# A data-driven intelligent decision support system that combines predictive and prescriptive analytics for the design of new textile fabrics

Rui Ribeiro<sup>1,4</sup>, André Pilastri<sup>1</sup>, Carla Moura<sup>2</sup>, José Morgado<sup>3</sup> and Paulo Cortez<sup>4\*</sup>

 <sup>1</sup>EPMQ - IT, CCG ZGDV Institute, University of Minho -Campus of Azurém, Guimarães, 4800-058, Portugal.
 <sup>2</sup>Riopele, Pousada de Saramagos, 4770-405, Portugal.
 <sup>3</sup>CITEVE, V.N. Famalicão, 4760-034, Portugal.
 <sup>4</sup>ALGORITMI/LASI, University of Minho, Guimarães, 4804-533, Portugal.

\*Corresponding author(s). E-mail(s): pcortez@dsi.uminho.pt; Contributing authors: rui.ribeiro@ccg.pt;

#### Abstract

In this paper, we propose an Intelligent Decision Support System (IDSS) for the design of new textile fabrics. The IDSS uses predictive analytics to estimate fabric properties (e.g., elasticity) and composition values (% cotton) and then prescriptive techniques to optimize the fabric design inputs that feed the predictive models (e.g., types of yarns used). Using thousands of data records from a Portuguese textile company, we compared two distinct Machine Learning (ML) predictive approaches: Single-Target Regression (STR), via an Automated ML (AutoML) tool, and Multi-Target Regression (MTR), via a deep learning Artificial Neural Network (ANN). For the prescriptive analytics, we compared two Evolutionary Multi-objective Optimization (EMO) methods (NSGA-II and R-NSGA-II) when optimizing 100 new fabrics, aiming to simultaneously minimize the physical property predictive error and the distance of the optimized values when compared with the learned input space. The two EMO methods were applied to design of 100 new fabrics. Overall, the STR approach provided the best results for both prediction tasks,

with Normalized Mean Absolute Error (NMAE) values that range from 4.05% (weft elasticity) to 11.22% (pilling) in terms of the fabric properties and a textile composition classification accuracy of 87% when adopting a small tolerance of 0.01 for predicting the percentages of six types of fibers (e.g., cotton). As for the prescriptive results, they favored the R-NSGA-II EMO method, which tends to select Pareto curves that are associated with an average 11% predictive error and 16% distance.

**Keywords:** Textile development, Predictive, Regression, Machine Learning, Evolutionary Multi-objective Optimization.

# 1 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 Fabric Design (FD) process, often termed as fabric engineering, begins with the definition of a new prototype design, which includes several 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 design 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 [1]. Each time a prototype is produced, laboratory quality tests are required, to assure regulatory compliance of the physical properties [2]. 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 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 Intelligent Decision Support System (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 [3].

As shown in Section 2, there are few research works that address the FD task. In particular, there are only two studies that combine predictive and prescriptive analytics for FD [4, 5] and these studies contain several limitations, such as: optimization of a reduced set of fabric design features and number of designed fabrics; usage of a single ML algorithm; and use of a single objective optimization. In a preliminary previous work [6], we have obtained some interesting fabric property prediction results by using an Automated ML (AutoML). However, we did not address two important FD tasks that were requested by the Portuguese textile company: the prediction of the final textile composition (e.g., % of cotton) and the search for the best set of FD features. Aiming to provide a full FD solution, in this work we propose a novel data-driven IDSS that is based on the Adaptive Business Intelligence (ABI) [7] 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, see Section 2), 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 thousands of fabric creation records, are used to adapt and compare two ML main approaches: Single-Target Regression (STR), based on an AutoML tool [8]; and Multi-Target Regression (MTR) [9], based on a deep Multilayer Perceptron Artificial Neural Network (ANN) [10].

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 [11]: NSGA-II and 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.

The rest of the paper is organized as follows. Section 2 presents the related work. Section 3 describes the fabric design task. Section 4 presents the textile data and the proposed IDSS. Section 5 details the experiments conducted and the obtained results. Finally, Section 6 discusses the main conclusions and also presents future research directions.

## 2 Related Work

In the textile industry, even for the production of basic fabrics or garments, a large amount of data is typically created and stored (e.g., yarn characteristics, machine settings and quality tests). In recent years, the increase in the volume of data being stored has enabled AI tools to enhance textile production processes. In effect, several AI methods were applied successfully to the industry, with different approaches being employed to optimize different textile stages. Table 1 summarizes the most relevant related works, assuming a chronological order and the following characterizing elements: **TS** - the textile stage addressed by the study, **Data** - the type of data used, **Pred.** - if used, the type of predictive ML approach, **MH** - the type of metaheuristic used, **Obj.** - a description of the optimization objective, and **M** - the number of objective functions used by the metaheuristic (if M > 2 then the study assumes a Multi-Objective approach).

