{T}$ denotes the test set with a cardinality of $\#\mathcal{T}$, $y_{i,j}$ and $\hat{y}_{i,j}$ represent the desired and predicted value for output target i and test example j, and max (y_i) and min (y_i) corresponds to the highest and the lowest values of the target y_i (considering all available data). For both metrics, the closer the value is to zero the better is the regression.

Regarding the specific second ML goal, both STR or MTR prediction models can return values such that the sum of the six fiber percentages is not equal to 100%, resulting in unfeasible textile compositions (this direct normal output usage is termed here as the **N** approach). To solve this issue, we explore two output post-processing strategies that return feasible compositions. The first strategy assumes a proportional normalization (**P**), where first all negative values are replaced by zero and then transformed according to:

$$\hat{y}'_{i,j} = \frac{\hat{y}_{i,j}}{\sum_{k=1}^{K}(\hat{y}_{k,j})} \tag{7}$$

where $\hat{y}'_{i,j}$ is the transformed value of the predicted fiber target $\hat{y}_{i,j}$, $i \in \{1, 2, ..., K\}$ (one value for each fiber type, thus $K = 6$). The second strategy works similarly to the previous one except that the zero or positive values are now transformed by using the softmax (**S**) function:

$$\hat{y}'_{i,j} = \frac{\exp(\hat{y}_{i,j})}{\sum_{k=1}^{K}(\exp(\hat{y}_{k,j}))} \tag{8}$$

To select the best strategy and provide a final composition class value, we compute the overall classification accuracy for a given low tolerance T value, where a class composition is considered correct if all six fiber percentages are correctly predicted within the T absolute tolerance value. This measure is based on the *Regression Error Characteristic (REC)* curve concept, which allows an easy visual comparison of different prediction methods (Bi and Bennett, 2003). In this work, we selected 10 small tolerance values within the range $T \in \{0.01, 0.02, ..., 0.10\}$.

4.4.4 *Prescriptive Modeling*

A possible solution $s = (s_1, s_2, ..., s_I)$ is represented as a sequence of I =44 numeric values (as explained in Section 4.4.1), assuming the IDF standardized input space, as it allows to provide the same importance to each value when computing distance measures. Each s_i value is set within the range $[\min(s_i), \max(s_i)]$, where min (s_i) and max (s_i) denote the minimum and maximum values of the IDF standardized space using the training data. We repair solutions in two different types of design features: yarns and finishing operations. A yarn is represented by a code and its number of repetitions. Thus, when one of the two values is empty (thus 0 in the original space), then the other value is also set to zero. For instance, if a yarn code is 0 ("empty"), then the number of its repetitions is also set to 0. Moreover, we also assure that both the warp and weft have at least one yarn, set to the first yarn type with one repetition (if needed). In the case of the finishing operations, any zero intermediate value (no operation followed by an operation) is shifted right, such that the initial part of the sequence contains concrete operating values. Furthermore, since a fabric needs to be processed by at least two finishing operations, we set the lower bound as min $(s_i) > 0$ (in the original space) for the first two finishing operations values of the sequence.

The EMO goal is to simultaneously minimize the absolute error predictive error (f_p) and the training distance (f_d) associated with a candidate solution s. Let $\mathcal{D}$ denote the data used to train $\mathcal{M}_i$, where $\mathcal{M}_i$ is the selected ML method to predict a desired physical property i, under the mapping: $i = 1$ – bias distortion; $i = 2$ – warp elasticity; $i = 3$ – weft elasticity; and $i = 4$ – pilling. Let $\mathcal{Y}_i(s)$ and $\widehat{\mathcal{Y}}_i(s)$ denote normalized (using a min-max normalization within the [0,1] range) desired target and predicted values when using model $\mathcal{M}_i$ and the input solution s. In this work, we assume normalized objective functions, set within the [0,1] range. This facilitates the EMO result analysis since each computed objective can be interpreted as a percentage value:

$$\begin{aligned} f_p(s) &= \frac{\sum_{i=1}^{4} |\mathcal{Y}_i(s) - \widehat{\mathcal{Y}}_i(s)|}{4} && \text{(predictive error)} \\ f_d(s) &= \frac{\arg\min_{x \in \mathcal{D}} (\sqrt{(x-s)^2})}{\sqrt{I}} && \text{(distance)} \end{aligned} \tag{9}$$

where $f_p(s)$ corresponds to the MAE error for all four desired properties and $f_d(s)$ is the minimum Euclidean distance of solution s when compared with the training set $\mathcal{D}$. For the second objective, the term $\sqrt{I}$ corresponds to a high distance value (deviation of 1 for each of the analyzed I =44 inputs). As explained in Section 4.2, ideally both f_p and f_p should present lower values. A low predictive error (f_p) means that the desired target properties are reached, while a small distance (f_d) reflects that the selected inputs (s) are close to the known input space, meaning that the predictions should be more reliable.

Two EMO approaches are compared in this work: NSGA-II and R-NSGA-II, as implemented in the `pymoo` Python module (Blank and Deb, 2020). The NSGA-II, adopts several distinct features (e.g., elitist strategy, fast crowded distance estimation procedure) and it is considered a parameterless approach (Deb et al., 2002). When compared with other EMO algorithms, such as based on the Hypervolume measure (e.g, SMS-EMOA), NSGA-II tends to obtain competitive results when only two or three objectives are optimized (Chiandussi et al., 2012). As for R-NSGA-II, it consists of a more recent NSGA-II variant that assumes a modified survival selection, which is based on one or more user-defined reference points (Deb and Sundar, 2006).

The EMO methods return a Pareto curve of optimized solutions, containing a set of trade-off points. Once the Pareto curve is optimized and for the user selected trade-offs, the IDSS computes the prediction of the textile composition (using the second ML goal prediction models), such that the user can further inspect the quality of the obtained solutions. In order to obtain a single measure per Pareto curve, we adopt the hypervolume measure, which is computed by defining a baseline reference point (anti-optimal) (While et al., 2006). The higher the hypervolume value, the better is the Pareto curve optimization.

For the evaluation of the EMO methods, we use the selected 100 external fabric records, that are not used in the predictive experiments. The best physical property prediction models (as shown in Section 4.5.1) are selected and retained with all predictive experiment data (8,550 records). The EMO methods are then run by adopting an initial random population of individuals.

4.5. Experiments and Results

All experiments were conducted using code written in the Python programming language. The experiments were executed on a personal computer with an Intel Core i7 2.20GHz processor, with 6 cores, a NVIDIA GeForce GTX 1050 Ti, using a Windows operating system.

4.5.1 *Physical Property Prediction Results*

Table 13 presents the physical property predictive performance results. For each one of the predicted tasks (**Property**), we compare the two **STR** (AutoML) and **MTR** (deep ANN) approaches. The results are shown in terms of the mean MAE and NMAE values, associated with its student-*t* 95% confidence intervals, for the external 10 folds, with the best values being highlighted by using a **boldface** text font (statistical significance is measured by executing a paired t-test). When comparing both approaches it becomes clear that **STR** is the best regression strategy, achieving the lowest regression errors for all four targets, with the NMAE values ranging from 4.05% to 11.22%. Thus, the STR approach, as provided by the AutoML modeling, is the selected regression approach that is adopted by the EMO methods.

Table 13: Predictive results for each quality test target (mean MAE and NMAE test set values pm 95% confidence intervals; best results per task are in **bold**).

	STR		MTR	
Property	**MAE**	**NMAE**	**MAE**	**NMAE**
Bias Distortion	**0.555**$\pm$0.199	**5.92**%$\pm$0.448	0.558$\pm$0.028	6.62%$\pm$0.479
Elasticity (warp)	**2.358**$\pm$0.178	**4.81**%$\pm$0.355	3.599$\pm$1.366	7.38%$\pm$2.878
Elasticity (weft)	**1.952**$\pm$0.041*	**4.05**%$\pm$0.175*	3.529$\pm$1.326	7.31%$\pm$2.740
Pilling	**0.350**$\pm$0.008*	**11.22**%$\pm$0.306*	0.597$\pm$0.046	19.13%$\pm$1.648

$*$ - Statistically significant when compared with MTR (p-value$<$0.05).

For demonstration purposes, Figure 25 shows the REC curves for the four targets and the 6th external k-fold iteration. Each REC curve plots the percentage of correctly predicted examples (y-axis) for a given absolute error tolerance (x-axis) (Bi and Bennett, 2003). The curves confirm that the **STR** presents slightly better results for the pilling and bias distortion tests, and much better results for both elasticity tests. For example, for the warp elasticity, the STR method correctly predicts around 80% of the examples when a small tolerance of T =0.1 points.

To complement the visualization of the obtained STR results, Figure 26 presents the scatter plots of the measured (x-axis) versus the predicted values (y-axis), complemented with the coefficient of determination (R^2) for the specific 6th external fold. This metric has a positive orientation, that is, the closer the result is to 1 the better are the predictions.