Most of the analyzed studies are focused on the manufacturing of fabrics or garments (as shown in column **TS** of Table 1). There are only four related works that directly address the FD process. When compared with our approach (last row of Table 1), the FD studies only optimize a reduced set of fabric features (F), ranging from four to six inputs. For example, Majumdar et al. (2017) [5] only optimized the loop length, carriage speed, yarn input tension and yarn count. In contrast, our work assumes a richer and more complex set of F=44input parameters (see Section 4.1), reflecting diverse elements such as machine settings, warp and weft yarns and finishing operations. Within our knowledge, this work is the first to use finishing operations data (e.g., dyeing, washing, drying). Moreover, our study targets four desired physical properties (P), which is a higher number when compared with the related works that range from one [13] to three targets [4, 5, 16]. In addition, only two of the related FD works assumed a combination of predictive and prescriptive analytics [4, 16]. Regarding the number of optimized fabrics (V), the FD works optimize a reduced number of new fabrics, ranging from 1 to 4. In contrast, this work optimizes a substantially higher number of 100 new fabrics. Furthermore, none of the related works with predictive approaches (column **Pred** from Table 1) have modeled the final textile composition (e.g., % of cotton, % of polyester) based on the fabric design features, which is addressed in our work. Regarding the two FD studies that performed a prediction of the physical properties, they used only one ML algorithm, a simpler ANN structure composed of a Multilayer Perceptron with only one hidden layer. In a previous work [6], we have obtained interesting fabric property prediction results by using an AutoML

5

Study	тѕ	Data	Pred	МН	Obj	M
Ghosh et al. [12]	Mf	F,Y	ANN	NSGA-II	RMQ,YS	2
Das et al. [13]	FD(6,1,1)	FC	_	PSO	GSM	1
Gloy et al. [14]	Mf	MS	_	GA	WMP	1
Junior et al. [15]	Mf	YDO,PR	_	GA,MA	YDS	1
Mitra et al. [16]	FD(4,3,2)	FC, FP	ANN	GA	EPI,PPI, WaC,WeC	1
Majumdar et al. [4]	FD(5,3,4)	FC,FP	ANN	GA	CP,UPF	1
Majumdar et al. [5]	FD(4,3,2)	MS,FC	_	NSGA-II	AP,ThC	2
Zhang et al. [17]	Mg	DO,PR	_	MO-ABC, NSGA-II	TC, URDV	2
Chakraborty et al. [18]	Mf	FC,MS	-	ABC,ACO NSGA-II,PSO	RiSP,RoSP	2
Huynh et al. [19]	Mf	DO,PR	_	MSGA-H	STBD	1
Lorente-Leyva et al [20]	Mg	I,PĆ, PO,SF	-	GA	MPS	1
Liyanage et al. [21]	Mf	PO,PR,TR	_	GP	FW,QAW,SW	1
Xu et al. [22]	Mg	CO,PR,FC	_	GA	COP	1
Jaouachi et al. [23]	Mg	MS,OP	_	ACO,GA,PSO	STC	1
Ferro et al. [24]	Mf	PO,PR,ST	_	GA	PPWP	1
Ribeiro et al. [25]	Mg	PO,SR	AutoML	NSGA-II	GPC,GPT	<b>2</b>
Tsao et al. [26]	Mg	FCp	_	GA,SA, HGASA	ML	1
Zhang et al. [27]	Mf	DO,PiO,PR	_	pesgMESO	CE,TT	<b>2</b>
Elahi et al. [28]	Mf	DO,WC,EC	_	MOGCO-II	WC,PP,WTC	3
This work	FD(44,4,100)	FC,FO, FP,MS,Y	AutoML, deep ANN	NSGA-II, R-NSGA-II	FDF	2

 Table 1
 Summary of the related work.

Acronyms used in the table are (see also Appendix A):

 ${}^{a}FD(F,P,V)$  - Fabric Design  $(F, P \text{ and } V \text{ denote the number of: input features, physical properties that are targeted and validation fabric samples); <math>Mf$  - Manufacturing of fabrics; Mg - Manufacturing of Garments.

 $^{b}CO$  - Cutting Orders; DO - Dyeing Orders; EC - Energy Consumption; F - Fibers; FC - Fabric Construction parameters; FCp - Fabric Cutting patterns; FO - Finishing Operations; FP - Fabric Properties; MS - Machine Settings; OP - Operation Parameters; PC - Production Capacities; PO - Production Orders; PiO - Printing Orders; PR - Productive Resources; SF - Sales Forecast; SR - Subcontractor Resources; ST - Setup Times; TR - Transportation Resources; WC - Water Consumption; Y - Yarns; YDO - Yarn Dyeing Orders.

<sup>c</sup>ANN - Artificial Neural Network; AutoML - Automated Machine Learning.

 $^dABC$  - Artificial Bee Colony; ACO - Ant Colony Optimization; GA - Genetic Algorithm; GP - Genetic Programming; HGASA - Hybrid Genetic Algorithm Simulated Annealing; MA - Memetic Algorithm; MO-ABC - Multi-Objective Artificial Bee Colony; MOGCO-II - Multi-Objective Group Counseling Optimizer II; MSGA-H - Multi-Subpopulation Genetic Algorithm Heuristics; NSGA-II - Non-dominated Sorting Genetic Algorithm II; pesgMESO - parallel evolution scenario generation-based Multi-objective Evolutionary Stochastic Optimization; PSO - Particle Swarm Optimization; R-NSGA-II - Reference point based Non-dominated Sorting Genetic Algorithm II; SA - Simulated Annealing.

 $^eAP$  - Air Permeability; CE - Carbon Emission; CP - Comfort Properties; COP - Cutting Order Planning: EPI - Ends per Inch, FC - Freshwater Consumption; FDF - Fabric Design Features; FW - Finishing Workflow; GPC - Garment Production Cost; GPT - Garment Production Time; GSM - fabric Grams per Square Meter, MPS - Master Production Scheduling; ML - Marker Layout; PP - Production Planning; PPI - Picks per Inch; PPWP - Production Planning of Weaving Process; QAW - Quality Assurance Workflow; RMQ - Raw Material Quality; RiSP - Ring Spinning Process; RoSP - Rotor Spinning Process; STBD - Scheduling Textile Batch Dyeing; STC - Sewing Thread Consumption; SW - Sewing Workflow; TC - Tardiness cost, ThC - Thermal Conductivity; TT - Total Tardiness; UPF - Ultraviolet Protection Factor; URDV - Utilization Rate of Dyeing Vats; WaC - Warp Count; WeC - Weft Count; WMP - Weaving Machine Parameters; WTC - Wastewater Treatment Costs; YDS - Yarn Dyeing Scheduling; YS - Yarn Strength.

STR modeling. In this research, we compare this AutoML STR approach with an alternative MTR based on a deep learning ANN. Finally, while all four FD studies have employed metaheuristics in order to optimize the fabric design elements, only one work adopted a EMO approach [4], via the NSGA-II method. In this paper, we propose a novel multi-objective approach that simultaneously minimizes the physical properties predictive error and also the distance of the selected inputs when compared with the learned input space. To perform the multi-objective task, we compare two EMO algorithms, NSGA-II and R-NSGA-II.

### **3** Problem Statement

This work was carried out with collaboration of a Portuguese textile company that produces high-quality fabrics, based on natural, synthetic, artificial and recycled fibers, focusing on 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 1 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.

Under this context, is essential to reduce the number of fabric development cycles. This work proposes a data-driven IDSS approach to solve this task, which first uses ML to model the fabric physical properties and final textile composition, and then performs an EMO to select the best set of fabric design values based on the previously obtained ML models. This is a nontrivial task due to several reasons. 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 design 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



Fig. 1 Flowchart for the development process of a new fabric (Y and N denote Yes and No, respectively).

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 manually counting of the number of weft and warp yarns for each fiber type contained in a fabric square of 2.54 centimeters.

# 4 Material and Methods

### 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, such as yarns, finishing operations, machine settings and identification codes, and the laboratory testing database, which contained the fabric quality tests performed between 2012 and 2019. We implemented then an Extraction, Transformation, 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

7

relevant attributes by removing irrelevant data (e.g., designer and customer identification codes) and data with a low variance (thus almost constant), such as the elasticity type of the fabric. Since the number of potential attributes was still high, we then performed several feature selection iterations by using domain knowledge (retrieved from textile experts) to discard more irrelevant features (e.g., color, falling weight and rapport), 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 [6]. 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 2 summarizes the adopted set of fabric attributes in terms of their **Attribute** name, **Description** and **Range** of the domain values. The first

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
ор	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]
СО	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]

Table 2List of fabric design features.

13 rows of Table 2 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 2 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 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 slight 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 2. Finally, the last 6 rows Table 2 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 multiclass (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 2 exemplifies some of the design features related to real textile fabrics. The left of Figure 2 presents a yarn composed of four wool fibers. The middle of Figure 2 presents a fabric where it is possible to distinguish between the yarns that compose the weft (horizontal yarns) and the ones that compose the warp (vertical yarns). The right of Figure 2 presents a wool fabric with a higher number of reed dents per centimeter making the spaces between the weft and warp yarns much narrower.



Fig. 2 Example of a wool yarn (left) and two different textile fabrics (middle and right).

Figure 3 presents two different components of a loom machine. In the left of Figure 3, the coils contain the warp yarns used to produce the fabric. The right of Figure 3 presents the loom that will raise alternate warp yarns to create a shed using a harness, which has a series of wires called heddles and the warp yarns pass through the heddle eyelets. The weft yarn is inserted through the shed by a carrier device. A single crossing of the filling from one side of the loom to the other is called a pick. The reed pushes each weft yarn against the portion of the fabric that has already been formed, resulting in a firm and compact fabric.



Fig. 3 Example of two components of a textile machine, namely yarn coils (left) and a loom machine (right).

Figure 4 further 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. The left of Figure 4 presents a view of a fabric that contains interlaced yarns related with the warp (blue color) and weft (gray 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 is 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 varn of the fabric is measured. The top right of Figure 4 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 4 presents a representation of a loom machine, where the weft yarns (gray 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 2 (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 [6], we



A data-driven intelligent decision support system

Fig. 4 Visualization of some woven fabric and machine features.

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:  $6 \times 2$  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 2 (e.g., finished width, weave design). We note that in [6] several ML feature selection experiments were executed but worst predictive results were obtained when removing some of the 44 fabric input elements (e.g., finishing operations).

In terms of data preprocessing, the three categorical attributes in Table 2 (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 [29]. 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 [30]:

$$IDF(x_i) = ln(\frac{n}{f_l}) \tag{1}$$

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 [31]. 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.

### 4.2 Intelligent Decision Support System

The proposed IDSS consists of three main modules (Figure 5): 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.

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, an 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 4.3.

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



Fig. 5 Flow diagram describing the components of the proposed IDSS.

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 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, 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 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 4.3).

After selecting the best ML approaches, we then evaluate the EMO methods (Section 4.4).

The proposed IDSS can be integrated into the textile information system by deploying it in the company computer server. If an optimized IDSS fabric design is accepted by the textile designer then a textile production order is created. Currently, any production order (manually designed or IDSS assisted) involves both automated and manual machine settings. For instance, the warp yarn coils (left of Figure 3) are manually inserted but once the yarns are fed into the loom (right of Figure 3), the fabric production process is automated. As mentioned in Section 1, we note that the main IDSS goal is to reduce the number of fabric prototype development attempts and not the fabric production order execution time. Nevertheless, in the future it is possible to further optimize the production order execution time by adopting more Industry 4.0 components (e.g., usage of automated and digitally connected machines).