Visually, it can be seen that the predictions for the elasticity warp and elasticity weft tests (top of Figure 26) are close to the ideal prediction diagonal line and both present very good R^2 values of 0.90 and 0.87 respectively. While the same effect is not that visible for the pilling predictions (bottom right of Figure 26), it should be noted that the real values are mainly distributed in 6 clusters, and the predicted values variate within each cluster, resulting in an interesting R^2 of 0.64. The bias distortion test presented the lowest R^2 (0.32), with a higher concentration of values between 1.5 and 4, which confirms that this property is more difficult to be predicted. Nevertheless, the REC curve (bottom left plot of Figure 25) shows that, for the same 6th external fold, the selected **STR** prediction model for the

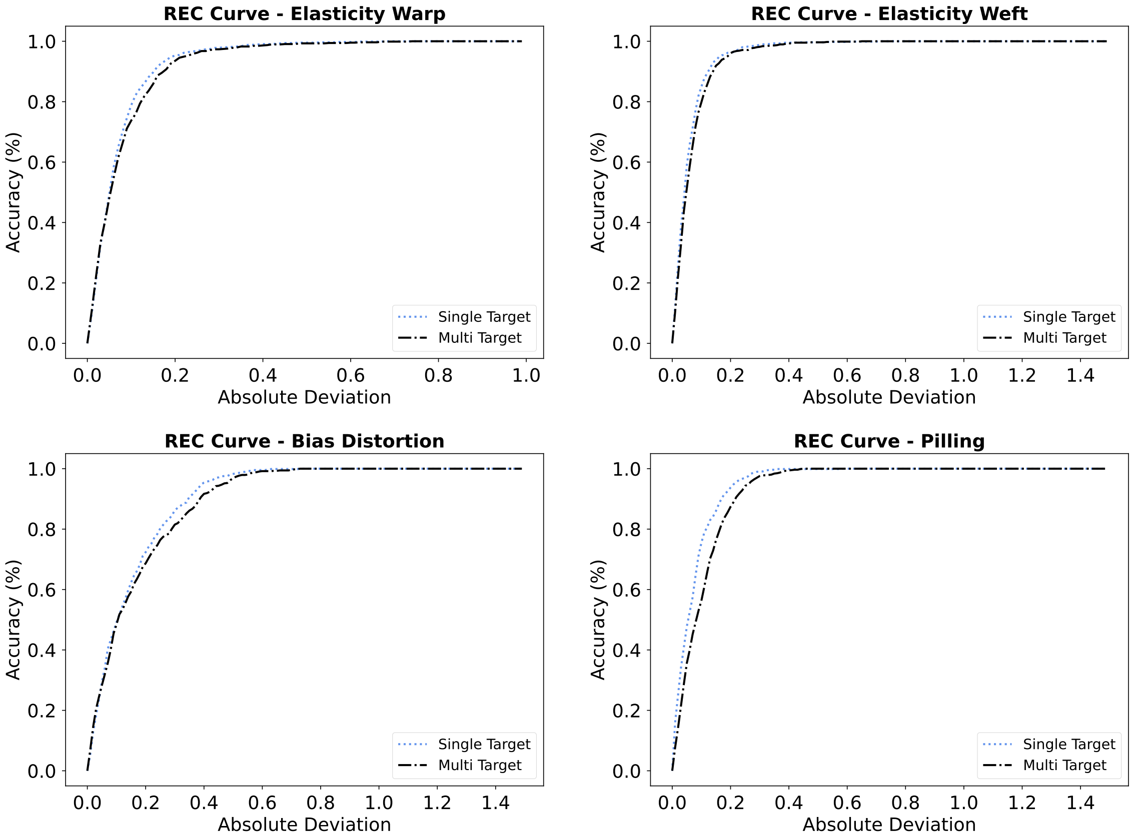

Figure 25: REC curves of elasticity warp (top left), elasticity weft (top right), bias distortion (bottom left) and pilling (bottom right) predictions.

bias distortion is still capable of predicting around 50% of the examples for a 0.1 tolerance and around 80% of the examples for a 0.25 tolerance value.

4.5.2 *Textile Composition Prediction Results*

Table 14 presents the predictive performance results of the test data regarding the fabric composition. For each predicted task (**%Fiber**), we compare the two regression approaches (**STR**, **MTR**). The results are shown in terms of the mean MAE values (with their student-*t* 95% confidence intervals), for the external 10 folds with the best values being highlighted by using a **boldface** text font. It should be noted that only the MAE error is computed, since the targets are already scaled within [0,1], thus the NMAE values are identical to MAE in this case.

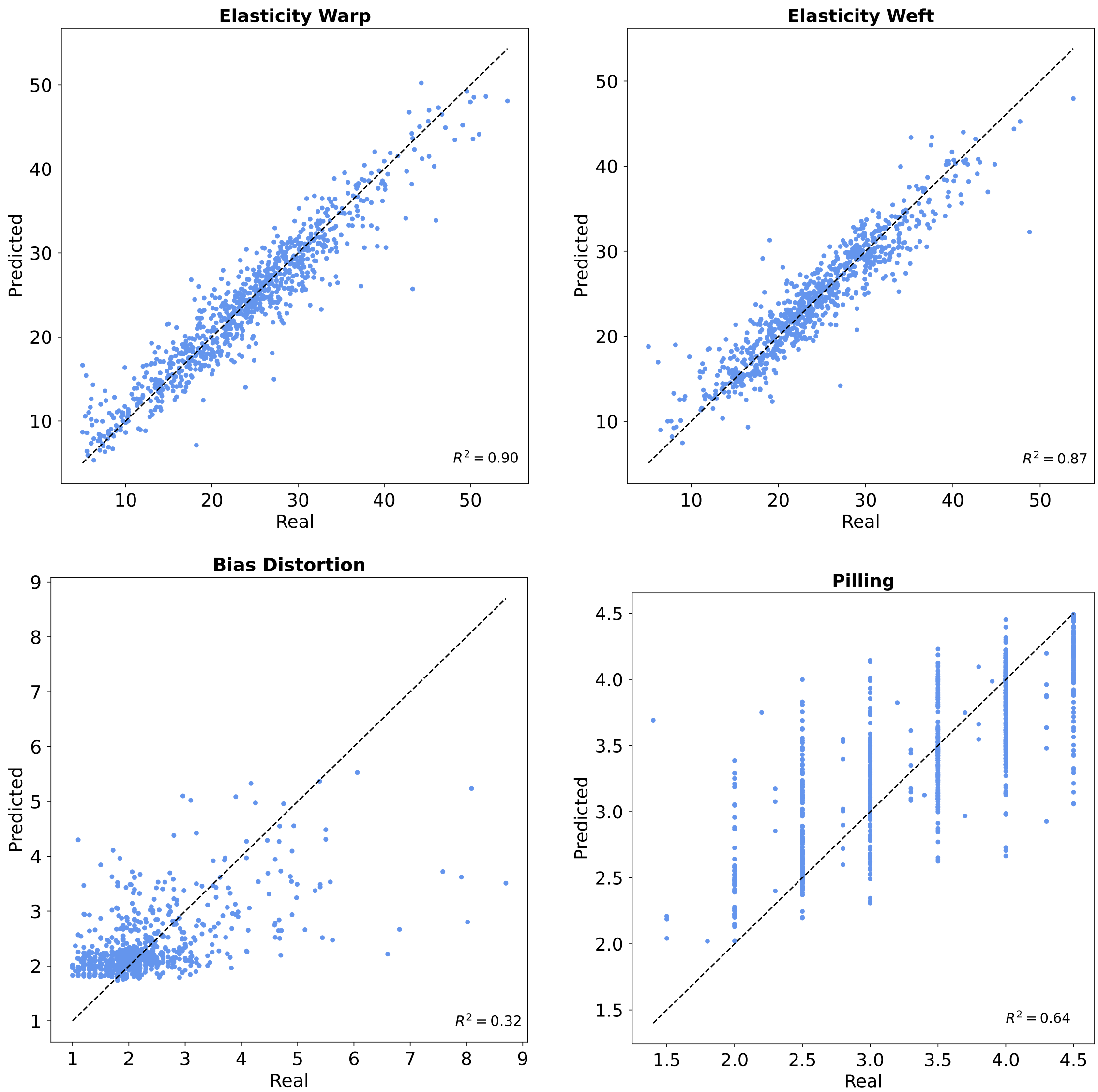

Figure 26: Regression scatter plot of elasticity warp (top left), elasticity weft (top right), bias distortion (bottom left) and pilling (bottom right) predictions.

Similarly to the physical property results, the **STR** approach presents the lowest MAE values for all six types of fibers, ranging from 0.002 to 0.01. It also obtains the lowest average MAE value over all six fibers (difference of 0.007 points when compared with **MTR**).

Additional comparative results are shown in Table 15, which presents the composition classification accuracy (in %) when assuming a small tolerance (**T**). For each regression approach (**STR** and **MTR**), we compare the three output normalization strategies (**N**, **P** and **S**), with the best results for a given tolerance value being highlighted by using a **boldface** text font. The results clearly favor the **STR** regression approach. As for the output normalization strategies, all three strategies provide high clas-

Table 14: Fabric composition predictive results (mean MAE test set values associated with its student-t 95% confidence intervals; best results are in **bold**).

	MAE	
%Fiber	**STR**	**MTR**
CO	**0.009**$\pm$0.002	0.013$\pm$0.002
CV	**0.006**$\pm$0.000*	0.017$\pm$0.001
EL	**0.002**$\pm$0.000*	0.006$\pm$0.000
PA	**0.002**$\pm$0.001*	0.005$\pm$0.001
PES	**0.010**$\pm$0.001*	0.020$\pm$0.002
WO	**0.002**$\pm$0.001*	0.011$\pm$0.002
Average	**0.005**	0.012

$*$ - Statistically significant when compared with MTR (p-value$<$0.05).

Table 15: Percentage of fabric compositions correctly predicted per approach (best results per T value are in **bold**).