### 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 2). Under this context, performing a MTR can be interesting modeling approach, since it requires a single ML model to simultaneously model several equally important targets [9]. 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 a human effort. In this work, we adopt the H2O AutoML tool [32], which achieved good results in a recent AutoML benchmark study [8]. Since the H20 AutoML only performs a STR, the H20 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 [33]. Table 3 summarizes the main characteristics of the ANNs implemented for the MTR experiments. The ANN consists of a fully connected (thus dense) multilayer perceptrons [10]. 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. In this work, the ANNs assume a total of I = 44 fabric design inputs (see Section 4.1), while the number of outputs depends on the number of MTR tasks (O = 4 for the four physical properties; and O=6 for the six textile composition percentages). Some preliminary experiments were held to set the ANN structure (in terms of number of hidden layers and their layer sizes) using older textile data and triangular-shaped multilayer perceptrons, in which each subsequent layer size is smaller [34]. 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) (h=3 hidden layers); and second ML goal (textile composition) - (44, 18, 12, 6) (h=2 hidden layers). The popular ADAM optimizer was used to adjust the ANN weights during the training phase [35], assuming the Mean Absolute Error (MAE) loss function, an early stopping (with 10% of the training data being used as the validation set) and maximum of 1,000 epochs.

ML goal	Activation Function	Structure	Loss Function	Optimizer Function	Max. epochs
Physical Property	ReLu	(44, 28, 20, 12, 4)	MAE	ADAM	1000
Textile Composition	ReLu	(44, 18, 12, 6)	MAE	ADAM	1000

Table 3 Summary of the main characteristics of the ANN MTR models.

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. We used the same ML search setup adopted in [8], which provided competitive results for the H20 tool when compared with seven other AutoML frameworks (e.g., TPOT, AutoGluon). Thus, a total of six different ML algorithms are automatically searched by the AutoML tool for each regression task: Random Forests (RF), Extremely Randomized Trees (XRT), Generalized Linear Models (GLM), Gradient Boosting Machines (GBM), XGBoost (XG) and Stacked Ensembles (SE). The H20 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 secondlevel 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 [29] was implemented, which is a standard ML evaluation procedure (e.g., used in [36]). The regression quality was assessed by using the MAE and the Normalized MAE (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 [37]:

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

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

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:

$$\hat{y}'_{i,j} = \frac{\hat{y}_{i,j}}{\sum_{k=1}^{K} (\hat{y}_{k,j})}$$
(3)

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:

$$\hat{y}_{i,j}' = \frac{\exp(\hat{y}_{i,j})}{\sum_{k=1}^{K} (\exp(\hat{y}_{k,j}))}$$
(4)

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 [38]. In this work, we selected 10 small tolerance values within the range  $T \in \{0.01, 0.02, ..., 0.10\}$ .

### 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 4.1), 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 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  $\hat{\mathcal{Y}}_i(s)$  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:

$$f_p(s) = \frac{\sum_{i=1}^{4} |\mathcal{Y}_i(s) - \hat{\mathcal{Y}}_i(s)|}{4} \quad \text{(predictive error)} \\ f_d(s) = \frac{\arg\min_{x \in \mathcal{D}} (\sqrt{(x-s)^2})}{\sqrt{I}} \quad \text{(distance)}$$
(5)

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 1, 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 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 [39]. The Non-dominated Sorting Genetic Algorithm (NSGA)-II, adopts several distinct features (e.g., elitist strategy, fast crowded distance estimation procedure) and it is considered a parameterless approach [40]. 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 [41]. As for R-NSGA-II, it consists of a more recent NSGA

variant that assumes a modified survival selection, which is based on one or more user defined reference points [42].

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) [43]. 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 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. We highlight that each EMO method is executed for V=100 times (one run for each new targeted fabric), which is a number that is substantially higher when compared with the state-of-the-art FD works (see Section 2).

## 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 a Intel Core i7 2.20GHz processor, with 6 cores, a NVIDIA GeForce GTX 1050 Ti, using a Windows operating system.

### 5.1 Physical Property Prediction Results

Table 4 presents the physical property predictive performance results. For each predicted task (**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.

For demonstration purposes, Figure 6 shows the Regression Error Characteristic (REC) curves for the four targets and the  $6^{th}$  external k-fold iteration. Each REC curve plots the percentage of correctly predicted examples (y-axis) for a given absolute error tolerance (x-axis) [38]. The curves confirm that the **STR** presents slight 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

	$\mathbf{STR}$		Μ	ITR
Property	MAE	NMAE	MAE	NMAE
Bias Distortion	0.555±0.199	<b>5.92</b> %±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	$0.350 \pm 0.008^*$	$11.22\% \pm 0.306^{*}$	$0.597 {\pm} 0.046$	$19.13\% \pm 1.648$

Table 4 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**).