	STR			MTR		
T	**N**	**P**	**S**	**N**	**P**	**S**
0.01	**87%**	**87%**	**87%**	56%	58%	56%
0.02	**89%**	**89%**	**89%**	66%	65%	66%
0.03	**90%**	**90%**	**90%**	73%	72%	73%
0.04	**91%**	**91%**	**91%**	81%	76%	81%
0.05	**92%**	**92%**	**92%**	84%	84%	84%
0.06	**93%**	**93%**	**93%**	86%	86%	86%
0.07	93%	**94%**	93%	86%	86%	86%
0.08	**94%**	**94%**	**94%**	86%	87%	86%
0.09	**94%**	**94%**	**94%**	87%	87%	87%
0.10	94%	**95%**	94%	87%	87%	87%

sification accuracy values even for very low tolerances (e.g., 87% when T=0.01%). Given that there is a large number of different composition classes (95), this is a very interesting result, confirming the value of the proposed ML textile composition approach. In particular, the second output normalization strategy, based on proportions (**P**), obtained a slight increase of 1 percentage point for the tolerance values of $T = 0.07$ and $T = 0.10$. Given that when compared with the no normalization strategy (**N**), it presents the additional advantage of always showing feasible compositions, the **STR** approach with the **P** output transformation method was selected to be used in the proposed IDSS system.

4.5.3 *Fabric Input Optimization Results*

The selected 100 external fabrics for the EMO experiments (not used in the predictive experiments) present the following physical property ranges: Bias Distortion – [1.1, 8.9]; Elasticity Warp – [5.2, 45.5]; Elasticity Weft – [5.4, 44.3]; and Pilling – [2, 4.5]. A distinct EMO execution is performed for each target fabric, thus 100 runs were executed for each of the tested NSGA-II and R-NSGA-II methods. The initial populations were randomly generated (as explained in Section 4.4.4). Moreover, all solutions were repaired (e.g., removal of yarn codes when there are no repetitions) before computing the two objective functions.

The NSGA-II and R-NSGA-II algorithms were configured with a check procedure that eliminates duplicates, ensuring that the mating produces offspring that are different from themselves and the existing population regarding their design space values. The NSGA-II was setup with the default values provided by the `pymoo` Python module: population size of 100, two-point crossover with 90%, polynomial mutation probability of 20%. After some preliminary experiments, in which the hypervolume measure was monitored for 5 fabrics, the total number of generations was set to 200 generations, with an average execution time of 1,750 seconds per fabric. The R-NSGA-II method was setup with the same configuration. Given that in this domain it is highly relevant to obtain a low predictive error and a low input to training set distance, two reference points were adopted close to the ideal (0,0) point: $(0.05, 0.01)$ and $(0.01, 0.05)$. The goal is to guide the R-NSGA-II to obtain trade-off points near the (0,0) region. R-NSGA-II contains an additional parameter, ϵ that was set to a low value (0.01), which increases the search pressure to select points closer to the two reference points. The R-NSGA-II had an average execution time of 1,852 seconds per fabric. Following a similar reasoning to the one used for the setting the R-NSGA-II reference points, we defined the baseline reference point for the hypervolume metric as $(0.3, 0.3)$ when evaluating the EMO methods. For each optimized fabrics, we discard the solutions that are outside that area and calculate the hypervolume.

Table 16 presents the obtained hypervolume performance results for the two EMO methods (% values, where 100% denotes the perfect Pareto curve). For each **Method**, we present the mean hypervolume value associated with its student-*t* 95% confidence interval (column **Mean**), and the **Median** and its associated nonparametric Wilcoxon-Signed-Rank 95% confidence interval (since this interval is not symmetric, it is fully shown in a different column) (Hollander et al., 2013). Statistical significance is measured by executing paired *t*-test (for the **Mean** values) and Wilcoxon tests (for the **Median** values). When comparing both methods, it becomes clear that the R-NSGA-II is the best approach to optimize fabrics development. It outperforms the NSGA-II in all the selected metrics, with a difference of 5 points for the mean and median.

Table 16: Hypervolume performance results (in %) for the 100 fabrics (best results per approach are in **bold**).

Method	Mean	Median	Median Interval
NSGA-II	20.86$\pm$3.25	24.31	[21.21, 28.04]
R-NSGA-II	**25.95$\pm$4.31**	**34.22***	**[30.09, 38.78]**

$*$ - Statistically significant when compared with NSGA-II (p-value$<$0.05).

To complement the hypervolume results, for each optimized Pareto front, the nearest points to the ideal (0,0) solution were computed, assuming an Euclidean distance under the optimized multi-objective space. These points represent potential interesting solutions that are associated with both low predictive error and distance values. Table 17 summarizes the obtained results in terms of the mean and median values of all 100 selected points for each individual objective (f_p and f_d). Similarly to Table 16, the student-*t* and Wilcoxon-Signed-Rank 95% confidence intervals also associated with the mean and median values.

The results confirm that R-NSGA-II is the best approach for the optimization process. In effect, R-NSGA-II presents lower mean and median values for both objectives, with a difference of 2 percentage points for the distance (f_d) and 0.03 points for the predictive error (f_p) when compared with NSGA-II. The R-NSGA-II optimized points that are closer to the perfect (0,0) point present similar mean and median values, which are 11% for the predictive error (f_p) and 16% for the input distance (f_d).

Table 17: Summary of the selected point results (best results per approach are in **bold**)

Method		**Mean**	**Median**	**Median Interval**
NSGA-II	Distance (f_d)	0.18 ± 0.01	0.18	[0.17, 0.19]
	Predictive error (f_p)	0.14 ± 0.01	0.14	[0.12, 0.16]
R-NSGA-II	Distance (f_d)	**0.16±0.01***	**0.16***	**[0.15, 0.17]**
	Predictive error (f_p)	**0.11±0.01***	**0.11***	**[0.10, 0.12]**

$*$ - Statistically significant when compared with NSGA-II (p-value<0.05).

For demonstration purposes, we selected two fabrics (#64 and #82) from the 100 external fabrics used to evaluate the optimization process. Figure 27 presents the evolution of both EMO methods in terms of a percentage hypervolume measure ($y-$axis) through the executed 200 generations. It should be noted that the measure is calculated for the baseline reference point $(0.3, 0.3)$, thus it returns a value of zero for the first EMO generations, since these include worst solutions than the baseline. The plots show a substantial improvement that is obtained by R-NSGA-II when compared with NSGA-II. In fact, R-NSGA-II requires fewer generations to start to obtain solutions that are within the evaluation range (e.g., 75 generations for fabric #64), and after 200 generations, the difference in the hypervolume % value of both fabrics is considerable, with an improvement of 28 percentage points for fabric #64 and 33 percentage points for fabric #82.

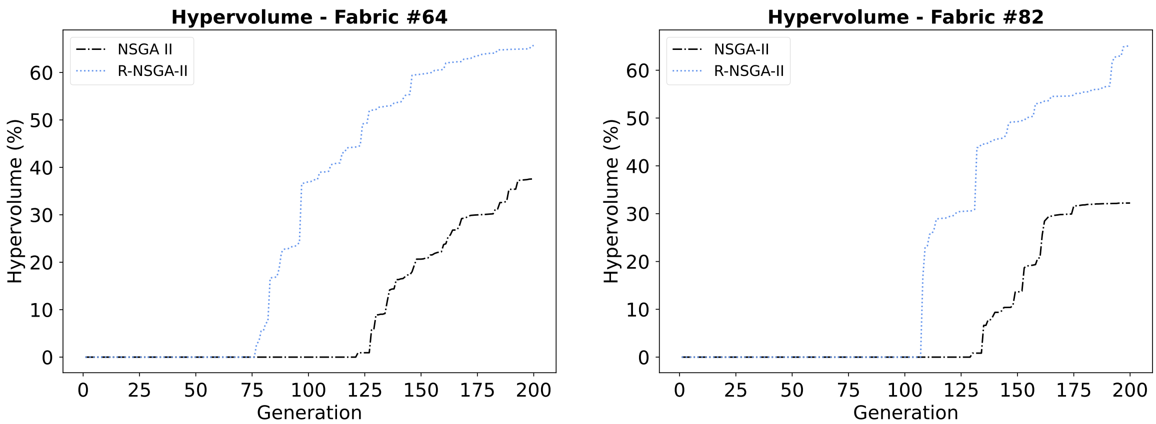

Figure 27: NSGA-II and R-NSGA-II hypervolume ($y-$axis, in %) generation evolution ($x-$axis) for fabric #64 (left) and fabric #82 (right).

Another demonstration example is provided in Figure 28, which presents the Pareto front of both algorithms after 200 generations when considering the #64 (left) and #82 (right) fabrics, with the defined reference points utilized by R-NSGA-II. Considering fabric #64, the Pareto front of NSGA-II contains 31 solutions, with the distance objective ranging from 0.172 to 0.290 and predictive error objective ranging from 0.023 to 0.060. As for R-NSGA-II, it returns a Pareto curve with 100 non-dominated solutions, with the distance measure ranging from 0.085 to 0.290 and the predictive error going from 0.015 to 0.055. The computed hypervolume (HV) is thus much higher for R-NSGA-II (66%) when compared with NSGA-II (38%). Regarding fabric #82, the Pareto front of NSGA-II contains 37 solutions (inputs distance ranging from 0.185 to 0.298; predictive error within 0.026 to 0.068; hypervolume of 32%), while the R-NSGA-II returns a Pareto front with 44 solutions (distance ranging from 0.077 to 0.229; predictive error from 0.023 to 0.076; hypervolume of 65%). The nearest point to the ideal point (0,0) is also shown in both plots. They belong to the R-NSGA-II optimized fronts and correspond to: fabric #64 – (f_d =0.155,f_p =0.026) and fabric #82 – (f_d =0.097,f_p =0.053).

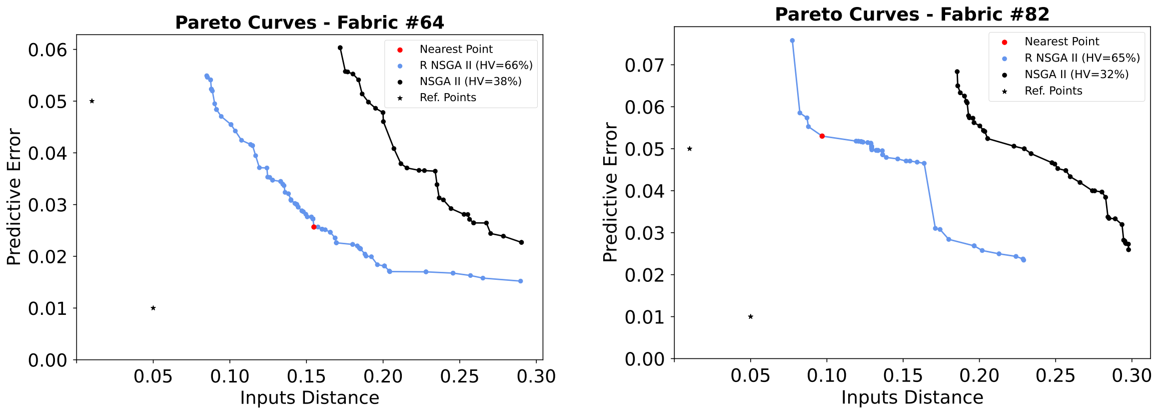

Figure 28: NSGA-II and R-NSGA-II Pareto curves for fabrics #64 (left) and #82 (right). The respective hypervolume (HV) % values are shown in parentheses. The $x-$axis denotes the inputs distance (f_d), while the $y-$axis represents the predictive error (f_p).

These examples confirm that R-NSGA-II outperforms NSGA-II. In fact, R-NSGA-II tends to provide more Pareto front solutions and lower values for both optimized objectives. Thus, we select R-NSGA-II for the proposed IDSS.

To illustrate the EMO convergence, Figure 29 presents the evolution of the solutions optimized by the R-NSGA-II towards the Pareto-optimal front. Each point represents a potential solution and line segments are used to connect the points that belong to the Pareto front. A color scheme is employed to facilitate the visual inspection of the plots, ranging from light grey (first generation) to full black (last generation). Both #64 and #82 fabrics have an initial fast convergence, with significant movements of the Pareto front towards the bottom left region.

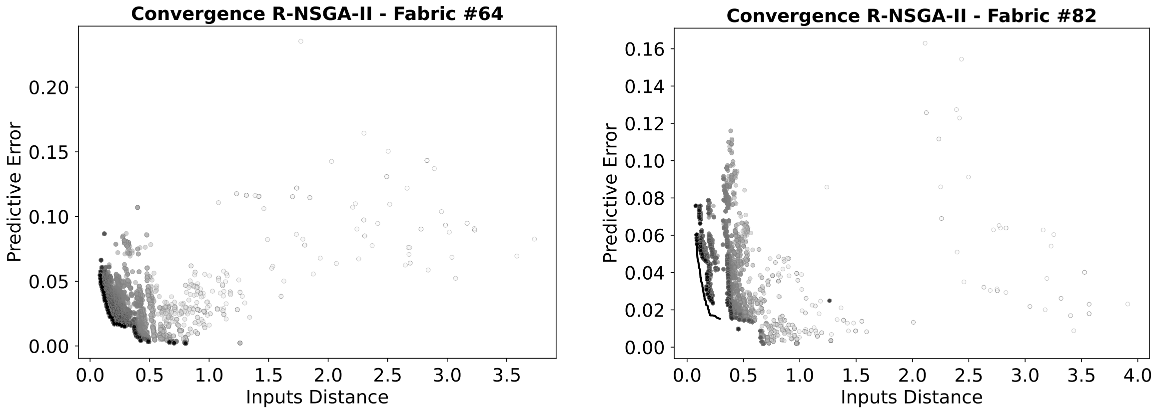

Figure 29: Example of the convergence of R-NSGA-II algorithm for fabric #64 (left) and fabric #82 (right). The $x-$axis denotes the inputs distance (f_d), while the $y-$axis represents the predictive error (f_p).

4.6. Conclusions

Due to fashion trend dynamics, the Textile and Clothing industry is constantly designing new fabrics. However, the creation of a new fabric is a nontrivial, costly and time-consuming process, often based on the textile designer experience and requiring a large number of design, prototype production and laboratory testing cycles. Aiming to reduce the number of fabric prototype production attempts, this paper proposes a purely data-driven and automated (IDSS) that combines predictive and prescriptive analytics.

Using a large and realistic set of 44 fabric construction components (e.g., yarns used for the warp and weft components), two main predictive goals were used by the proposed IDSS, the estimation of four desired physical properties (e.g., warp elasticity) and the detection of the final textile composition (e.g., % of cotton, % of polyester). Thousands of historical fabric production records, collected from a Portuguese textile company, were used to compare two distinct Machine Learning (ML) approaches, a Single-Target Regression (STR) using an Automated ML (AutoML) approach, and Multi-Target Regression (MTR) performed by a deep Artificial Neural Network (ANN).

Overall, the STR approach provided the best results for both predictive goals, resulting in a Normalized Mean Absolute Error (NMAE) values that ranged from 4% (weft elasticity) to 11% (pilling), when predicting the physical properties, and a textile composition classification accuracy of 87%, when assuming a small tolerance of 0.01 for predicting the percentages of six main types of fibers (e.g., cotton, polyester).

Regarding the prescriptive analytics, they assume an Evolutionary Multi-objective Optimization (EMO) search. Using the best physical property prediction models (provided by STR), the EMO optimizes the set of 44 fabric construction inputs that simultaneously minimizes the physical property predictive error and its distance to the training set. Using 100 additional fabric records (not used in the predictive experiments), two EMO methods were compared: NSGA-II and R-NSGA-II. Thus, each method was executed 100 times, resulting in 100 distinct Pareto fronts. Several EMO measures, such as hypervolume and selection of the Pareto front points that were closer to the ideal solution, allowed to confirm the R-NSGA-II as the best EMO method, thus being included in the proposed IDSS. On average, the R-NSGA-II selected points closer to the perfect (0,0) value and that are associated with an 16% distance and 11% predictive error.

The predictive and prescriptive results were shown to the textile company experts, which provided very positive feedback. In effect, in future work, we intend to deploy the proposed IDSS in the real textile company environment, aiming to gather further feedback and confirm its potential to reduce the costs and time associated with each new fabric development. Also, we plan to study a many objective approach (Guo, 2022), which is more challenging than the approached bi-objective EMO, aiming to simultaneously minimize the individual predictive error when predicting a larger set of fabric physical properties (e.g., abrasion, seam slippage, stability to steam).

5

CONCLUSIONS

5.1. Overview

The Textile and Clothing industry is one of the most important sectors in the manufacturing industry that is now transitioning into the fourth industrial revolution (also referred to as Industry 4.0), with the integration of new technologies, such as *Internet of Things (IoT)*, *Cyber-Physical System (CPS)*, Big Data Analytics and cloud computing, improving processes and products and transforming existing business models. These improvements led to an exponential increase in the amount of produced and stored data. Therefore, a large volume of data is now available to build data-driven systems, that can potentially add new insights and process improvements to different manufacturing processes.

This doctoral project was developed under a *Research and Development (R&D)* project, named *Tex-Boost - less Commodities more Specialties*. The project consortium involved a real textile company and *Centro de Computação Gráfica*, and whose main objective was digitization and dematerialization of prototype fabrics aiming to create new products, processes or services, innovative and technologically advanced. This R&D project had two main tasks: the first task was the extraction of data from multiple data sources and its integration into a unified data warehouse, oriented for business analysis, and the second task was to research and develop an *Intelligent Decision Support System (IDSS)*, utilizing state-of-the-art technologies within the concept of Industry 4.0, to assist in new fabric design process. This PhD is focused on the second task, although it will make use of the unified data warehouse, since it will be the IDSS data source.

Currently, the development of new textile fabrics consists of several trial-and-error cycles that are executed until the client requirements are met, making it a very expensive and time-consuming task. This is a process heavily sustained in the knowledge and intuition of the textile designer, which means that whenever a textile designer leaves the company, all the knowledge is lost. Thus the company wanted to research the design of an IDSS that reduced the number of iterations needed to design a fabric that met the customer specified requirements.

After conducting a literature review, it was verified there are no studies that support the whole process of textile design of new textile fabrics based on data analytics. In particular, studies that use predictive models to estimate several textile properties of the fabric and then use prescriptive methods, attempting to optimize the different construction parameters such as machine settings, warp and weft yarns and finishing operations to fed the predictive model such that the desired client properties are reached. Thus, the main objective of this doctoral project was to design and implement an IDSS that could potentially reduce the number of iterations needed to design a fabric that met the customer specified requirements. To address this challenge, we defined two tasks: the prediction of laboratory quality tests (e.g., Bias Distortion, Pilling) and the optimization of fabric design features.