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

a small tolerance of T = 0.1 points. To complement the visualization of the



**Fig. 6** REC curves of Elasticity Warp (top left), Elasticity Weft (top right), Bias Distortion (bottom left) and Pilling (bottom right) predictions.

obtained STR results, Figure 7 presents the scatter plots of the measured (xaxis) versus the predicted values (y-axis), complemented with coefficient of determination ( $\mathbb{R}^2$ ) for the specific 6<sup>th</sup> 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 7) are close to the ideal prediction diagonal line and both present very good  $\mathbb{R}^2$  values of 0.90 and 0.87 respectively. While the same effect is not that visible for the pilling predictions (bottom right of Figure 7), 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  $\mathbb{R}^2$  of 0.64. The bias distortion test presented the lowest  $\mathbb{R}^2$  (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 6) shows that, for the same 6<sup>th</sup> external fold, the



Fig. 7 Regression scatter plot of Elasticity Warp (top left), Elasticity Weft (top right), Bias Distortion (bottom left) and Pilling (bottom right) predictions.

selected **STR** prediction model for the 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.

### 5.2 Textile Composition Prediction Results

Table 5 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-fold test sets, 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. Similarly to the physical property results, the **STR** approach presents 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 6, 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

	MAE		
%Fiber	STR	MTR	
СО	<b>0.009</b> ±0.002	$0.013 \pm 0.002$	
CV	$0.006 \pm 0.000^*$	$0.017 {\pm} 0.001$	
$\mathbf{EL}$	0.002±0.000*	$0.006 {\pm} 0.000$	
PA	0.002±0.001*	$0.005 {\pm} 0.001$	
PES	0.010±0.001*	$0.020 {\pm} 0.002$	
WO	$0.002 \pm 0.001^*$	$0.011 {\pm} 0.002$	
Average	0.005	0.012	

Table 5Fabric composition predictive results (mean MAE test set values associated with<br/>its student-t 95% confidence intervals; best results are in **bold**).

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

**Table 6** Percentage of fabric compositions correctly predicted per approach (best results per T value are in **bold**).

		STR			MTR	
т	N	Р	s	N	Р	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%

strategies, all three strategies provide high classification 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.

### 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). 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 set up 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 set up 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,  $\epsilon$  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 7 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) [44, 45]. Statistical significance is measured by executing paired t-test (for the **Mean** values) and Wilcoxon tests (for the **Median** values). When comparing the EMO methods, it becomes clear that the R-NSGA-II is the best approach. It outperforms the NSGA-II in both aggregation measures, with a difference of around 5 points for the mean and around 10 points for the median.

Table 7Hypervolume performance results (in %) for the 100 fabrics (best results perapproach are in **bold**).

Method	Mean	Median	Median Interval
NSGA-II	$20.86 {\pm} 3.25 \\ \textbf{25.95} {\pm} \textbf{4.31}$	24.31	[21.21, 28.04]
R-NSGA-II		<b>34.22</b> *	[ <b>30.09, 38.78</b> ]

\* - 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 8 summarizes the obtained results in terms of the mean and median values of all 100 selected points for each individual objective ( $f_p$  and  $f_d$ ). Table 8 also presents the mean and median values for the required CPU time (in s) and final number of optimized Pareto solutions. Similarly to Table 7, the student-t and Wilcoxon-Signed-Rank 95%confidence intervals and paired tests were also computed. 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 3 percentage points for the predictive error  $(f_p)$  when compared with NSGA-II. The R-NSGA-II optimized points that are closer to the ideal (0,0) solution present similar mean and median values, which are 11% for the predictive error  $(f_p)$  and 16% for the inputs distance  $(f_d)$ . Regarding the additional EMO evaluation criteria, R-NSGA-II requires a slight higher computational effort when compared with NSGA-II, returning an average execution of 1,852 seconds. This value is still considered reasonable for a new fabric design optimization, since it corresponds to around 30 minutes. More importantly, R-NSGA-II tends to return a richer Pareto front, containing on average around 50 distinct solutions, which is substantially better when compared with NGSA-II (average of around 22) Pareto points).

Method		Mean	Median	Median Interval
NSGA-II	Distance $(f_d)$ Predictive error $(f_p)$ CPU time	$\begin{array}{c} 0.18 \pm 0.01 \\ 0.14 \pm 0.01 \\ \mathbf{1,750.2 \pm 0.3} \end{array}$	0.18 0.14 <b>1,750</b>	$\begin{matrix} [0.17,\ 0.19] \\ [0.12,\ 0.16] \\ [1749.9,\ 1750.4] \end{matrix}$
R-NSGA-II	Pareto solutions Distance $(f_d)$ Predictive error $(f_p)$	$\begin{array}{c} 22.1 \pm 3.5 \\ \textbf{0.16} {\pm} \textbf{0.01}^* \\ \textbf{0.11} {\pm} \textbf{0.01}^* \end{array}$	20 0.16* 0.11*	[21.5, 27.0] [0.15, 0.17] [0.10, 0.12]
R-N5GA-II	CPU time Pareto solutions	$\begin{array}{c} 1,852.2\pm 0.3 \\ \textbf{49.7} {\pm} \textbf{8.0}^{*} \end{array}$	1,852 <b>45.5</b> *	[1,852.0, 1,852.5] [ <b>57.5, 73.0</b> ]

 Table 8
 Summary of the selected point results, required EMO CPU time and obtained number of Pareto solutions (best results per approach are in **bold**)

\* - 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 8 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 comparing 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.



**Fig. 8** NSGA-II and R-NSGA-II hypervolume (y-axis, in %) generation evolution (x-axis) for fabrics #64 (left) and #82 (right).