For the laboratory quality tests, the textile company provided a set of tests that were considered the most important to predict. In the first experiment, we selected two tests (the tear strength in warp and weft directions), and performed three *Cross-Industry Standard Process for Data Mining (CRISP-DM)* iterations aiming to explore distinct data preprocessing operations and the inclusion of new features. Historical data related to yarns, machine settings and fabric characteristics were utilized as inputs to the *Machine Learning (ML)* model. We proposed a novel input combination of features in which we included the sequence of all possible yarns for both warp and weft. The best warp tear strength predictions were achieved at the second CRISP-DM iteration, while the best weft test results were obtained at the third CRISP-DM iteration, which included the final fabric composition as an extra input feature. For the ML task, an *Automated Machine Learning (AutoML)* procedure was used, which allowed us to perform more quickly different CRISP-DM iterations after obtaining feedback from the textile company.

Next, on a second research study, we targeted nine laboratory quality tests and implemented two iterations of the CRISP-DM methodology, with each iteration focused on a feature engineering task, aiming to check the value of different fabric yarn and finishing feature representations. Initially, we utilized the same features that were selected in the previous work, however for this work, we proposed a different yarn representation and a different categorical variable transformation technique. During the modeling stage of CRISP-DM, similarly to the previous work, we utilized an AutoML procedure. In the first CRISP-DM iteration, we compared two yarn code representation approaches and in the second iteration, we selected the best previous yarn code representation approaches to be compared with the new set of features that merged all inputs with the finishing features. The best results were achieved by an input set of features that includes a fixed sequence with a simple yarn code representation and another fixed sequence with fabric finishing operations. The results were shown to the textile company, which considered them valuable to reduce the number of fabric creation attempts, thus having the potential to save production time and costs.

Regarding the optimization task, another R&D Textile project was utilized to implement an initial IDSS that included all the desired *Adaptive Business Intelligence (ABI)* elements proposed for this PhD work, namely predictive models based on ML that were integrated with a prescriptive system based on *Modern Optimization (MO)*. as the one proposed in this PhD work. However, it should be noted that this IDSS targeted a distinct textile industry task, the support of production planning for garment manufacturing An AutoML approach was utilized to address four predictive tasks that are crucial for estimating production planning indicators. Then, we designed a model that uses one of the predicted variables to automatically allocate subcontractors to execute sequential operations associated with a textile order. This MO model was based on the *Non-dominated Sorting Genetic Algorithm II (NSGA-II)* algorithm, assuming a Pareto approach and it was designed to simultaneously minimize the cost and time to execute an order. The results obtained by this approach were considered realistic by the company which signaled that this approach provided a richer set of trade-off solutions, and was faster to compute when compared with the currently adopted manual subcontractor allocation. The performed research provided practical knowledge about the implementation of an IDSS and in particular *Evolutionary Multi-objective Optimization (EMO)* algorithms in the Textile industry.

Finally, we targeted the original PhD goal, the optimization of fabric design features and also the development of the full IDSS itself (with predictive and prescriptive analytics). The IDSS uses predictive analytics to estimate fabric properties and the final fabric composition values and then prescriptive techniques to optimize the best set of fabric construction inputs that feed the predictive models. Two distinct ML predictive approaches were compared: Single-Target Regression, via an AutoML tool, and Multi-Target Regression via a *Artificial Neural Network (ANN)*. For the prescriptive analytics, two EMO methods were compared (NSGA-II and *Reference Point Based Non-dominated Sorting Genetic Algorithm II (R-NSGA-II)*). We assumed two normalized objective functions, set within the [0,1] range, aiming to simultaneously minimize the physical property predictive error and the distance of the optimized values when compared with the learned (thus "know") input space. Overall, the best predictions were obtained by the Single-Target Regression approach, while R-NSGA-II provided the best optimized Pareto curves.

The predictive and prescriptive results were shown to the textile company experts, which provided very positive feedback.

5.2. Discussion

Overall, the proposed IDSS offers a set of valuable functionalities that currently Textile companies do not have. These features bring major improvements to the design of new textile fabrics process. The laboratory quality tests forecasting models allow the textile designer to obtain a quick estimate of the result of a specific test, while previously, it was necessary to produce a prototype of the fabric.

The textile composition is often measured by using a produced fabric prototype, by manually counting the number of weft and warp yarns for each type of fiber contained in a 2.54 centimeter square of fabric, and with the prediction of the final textile composition, the proposed IDSS can provide in real-time this valuable information to both textile designers and customers, with no need to produce a fabric prototype.

Furthermore, the proposed IDSS can potentially lead to multiple forms of interactions with the textile designer. For instance, the designer can inspect the historical data and search for the most similar fabrics, for the desired set of physical properties. Also, the designer can alter some of the components of the selected fabrics, with the IDSS immediately presenting the predicted physical properties and final textile composition. Moreover, the solutions returned by the EMO can be further inspected, allowing the textile designer to select one or more interesting candidate designs for prototype production. Finally, as more prototypes and fabrics are produced and tested, this new data can be used to retrain the predictive models, thus implementing the adaptive aspect of the ABI concept.

Although the results were interesting, and the feedback from the textile company was very positive, several limitations were found throughout this doctoral project. The R&D project work had a duration of 36 months. During the first 12 months, several difficulties become clear. The company did not have a good understanding of the ML and MO true capabilities, thus it was difficult to agree on the business requirements for the IDSS tool. Another difficulty was related to the company *Information System (IS)*. It was developed many years ago and was designed mainly for operational purposes, under an incremental growth. This means that the IS contained its vital data elements spread across different databases, with different quality standards and it was often used differently by different people (e.g., some numeric fields that often include comments). This slowed down the initial R&D execution plan.

Moreover, since this doctoral project had to use the unified data warehouse that was being created for the project, we only obtained the entirety of the data at the beginning of the third year of this PhD work, also slowing down the design and implementation of the predictive models. During the development of the predictive module of the proposed IDSS, we encountered some limitations as only 27 fabrics had the full laboratory quality tests selected by the company, and the number of records for some of the tests was very reduced. Finally, during the development of this IDSS, we encountered some limitations mainly in the evaluation of the IDSS by the textile company, due to the fact that the R&D project had already been finalized when we started the development of the IDSS, which made the evaluation performed by the company to not be as exhaustive as it was for the predictive component of the IDSS. As already explained in Chapter 1, ideally, the proposed IDSS should be validated in a real environment, aiming to measure if its usage would effectively reduce the number of new fabric design attempts (e.g., by employing an A/B testing approach). Since it was not possible to perform such validation, we rather focused on an objective evaluation of the IDSS predictive and prescriptive elements by performing several computational experiments over the company historical data records.

5.3. Future Work

In this section, we highlight several interesting research directions that could be executed in future work.

The first one is, as previously mentioned, the implementation of the IDSS in the Textile company that collaborated on this project. The full integration of the IDSS with a textile design software would enhance not only the system capabilities, but also provide a form to visualize the fabric and patterns. This will also allow to monitor the proposed IDSS in a real environment allowing to confirm its robustness and to obtain valuable feedback from the textile designer to see if is necessary to make adjustments over time. The second research possibility is to expand the predictive component to more laboratory quality tests. During the execution of this doctoral work, we had access to data that contained a total of 15 laboratory tests, but several tests such as Electrostatic properties, were not studied in this work due to the lack of sufficient data records. Since the number of tests is high, this leads to another possibility related to the study of different optimization approaches, such as a many objective approach (Guo, 2022), that would aim to simultaneously minimize the individual predictive error when predicting a larger set of fabric physical properties.

Another interesting possibility is the implementation of an explainable machine learning procedure to demonstrate that valuable knowledge that can be extracted from ML models. For instance, sensitivity analysis and visualization tools (Cortez and Embrechts, 2013) could be used to provide the textile designer with more information.

Finally, we intend to standardize the different methods and components of the IDSS in order to cover the textile design process for different Textile companies. This will allow a Textile company that produces different types of fabrics or perform other types of tests to use the proposed IDSS. We intend to check how general is the proposed approach, whether adjustments need to be made and also if it can provide value to other Textile companies.

REFERENCES

Abakar, K. A. and Yua, C. (2014). Performance of svm based on puk kernel in comparison to svm based on rbf kernel in prediction of yarn tenacity. *Indian Journal of Fibre & Textile Research*, 39:55–59.

Admuthe, L., Apte, S., and Admuthe, S. (2009). Topology and parameter optimization of ann using genetic algorithm for application of textiles. In *2009 IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, pages 278–282. IEEE.

Ahmad, G. G. (2016). Using artificial neural networks with graphical user interface to predict the strength of carded cotton yarns. *The Journal of The Textile Institute*, 107(3):386–394.

Ali, J., Khan, R., Ahmad, N., and Maqsood, I. (2012). Random forests and decision trees. *International Journal of Computer Science Issues (IJCSI)*, 9(5):272.

Anthony, R. (1965). Planning and control systems: A framework for analysis. *Studies in Management Control*.

Anthony Gorry, G. and Scott-Morton, M. (1971). A framework for management information systems,. *Sloan Management Review*, 13.

Arashloo, S. R. and Kittler, J. (2022). Multi-target regression via non-linear output structure learning. *Neurocomputing*, 492:572–580.

Arnott, D. and Pervan, G. (2014). A critical analysis of decision support systems research revisited: the rise of design science. *J. Inf. Technol.*, 29(4):269–293.

Baghdadi, R., Alibi, H., Fayala, F., and Zeng, X. (2016). Investigation on air permeability of finished stretch plain knitted fabrics. i. predicting air permeability using artificial neural networks. *Fibers and Polymers*, 17(12):2105–2115.

Bao, X.-m., Peng, X., Wang, Y.-m., and Cao, Z.-b. (2009). Textile image segmentation based on semi-supervised clustering and bayes decision. In *2009 International Conference on Artificial Intelligence and Computational Intelligence*, volume 3, pages 559–562. IEEE.