Another demonstration example is provided in Figure 9, 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 0.290and 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 0.015to 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  $\#82 - (f_d = 0.097, f_p = 0.053).$ 

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 R-NSGA-II convergence, Figure 10 presents the evolution of the solutions towards the Pareto-optimal front. Each point represents a potential solution and line segments are used to connect the points that belong



**Fig. 9** 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  $(f_p)$ .

to the Pareto front. A color scheme is employed to facilitate the visual inspection of the plots, ranging from light gray (first generation) to full black (last generation). For both #64 and #82 fabrics, there is an initial fast R-NSGA-II convergence, with substantial movements of the Pareto front towards the interesting bottom left region.



Fig. 10 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  $(f_p)$ .

### 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 Intelligent Decision Support System (IDSS) that combines predictive and prescriptive analytics.

### Springer Nature 2021 IATEX template

### 26 A data-driven intelligent decision support system

Using a large and realistic set of 44 fabric design 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 and weft 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 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 Multiobjective Optimization (EMO) search. Using the best physical property prediction models (provided by STR), the EMO optimizes the set of 44 fabric inputs that simultaneously minimizes the physical property predictive error and its distance to the training input space. 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. Moreover, the R-NSGA-II method returned a richer set of Pareto points (average of around 50 distinct solutions for each new fabric).

We note that the obtained results can not be directly compared with the state-of-the-art works, since the four related Fabric Design (FD) studies (Section 2) are more limited and target different FD goals. For instance, the ANNs used in [4] predicted different physical properties (ultraviolet protection factor, air permeability and moisture vapor transmission rate). The study used a smaller dataset (with just 42 fabric examples) and a simpler validation procedure (random holdout split and not the 10-fold cross-validation adopted in this work). Furthermore, in [4] a Genetic Algorithm was used to reduce the ANN predictive error (single objective) regarding the three physical properties and only 4 new fabrics were optimized, presenting a fabric property deviation that ranged from 2% to 20%. While these results are comparable to our average 11% predictive error, we note that our research contains substantial differences. For instance, we target four different physical properties (e.g., weft elasticity) and optimize a much larger set of 100 fabrics under a true EMO approach that also aims to reduce the distance to the learned input space. In terms of scientific contributions, this research shows that the combination of predictive and prescriptive analytics can be a valuable tool to address the nontrivial FD task. Indeed, the predictive and prescriptive IDSS results were shown to the textile company experts, which provided a very positive feedback.

While interesting results were obtained, this research includes some limitations. As already mentioned in Section 4.2, this paper assumes an objective evaluation of the EMO results by using unseen test data. Thus, there was no assessment of the IDSS interaction with textile designers. Moreover, since the proposed IDSS was not deployed in a real environment, we have not directly measured the IDSS value in terms of reducing the number of FD prototype attempts. In effect, in future work, we intend to address these limitations by deploying the IDSS in the real textile company environment. This will allow us to gather further feedback from the textile designers by using the Technology Acceptance Model (TAM) model, such as executed in [46]. Also, the IDSS deployment will be used to confirm its potential to reduce the costs and time associated with each new fabric development. Also, we plan to study a many objective approach [47], which is more challenging than the approached biobjective 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). Finally, although the combination of predictive and prescriptive analytics was targeted to support the creation of new textile fabrics, it can be adapted to other production domains that have similar prototype design processes, such the plastic and chemical industries.

# 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).

# Data availability

The datasets analyzed during the current study are not publicly available due to business privacy issues of the Portuguese textile company but are available from the corresponding author on reasonable request.

# Compliance with ethical standards

Conflict of interest – The authors declare no conflict of interest.

# References

 Rachel Studd: The textile design process. The Design Journal 5(1), 35–49 (2002)

- Hu, J.: Introduction to fabric testing. In: Hu, J. (ed.) Fabric Testing. Woodhead Publishing Series in Textiles, pp. 1–26. Woodhead Publishing, Cambridge (2008). https://doi.org/10.1533/9781845695064.1
- [3] Arnott, D., Pervan, G.: A critical analysis of decision support systems research revisited: The rise of design science. Journal of Information Technology 29(4), 269–293 (2014). https://doi.org/10.1057/jit.2014.16
- [4] Majumdar, A., Das, A., Hatua, P., Ghosh, A.: 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 (2016). https://doi.org/10.1007/s00521-015-2025-6
- [5] Majumdar, A., Mal, P., Ghosh, A., Banerjee, D.: 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 (2017). https://doi.org/10.1080/00405000.2016.1159270
- [6] Ribeiro, R., Pilastri, A., Moura, C., Rodrigues, F., Rocha, R., Morgado, J., Cortez, P.: Predicting physical properties of woven fabrics via automated machine learning and textile design and finishing features. In: Maglogiannis, I., Iliadis, L., Pimenidis, E. (eds.) Artificial Intelligence Applications and Innovations, pp. 244–255. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-49186-4\_21
- Michalewicz, Z., Schmidt, M., Michalewicz, M., Chiriac, C.: Adaptive Business Intelligence. Springer, Berlin Heidelberg (2006). https://doi.org/ 10.1007/978-3-540-32929-9
- [8] Ferreira, L., Pilastri, A.L., Martins, C.M., Pires, P.M., Cortez, P.: 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, pp. 1–8. IEEE, NJ, USA (2021). https://doi.org/10.1109/IJCNN52387.2021.9534091
- [9] Arashloo, S.R., Kittler, J.: Multi-target regression via non-linear output structure learning. Neurocomputing 492, 572–580 (2022). https: //doi.org/10.1016/j.neucom.2021.12.048
- [10] Goodfellow, I.J., Bengio, Y., Courville, A.C.: Deep Learning. Adaptive computation and machine learning. MIT Press, MA, USA (2016). http://www.deeplearningbook.org/
- [11] Cortez, P.: Modern Optimization with R. Springer, New York (2021). https://doi.org/10.1007/978-3-030-72819-9
- [12] Ghosh, A., Das, S., Banerjee, D.: 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 (2013). https://doi.org/10.1007/s40034-013-0015-8