Beltran, R., Wang, L., and Wang, X. (2006). Measuring the influence of fibre-to-fabric properties on the pilling of wool fabrics. *Journal of the Textile Institute*, 97(3):197–204.

Bergmann, U., Jetchev, N., and Vollgraf, R. (2017). Learning texture manifolds with the periodic spatial gan. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 469–477. JMLR. org.

Bhatia, N. and Vandana (2010). Survey of nearest neighbor techniques. *CoRR*, abs/1007.0085.

Bi, J. and Bennett, K. P. (2003). Regression error characteristic curves. In Fawcett, T. and Mishra, N., editors, *Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003), August 21-24, 2003, Washington, DC, USA*, pages 43–50, California, USA. AAAI Press.

Bijoya, B. and SB, M. (2005). Comparative analysis of modeling methods for predicting woven fabric properties. *Journal of Textile Engineering*, 51(1):1–9.

Blank, J. and Deb, K. (2020). Pymoo: Multi-objective optimization in python. *IEEE Access*, 8:89497–89509.

Boubaker, J. and Khedher, F. (2015). Evaluation of sewed thread consumption of jean trousers using neural network and regression methods. *Fibres & Textiles in Eastern Europe*, 23:91–96.

Breiman, L. (1996). Stacked regressions. *Machine Learning*, 24(1):49–64.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classification and Regression Trees*. Wadsworth.

Bui, T. X. and Jr., R. H. S., editors (2016). *49th Hawaii International Conference on System Sciences, HICSS 2016, Koloa, HI, USA, January 5-8, 2016*. IEEE Computer Society.

Campos, G. O., Zimek, A., Sander, J., Campello, R. J., Micenková, B., Schubert, E., Assent, I., and Houle, M. E. (2016). On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. *Data Mining and Knowledge Discovery*, 30(4):891–927.

Campos Ciro, G., Dugardin, F., Yalaoui, F., and Kelly, R. (2016). A nsga-ii and nsga-iii comparison for solving an open shop scheduling problem with resource constraints. *IFAC-PapersOnLine*, 49(12):1272–1277. 8th IFAC Conference on Manufacturing Modelling, Management and Control MIM 2016.

Caruana, R. and Niculescu-Mizil, A. (2004). Data mining in metric space: an empirical analysis of supervised learning performance criteria. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 69–78.

Chakrabarti, S., Ester, M., Fayyad, U., Gehrke, J., Han, J., Morishita, S., Piatetsky-Shapiro, G., and Wang, W. (2006). Data mining curriculum: A proposal (version 1.0). *Intensive Working Group of ACM SIGKDD Curriculum Committee*, 140.

Chakraborty, S. and Diyaley, S. (2018). Multi-objective optimization of yarn characteristics using evolutionary algorithms: A comparative study. *Journal of The Institution of Engineers (India): Series E*, 2(99):129–140.

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., Wirth, R., et al. (2000). Crisp-dm 1.0: Step-by-step data mining guide. *SPSS inc*, 9:1–73.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.

Chen, T., Zhang, C., Chen, X., and Li, L. (2009). An input variable selection method for the artificial neural network of shear stiffness of worsted fabrics. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 1(5):287–295.

Chiandussi, G., Codegone, M., Ferrero, S., and Varesio, F. (2012). Comparison of multi-objective optimization methodologies for engineering applications. *Computers & Mathematics with Applications*, 63(5):912–942.

Chicco, D., Warrens, M. J., and Jurman, G. (2021). The coefficient of determination r-squared is more informative than smape, mae, mape, mse and rmse in regression analysis evaluation. *PeerJ Computer Science*, 7:e623.

Chollet, F. (2021). *Deep learning with Python*. Simon and Schuster, NY, USA, 2nd edition.

Cook, D. (2016). *Practical machine learning with H2O: powerful, scalable techniques for deep learning and AI*. "O'Reilly Media, Inc.".

Cortez, P. (2010). Data mining with neural networks and support vector machines using the r/rminer tool. In *Industrial Conference on Data Mining*, pages 572–583. Springer.

Cortez, P. (2021). *Modern Optimization with R*. Springer.

Cortez, P. and Embrechts, M. J. (2013). Using sensitivity analysis and visualization techniques to open black box data mining models. *Information Sciences*, 225:1–17.

Curley, R. (2019). EncyclopÆdia brittanica.

Das, S., Ghosh, A., and Banerjee, D. (2014). Designing of engineered fabrics using particle swarm optimization. *International Journal of Clothing Science and Technology*, 26(1):48–57.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197.

Deb, K. and Sundar, J. (2006). Reference point based multi-objective optimization using evolutionary algorithms. In *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, GECCO '06, page 635–642, New York, NY, USA. Association for Computing Machinery.

Derigs, U. (2009). *Optimization and Operations Research–Volume IV*. EOLSS Publications.

Dhaenens, C. and Jourdan, L. (2016). *Metaheuristics for big data*. John Wiley & Sons.

Dietterich, T. G. (1997). Machine-learning research. *AI Magazine*, 18(4):97–136.

Drath, R. and Horch, A. (2014). Industrie 4.0: Hit or hype?[industry forum]. *IEEE industrial electronics magazine*, 8(2):56–58.

Du, W. and Zhan, Z. (2002). Building decision tree classifier on private data. In *Proceedings of the IEEE international conference on Privacy, security and data mining-Volume 14*, pages 1–8. Australian Computer Society, Inc.

El-Ghezal Jeguirim, S., Sahnoun, M., Babay Dhouib, A., Cheickrouhou, M., Schacher, L., and Adolphe, D. (2011). Predicting compression and surfaces properties of knits using fuzzy logic and neural networks techniques. *International Journal of Clothing Science and Technology*, 23:294–309.

Elahi, I., Ali, H., Asif, M., Iqbal, K., Ghadi, Y., and Alabdulkreem, E. (2022). An evolutionary algorithm for multi-objective optimization of freshwater consumption in textile dyeing industry. *PeerJ Computer Science*, 8:e932.

Eldessouki, M., Hassan, M., Qashqary, K., and Shady, E. (2014). Application of principal component analysis to boost the performance of an automated fabric fault detector and classifier. *Fibres & Textiles in Eastern Europe*.

Eltayib, H. E., Ali, A. H., and Ishag, I. A. (2016). The prediction of tear strength of plain weave fabric using linear regression models. *International Journal of Advanced Engineering Research and Science*, 3(11).

Erickson, N., Mueller, J., Shirkov, A., Zhang, H., Larroy, P., Li, M., and Smola, A. J. (2020). Autogluon-tabular: Robust and accurate automl for structured data. *CoRR*, abs/2003.06505.

Eriksson, T., Bigi, A., and Bonera, M. (2020). Think with me, or think for me? on the future role of artificial intelligence in marketing strategy formulation. *The TQM Journal*, 32(4):795–814.

Ertugrul, S. and Ucar, N. (2000). Predicting bursting strength of cotton plain knitted fabrics using intelligent techniques. *Textile Research Journal*, 70(10):845–851.

Fan, J. and Hunter, L. (1998). A worsted fabric expert system: Part ii: An artificial neural network model for predicting the properties of worsted fabrics. *Textile Research Journal*, 68(10):763–771.

Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. (1996a). From data mining to knowledge discovery in databases. *AI Magazine*, 17(3):37–54.

Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R., editors (1996b). *Advances in Knowledge Discovery and Data Mining*. AAAI/MIT Press.

Ferreira, L., Pilastri, A., Martins, C., Santos, P., and Cortez, P. (2020). An Automated and Distributed Machine Learning Framework for Telecommunications Risk Management. In *Proceedings of the 12th International Conference on Agents and Artificial Intelligence, ICAART 2020, Volume 2, Valletta, Malta, February*, pages 99–107. SciTePress.

Ferreira, L., Pilastri, A. L., Martins, C. M., Pires, P. M., and Cortez, P. (2021). A comparison of automl tools for machine learning, deep learning and xgboost. In *International Joint Conference on Neural Networks, IJCNN 2021, Shenzhen, China, July 18-22, 2021*, pages 1–8, NJ, USA. IEEE.

Ferro, R., Cordeiro, G. A., Ordóñez, R. E. C., Beydoun, G., and Shukla, N. (2021). An optimization tool for production planning: A case study in a textile industry. *Applied Sciences*, 11(18):8312.

Feurer, M., Klein, A., Eggensperger, K., Springenberg, J., Blum, M., and Hutter, F. (2015). Efficient and robust automated machine learning. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Fulcher, J. (2008). Computational intelligence: an introduction. In *Computational intelligence: a compendium*, pages 3–78. Springer.

Furferi, R. and Carfagni, M. (2010). Colour mixing modelling and simulation: optimization of colour recipe for carded fibres. *Modelling and Simulation in Engineering*, 2010:6.

Furferi, R., Governi, L., and Volpe, Y. (2012). Modelling and simulation of an innovative fabric coating process using artificial neural networks. *Textile Research Journal*, 82(12):1282–1294.

Geurts, P., Ernst, D., and Wehenkel, L. (2006). Extremely randomized trees. *Machine learning*, 63(1):3–42.

Ghosh, A., Das, S., and Banerjee, D. (2013). Multi objective optimization of yarn quality and fibre quality using evolutionary algorithm. *Journal of The Institution of Engineers (India): Series E*, 94(1):15–21.

Ghosh, A., Guha, T., Bhar, R., and Das, S. (2011). Pattern classification of fabric defects using support vector machines. *International Journal of Clothing Science and Technology*, 23(2/3):142–151.

Gloy, Y.-S., Renkens, W., Herty, M., and Gries, T. (2015). Simulation and optimisation of warp tension in the weaving process. *Journal of Textile Science & Engineering*, 5:1000179.