- [13] Das, S., Ghosh, A., Banerjee, D.: Designing of engineered fabrics using particle swarm optimization. International Journal of Clothing Science and Technology 26(1), 48–57 (2014)
- Gloy, Y.-S., Renkens, W., Herty, M., Gries, T.: Simulation and optimisation of warp tension in the weaving process. Journal of Textile Science & Engineering 5, 1000179 (2015). https://doi.org/10.4172/2165-8064. 1000179
- [15] Junior, A.U., de Freitas Filho, P.J., Silveira, R.A.: 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., Galicia-Haro, S.N. (eds.) 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. Lecture Notes in Computer Science, vol. 9413, pp. 413–430. Springer, Cham (2015). https://doi.org/10.1007/ 978-3-319-27060-9.34
- [16] Mitra, A., Majumdar, P.K., Banerjee, D.: 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 (2015). https://doi.org/10.1007/s40034-014-0048-7
- [17] Zhang, R., Chang, P.-C., Song, S., Wu, C.: A multi-objective artificial bee colony algorithm for parallel batch-processing machine scheduling in fabric dyeing processes. Knowledge-Based Systems 116, 114–129 (2017). https://doi.org/10.1016/j.knosys.2016.10.026
- [18] Chakraborty, S., Diyaley, S.: 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 (2018). https://doi.org/10.1007/s40034-018-0121-8
- [19] Huynh, N., Chien, C.: A hybrid multi-subpopulation genetic algorithm for textile batch dyeing scheduling and an empirical study. Comput. Ind. Eng. 125, 615–627 (2018). https://doi.org/10.1016/j.cie.2018.01.005
- [20] 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., Blanco-Valencia, X.P.: 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., Corchado Rodríguez, E. (eds.) Hybrid Artificial Intelligent Systems, pp. 674–685. Springer, Cham (2019)

- [21] Liyanage, I., Nuwanga, S., Anjana, R., Rankothge, W., Gamage, N.: Sustainable manufacturing: Application of optimization to textile manufacturing plants. Global Journal of Computer Science and Technology (2020). https://doi.org/10.34257/GJCSTHVOL20IS2PG11
- [22] Xu, Y., Sébastien, T., Xianyi, Z.: 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 (2020). https://doi.org/10.1007/s00170-019-04866-w
- [23] Jaouachi, B., Khedher, F.: Assessment of jeans sewing thread consumption by applying metaheuristic optimization methods. International Journal of Clothing Science and Technology 34(3), 347–366 (2021). https: //doi.org/10.1108/IJCST-01-2021-0005
- [24] Ferro, R., Cordeiro, G.A., Ordóñez, R.E.C., Beydoun, G., Shukla, N.: An optimization tool for production planning: A case study in a textile industry. Applied Sciences 11(18), 8312 (2021). https://doi.org/10.3390/ app11188312
- [25] Ribeiro, R., Pilastri, A., Carvalho, H., Matta, A., Pereira, P.J., Rocha, P., Alves, M., Cortez, P.: An intelligent decision support system for production planning in garments industry. In: Yin, H., Camacho, D., Tino, P., Allmendinger, R., Tallón-Ballesteros, A.J., Tang, K., Cho, S.-B., Novais, P., Nascimento, S. (eds.) Intelligent Data Engineering and Automated Learning – IDEAL 2021, pp. 378–386. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-91608-4\_37
- [26] Tsao, Y.-C., Hung, C.-H., Vu, T.-L.: Hybrid heuristics for marker planning in the apparel industry. Arabian Journal for Science and Engineering 46(10), 10077–10096 (2021). https://doi.org/10.1007/ s13369-020-05210-1
- [27] Zhang, Z., Guo, C., Wei, Q., Guo, Z., Gao, L.: A bi-objective stochastic order planning problem in make-to-order multi-site textile manufacturing. Comput. Ind. Eng. 158, 107367 (2021). https://doi.org/10.1016/j. cie.2021.107367
- [28] Elahi, I., Ali, H., Asif, M., Iqbal, K., Ghadi, Y., Alabdulkreem, E.: An evolutionary algorithm for multi-objective optimization of freshwater consumption in textile dyeing industry. PeerJ Computer Science 8, 932 (2022). https://doi.org/10.7717/peerj-cs.932
- [29] Hastie, T., Tibshirani, R., Friedman, J.H.: The Elements of Statistical

Learning: Data Mining, Inference, and Prediction, 2nd Edition. Springer Series in Statistics. Springer, NY, USA (2009). https://doi.org/10.1007/978-0-387-84858-7