Goodfellow, I., Bengio, Y., and Courville, A. (2016a). *Deep Learning*. MIT Press.

Goodfellow, I. J., Bengio, Y., and Courville, A. C. (2016b). *Deep Learning*. Adaptive computation and machine learning. MIT Press, MA, USA.

Guo, X. (2022). A survey of decomposition based evolutionary algorithms for many-objective optimization problems. *IEEE Access*, 10:72825–72838.

Hadizadeh, M., Jeddi, A. A., and Tehran, M. A. (2009). The prediction of initial load-extension behavior of woven fabrics using artificial neural network. *Textile Research Journal*, 79(17):1599–1609.

Hall, M. E. (2000). Finishing of technical textiles. *Handbook of technical textiles*, page 152.

Han, J., Kamber, M., and Pei, J. (2011). *Data Mining: Concepts and Techniques, 3rd edition*. Morgan Kaufmann.

Harel, O. (2009). The estimation of r 2 and adjusted r 2 in incomplete data sets using multiple imputation. *Journal of Applied Statistics*, 36(10):1109–1118.

Hasnat, Ghosh, Khatun, and Halder (2017). Pattern classification of fabric defects using a probabilistic neural network and its hardware implementation using the field programmable gate array system. *Fibres & Textiles in Eastern Europe*.

Hastie, T., Tibshirani, R., and Friedman, J. H. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd Edition*. Springer Series in Statistics. Springer, NY, USA.

He, Y., Lin, J., Liu, Z., Wang, H., Li, L., and Han, S. (2018). AMC: automl for model compression and acceleration on mobile devices. In Ferrari, V., Hebert, M., Sminchisescu, C., and Weiss, Y., editors, *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VII*, volume 11211 of *Lecture Notes in Computer Science*, pages 815–832. Springer.

Hollander, M., Wolfe, D. A., and Chicken, E. (2013). *Nonparametric statistical methods*. John Wiley & Sons, NJ, USA.

Hu, J. (2008). 1 - introduction to fabric testing. In Hu, J., editor, *Fabric Testing*, Woodhead Publishing Series in Textiles, pages 1–26. Woodhead Publishing, Cambridge.

Hu, Z.-H. (2009). A hybrid system based on neural network and immune co-evolutionary algorithm for garment pattern design optimization. *JCP*, 4(11):1151–1158.

Hutter, F., Kotthoff, L., and Vanschoren, J., editors (2019). *Automated Machine Learning - Methods, Systems, Challenges*. The Springer Series on Challenges in Machine Learning. Springer.

Huynh, N. and Chien, C. (2018). A hybrid multi-subpopulation genetic algorithm for textile batch dyeing scheduling and an empirical study. *Comput. Ind. Eng.*, 125:615–627.

Jaouachi, B. and Khedher, F. (2021). Assessment of jeans sewing thread consumption by applying metaheuristic optimization methods. *International Journal of Clothing Science and Technology*, 34(3):347–366.

Jesson, J., Matheson, L., and Lacey, F. M. (2011). *Doing your literature review: Traditional and systematic techniques*. Sage.

Jetchev, N., Bergmann, U., and Vollgraf, R. (2016). Texture synthesis with spatial generative adversarial networks. *arXiv preprint arXiv:1611.08207*.

Junior, A. U., de Freitas Filho, P. J., and Silveira, R. A. (2015). E-HIPS: an extention of the framework HIPS for stagger of distributed process in production systems based on multiagent systems and memetic algorithms. In Sidorov, G. and Galicia-Haro, S. N., editors, *Advances in Artificial Intelligence and Soft Computing - 14th Mexican International Conference on Artificial Intelligence, MICAI 2015, Cuernavaca, Morelos, Mexico, October 25-31, 2015, Proceedings, Part I*, volume 9413 of *Lecture Notes in Computer Science*, pages 413–430, Cham. Springer.

Kalkanci, M., Kurumer, G., Öztürk, H., Sinecen, M., and Kayacan, ▯. (2017). Artificial neural network system for prediction of dimensional properties of cloth in garment manufacturing: Case study on a t-shirt. *Fibres & Textiles in Eastern Europe*, Nr 4 (124):135–140.

Kantardzic, M. (2011). *Data mining: concepts, models, methods, and algorithms*. John Wiley & Sons.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Kotb, N. (2009). Engineering of tearing strength for pile fabrics. *Journal of Textile and Apparel, Technology and Management*, 6(1).

Kularatne, S., Ranawaka, R., Fernando, E., Niles, S., Jayawardane, T., and Ranaweera, R. (2020). Machine learning-based approach for modelling elastic modulus of woven fabrics. In *2020 Moratuwa Engineering Research Conference (MERCon)*, pages 470–475.

Kuo, C.-F. J. and Juang, Y. (2016). A study on the recognition and classification of embroidered textile defects in manufacturing. *Textile Research Journal*, 86(4):393–408.

Landry, M. and Bartz, A. (2022). Machine learning with r and h2o. Technical report:, H2O.ai, Inc.

LeDell, E. and Poirier, S. (2020). H2O AutoML: Scalable automatic machine learning. *7th ICML Workshop on Automated Machine Learning (AutoML)*.

Lee, C., Choy, K. L., Ho, G. T., Chin, K.-S., Law, K., and Tse, Y. K. (2013). A hybrid olap-association rule mining based quality management system for extracting defect patterns in the garment industry. *Expert Systems with Applications*, 40(7):2435–2446.

Liyanage, I., Nuwanga, S., Anjana, R., Rankothge, W., and Gamage, N. (2020). Sustainable manufacturing: Application of optimization to textile manufacturing plants. *Global Journal of Computer Science and Technology*.

Lones, M. A. (2011). Sean luke: essentials of metaheuristics. *Genetic Programming and Evolvable Machines*, 12(3):333–334.

Lorente-Leyva, L. L., Murillo-Valle, J. R., Montero-Santos, Y., Herrera-Granda, I. D., Herrera-Granda, E. P., Rosero-Montalvo, P. D., Peluffo-Ordóñez, D. H., and Blanco-Valencia, X. P. (2019). Optimization of the master production scheduling in a textile industry using genetic algorithm. In Pérez García, H., Sánchez González, L., Castejón Limas, M., Quintián Pardo, H., and Corchado Rodríguez, E., editors, *Hybrid Artificial Intelligent Systems*, pages 674–685, Cham. Springer International Publishing.

Lucke, D., Constantinescu, C., and Westkämper, E. (2008). Smart factory-a step towards the next generation of manufacturing. In *Manufacturing systems and technologies for the new frontier*, pages 115–118. Springer.

Lv, Z., Xiang, Q., and Yang, J. (2011). Application of genetic algorithm-support vector machine for prediction of spinning quality". In *Proceedings of the World Congress on Engineering*, volume 2, pages 6–8.

Majumdar, A., Das, A., Hatua, P., and Ghosh, A. (2016). Optimization of woven fabric parameters for ultraviolet radiation protection and comfort using artificial neural network and genetic algorithm. *Neural Comput. Appl.*, 27(8):2567–2576.

Majumdar, A., Mal, P., Ghosh, A., and Banerjee, D. (2017). Multi-objective optimization of air permeability and thermal conductivity of knitted fabrics with desired ultraviolet protection. *The Journal of The Textile Institute*, 108(1):110–116.

Malik, Z. A., Malik, M. H., Hussain, T., and Arain, F. A. (2011). Development of models to predict tensile strength of cotton woven fabrics. *Journal of engineered fibers and fabrics*, 6(4):155892501100600407.

Matos, L. M., Azevedo, J., Matta, A., Pilastri, A., Cortez, P., and Mendes, R. (2022). Categorical attribute transformation environment (cane): A python module for categorical to numeric data preprocessing. *Software Impacts*, 13:100359.

Matos, L. M., Cortez, P., Mendes, R., and Moreau, A. (2019). Using deep learning for mobile marketing user conversion prediction. In *International Joint Conference on Neural Networks, IJCNN 2019 Budapest, Hungary, July 14-19, 2019*, pages 1–8, NY, USA. IEEE.

Matusiak, M. (2015). Application of artificial neural networks to predict the air permeability of woven fabrics. *Fibres & Textiles in Eastern Europe*, 109(1):41–48.

Michalewicz, Z. and Fogel, D. B. (2013). *How to solve it: modern heuristics*. Springer Science & Business Media.

Michalewicz, Z., Schmidt, M., Michalewicz, M., and Chiriac, C. (2006). *Adaptive business intelligence*. Springer.

Mills, B. R. (2018). A glossary of selected fiber and textile terms.

Mitra, A., Majumdar, P. K., and Banerjee, D. (2015). Production of engineered fabrics using artificial neural network–genetic algorithm hybrid model. *Journal of The Institution of Engineers (India): Series E*, 96(2):159–165.

Mohanty, A. K. and Bag, A. (2017). Detection and classification of fabric defects in textile using image mining and association rule miner. *Int. J. Electr. Electron. Comput*, 2(3):28–33.

Mohri, M., Rostamizadeh, A., and Talwalkar, A. (2012). *Foundations of Machine Learning*. Adaptive computation and machine learning. MIT Press.

Mozafary, V. and Payvandy, P. (2014). Application of data mining technique in predicting worsted spun yarn quality. *The Journal of The Textile Institute*, 105(1):100–108.

Murrells, C. M., Tao, X. M., Xu, B. G., and Cheng, K. P. S. (2009). An artificial neural network model for the prediction of spirality of fully relaxed single jersey fabrics. *Textile Research Journal*, 79(3):227–234.

Natekin, A. and Knoll, A. (2013). Gradient boosting machines, a tutorial. *Frontiers in neurorobotics*, 7:21.

Nelder, J. A. and Wedderburn, R. W. (1972). Generalized linear models. *Journal of the Royal Statistical Society: Series A (General)*, 135(3):370–384.

Ngan, H. Y., Pang, G. K., and Yung, N. H. (2011). Automated fabric defect detection—a review. *Image and Vision Computing*, 29(7):442–458.

North, M. (2012). *Data mining for the masses*. Global Text Project Athens.

Ogulata, S. N., Sahin, C., Ogulata, R. T., and Balci, O. (2006). The prediction of elongation and recovery of woven bi-stretch fabric using artificial neural network and linear regression models. *Fibres & Textiles in Eastern Europe*, 14(2):56.

Ojha, V. K., Abraham, A., and Snásel, V. (2017). Metaheuristic design of feedforward neural networks: A review of two decades of research. *Eng. Appl. of AI*, 60:97–116.

Oliveira, N., Cortez, P., and Areal, N. (2017). The impact of microblogging data for stock market prediction: Using twitter to predict returns, volatility, trading volume and survey sentiment indices. *Expert Syst. Appl.*, 73:125–144.

O'Neill, M. and Ryan, C. (2001). Grammatical evolution. *IEEE Transactions on Evolutionary Computation*, 5(4):349–358.

Pan, R., Gao, W., Liu, J., and Wang, H. (2011). Genetic algorithm-based detection of the layout of color yarns. *The Journal of The Textile Institute*, 102(2):172–179.

Pattanayak, A. K., Luximon, A., and Khandual, A. (2011). Prediction of drape profile of cotton woven fabrics using artificial neural network and multiple regression method. *Textile Research Journal*, 81(6):559–566.

Peffers, K., Tuunanen, T., Rothenberger, M. A., and Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of management information systems*, 24(3):45–77.

Pfrommer, J., Zimmerling, C., Liu, J., Kärger, L., Henning, F., and Beyerer, J. (2018). Optimisation of manufacturing process parameters using deep neural networks as surrogate models. *Procedia CIRP*, 72(1):426–431.

Piotrowski, A. P., Napiorkowski, M. J., Napiorkowski, J. J., and Rowinski, P. M. (2017). Swarm intelligence and evolutionary algorithms: Performance versus speed. *Information Sciences*, 384:34–85.

Rejali, M., Hasani, H., Ajeli, S., and Shanbeh, M. (2014). Optimization and prediction of the pilling performance of weft knitted fabrics produced from wool/acrylic blended yarns. *Indian Journal of Fibre & Textile Research 39, 83-88*, 39.

Ribeiro, R., Pilastri, A., Moura, C., Rodrigues, F., Rocha, R., Morgado, J., and Cortez, P. (2020). Predicting physical properties of woven fabrics via automated machine learning and textile design and finishing features. In Maglogiannis, I., Iliadis, L., and Pimenidis, E., editors, *Artificial Intelligence Applications and Innovations*, pages 244–255, Cham. Springer International Publishing.

Schölkopf, B. and Smola, A. J. (2002). *Learning with Kernels: support vector machines, regularization, optimization, and beyond*. Adaptive computation and machine learning series. MIT Press.

Semnani, D. and Vadood, M. (2010). Improvement of intelligent methods for evaluating the apparent quality of knitted fabrics. *Engineering Applications of Artificial Intelligence*, 23(2):217–221.

Shahid, M., Hossain, M., et al. (2015). Modeling the spirality of cotton knit fabric using fuzzy expert system. *Turkish Journal of Fuzzy Systems*, 6(2):56–67.

Shang, K., Ishibuchi, H., He, L., and Pang, L. M. (2021). A survey on the hypervolume indicator in evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 25(1):1–20.

Shishoo, R. (2012). *The global textile and clothing industry: technological advances and future challenges*. Elsevier.

Simon, H. (1977). *The new science of management decision*. Prentice Hall PTR.

Smola, A. J. and Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222.

Su, T.-L. and Lu, C.-F. (2011). Automated vision system for recognising lycra spandex defects. *Fibres & Textiles in Eastern Europe*, 19(1):43–46.

Sugiyama, M. (2015). *Introduction to statistical machine learning*. Morgan Kaufmann.

Taieb, A. H., Mshali, S., and Sakli, F. (2018). Predicting fabric drapability property by using an artificial neural network. *Journal of Engineered Fibers and Fabrics*, 13(3):87–93.

Teli, M., Khare, A., and Chakrabarti, R. (2008). Dependence of yarn and fabric strength on the structural parameters. *AUTEX Research Journal*, 8(3):63–67.

Thornton, C., Hutter, F., Hoos, H. H., and Leyton-Brown, K. (2012). Auto-weka: Automated selection and hyper-parameter optimization of classification algorithms. *CoRR*, abs/1208.3719.

Tjahjono, B., Esplugues, C., Ares, E., and Pelaez, G. (2017). What does industry 4.0 mean to supply chain? *Procedia Manufacturing*, 13:1175–1182.

Torgo, L. (2011). *Data mining with R: learning with case studies*. Chapman and Hall/CRC.

Truong, A., Walters, A., Goodsitt, J., Hines, K. E., Bruss, C. B., and Farivar, R. (2019). Towards automated machine learning: Evaluation and comparison of automl approaches and tools. In *31st IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2019, Portland, OR, USA, November 4-6, 2019*, pages 1471–1479. IEEE.

Tsao, Y.-C., Hung, C.-H., and Vu, T.-L. (2021). Hybrid heuristics for marker planning in the apparel industry. *Arabian Journal for Science and Engineering*, 46(10):10077–10096.

Turban, E., Sharda, R., Aronson, J. E., and King, D. (2008). *Business intelligence: A managerial approach*. Pearson Prentice Hall Upper Saddle River, NJ.

Vermesan, O., Friess, P., Guillemin, P., Gusmeroli, S., Sundmaeker, H., Bassi, A., Jubert, I. S., Mazura, M., Harrison, M., Eisenhauer, M., et al. (2011). Internet of things strategic research roadmap. *Internet of things-global technological and societal trends*, 1(2011):9–52.

Vikhar, P. A. (2016). Evolutionary algorithms: A critical review and its future prospects. In *2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC)*, pages 261–265. IEEE.

Wadje, P. (2009). Textile–fibre to fabric processing. *Journal of the Institution of Engineers (India), Part TX: Textile Engineering Division*, 90:28.

Webster, B. and Bernhard, P. J. (2003). A local search optimization algorithm based on natural principles of gravitation. Technical report, Florida Institute of Technology.

While, L., Hingston, P., Barone, L., and Huband, S. (2006). A faster algorithm for calculating hypervolume. *IEEE Trans. Evol. Comput.*, 10(1):29–38.

Willmott, C. J. and Matsuura, K. (2005). Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. *Climate Research*, 30(1):79–82.

Wirth, R. and Hipp, J. (2000). Crisp-dm: Towards a standard process model for data mining. In *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*, pages 29–39. Citeseer.

Witten, I. H., Frank, E., Hall, M. A., and Pal, C. J. (2016). *Data Mining: Practical Machine Learning Tools and Techniques*. Elsevier Inc.

Wu, X., Kumar, V., Ross Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Ng, A., Liu, B., Yu, P. S., Zhou, Z., Steinbach, M., Hand, D. J., and Steinberg, D. (2008). Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14(1):1–37.

Xu, Y., Sébastien, T., and Xianyi, Z. (2020). Optimization of garment sizing and cutting order planning in the context of mass customization. *The International Journal of Advanced Manufacturing Technology*, 106(7-8):3485–3503.

Xue, Z., Zeng, X., and Koehl, L. (2017). Development of a method based on fuzzy comprehensive evaluation and genetic algorithm to study relations between tactile properties and total preference of textile products. *The Journal of The Textile Institute*, 108(7):1085–1094.

Yap, P. H., Wang, X., Wang, L., and Ong, K.-L. (2010). Prediction of wool knitwear pilling propensity using support vector machines. *Textile Research Journal*, 80(1):77–83.

Yildirim, P., Birant, D., and Alpyildiz, T. (2018). Data mining and machine learning in textile industry. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, 8(1).

Yildiz, Z., DAL, V., Ünal, M., and Yildiz, K. (2013). Use of artificial neural networks for modelling of seam strength and elongation at break. *Fibres & Textiles in Eastern Europe*, 101:117–123.

Yuen, C., Wong, W. K., Qian, S., Chan, L., and Fung, E. (2009). A hybrid model using genetic algorithm and neural network for classifying garment defects. *Expert Systems with Applications*, 36(2):2037–2047.

Zeydan, M. (2010). Prediction of fabric tensile strength by modelling the woven fabric. *Woven Fabric Engineering*, page 155.

Zhang, N., Pan, R., Wang, L., Wu, Y., and Gao, W. (2020). Pattern design and optimization of yarn-dyed plaid fabric using modified interactive genetic algorithm. *The Journal of The Textile Institute*, 111(11):1652–1661.

Zhang, R., Chang, P.-C., Song, S., and Wu, C. (2017). A multi-objective artificial bee colony algorithm for parallel batch-processing machine scheduling in fabric dyeing processes. *Knowledge-Based Systems*, 116:114–129.

Zhang, Z., Guo, C., Wei, Q., Guo, Z., and Gao, L. (2021). A bi-objective stochastic order planning problem in make-to-order multi-site textile manufacturing. *Comput. Ind. Eng.*, 158:107367.