- [30] Campos, G.O., Zimek, A., Sander, J., Campello, R.J.G.B., Micenková, B., Schubert, E., Assent, I., Houle, M.E.: On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. Data Mining and Knowledge Discovery 30(4), 891–927 (2016). https: //doi.org/10.1007/s10618-015-0444-8
- [31] Matos, L.M., Azevedo, J., Matta, A., Pilastri, A., Cortez, P., Mendes, R.: Categorical attribute transformation environment (cane): A python module for categorical to numeric data preprocessing. Software Impacts 13, 100359 (2022). https://doi.org/10.1016/j.simpa.2022.100359
- [32] LeDell, E., Poirier, S.: H2O AutoML: Scalable automatic machine learning. 7th ICML Workshop on Automated Machine Learning (AutoML) (2020)
- [33] Chollet, F.: Deep Learning with Python, 2nd edn. Simon and Schuster, NY, USA (2021)
- [34] Matos, L.M., Cortez, P., Mendes, R., Moreau, A.: Using deep learning for mobile marketing user conversion prediction. In: International Joint Conference on Neural Networks, IJCNN 2019 Budapest, Hungary, July 14-19, 2019, pp. 1–8. IEEE, NY, USA (2019). https://doi.org/10.1109/ IJCNN.2019.8851888
- [35] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015). http://arxiv.org/abs/1412.6980
- [36] Sumeet Saurav, Saini, A.K., Saini, R., Singh, S.: Deep learning inspired intelligent embedded system for haptic rendering of facial emotions to the blind. Neural Comput. Appl. 34(6), 4595–4623 (2022). https://doi.org/ 10.1007/s00521-021-06613-3
- [37] Oliveira, N., Cortez, P., Areal, N.: 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 (2017). https://doi.org/10.1016/j.eswa.2016.12.036
- [38] Bi, J., Bennett, K.P.: Regression error characteristic curves. In: Fawcett, T., Mishra, N. (eds.) Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003), August 21-24, 2003, Washington, DC, USA, pp. 43–50. AAAI Press, California, USA (2003).

http://www.aaai.org/Library/ICML/2003/icml03-009.php

- [39] Blank, J., Deb, K.: Pymoo: Multi-objective optimization in python. IEEE Access 8, 89497–89509 (2020). https://doi.org/10.1109/ACCESS.2020. 2990567
- [40] Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. 6(2), 182–197 (2002). https://doi.org/10.1109/4235.996017
- [41] Chiandussi, G., Codegone, M., Ferrero, S., Varesio, F.E.: Comparison of multi-objective optimization methodologies for engineering applications. Computers & Mathematics with Applications 63(5), 912–942 (2012). https://doi.org/10.1016/j.camwa.2011.11.057
- [42] Deb, K., Sundar, J.: Reference point based multi-objective optimization using evolutionary algorithms. In: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation. GECCO '06, pp. 635–642. Association for Computing Machinery, New York, NY, USA (2006). https://doi.org/10.1145/1143997.1144112
- [43] While, L., Hingston, P., Barone, L., Huband, S.: A faster algorithm for calculating hypervolume. IEEE Trans. Evol. Comput. 10(1), 29–38 (2006). https://doi.org/10.1109/TEVC.2005.851275
- [44] Hollander, M., Wolfe, D.A., Chicken, E.: Nonparametric Statistical Methods. John Wiley & Sons, NJ, USA (2013)
- [45] Cortez, P., Pereira, P.J., Mendes, R.: Multi-step time series prediction intervals using neuroevolution. Neural Comput. Appl. 32(13), 8939–8953 (2020). https://doi.org/10.1007/s00521-019-04387-3
- [46] António João Silva, Cortez, P.: An Industry 4.0 Intelligent Decision Support System for Analytical Laboratories. In: Maglogiannis, I., Iliadis, L., MacIntyre, J., Cortez, P. (eds.) Artificial Intelligence Applications and Innovations 18th IFIP WG 12.5 International Conference, AIAI 2022, Hersonissos, Crete, Greece, June 17-20, 2022, Proceedings, Part II. IFIP Advances in Information and Communication Technology, vol. 647, pp. 159–169. Springer, Cham, Switzerland (2022). https://doi.org/10.1007/978-3-031-08337-2\_14
- [47] Guo, X.: A survey of decomposition based evolutionary algorithms for many-objective optimization problems. IEEE Access 10, 72825–72838 (2022). https://doi.org/10.1109/ACCESS.2022.3188762

# A Acronyms Used

Tables 9 and 10 present the full list of acronyms used in work.

Table 9 List of acror	yms used in Table 1.
-----------------------	----------------------

Acronym	Description
ABC	Artificial Bee Colony
ACO	Ant Colony Optimization
ANN	Artificial Neural Network
AP	Air Permeability
AutoML	Automated Machine Learning
CE	Carbon Emission
CO	Cutting Orders
COP	Cutting Order Planning
CP	Comfort Properties
DO	Dyeing Orders
EC	Energy Consumption
EPI	Ends per Inch
F	Fibers
FC	Fabric Construction parameters
FCp	Fabric Cutting patterns
FD	Fabric Design
FDF	Fabric Design Features
FO	Finishing Operations
FP	Fabric Properties
FW	Finishing Workflow
GA	Genetic Algorithm
GP	Genetic Programming
GPC	Garment Production Cost
GPT	Garment Production Time
GSM	Fabric Grams per Square Meter
HGASA	Hybrid Genetic Algorithm Simulated Annealing
MA	Memetic Algorithm
Mf	Manufacturing of fabrics
Mg	Manufacturing of garments
$\mathbf{ML}$	Marker Layout
MO-ABC	Multi-Objective Artificial Bee Colony
MOGCO-II	Multi-Objective Group Counseling Optimizer II
MPS	Master Production Scheduling
MS	Machine Settings
MSGA-H	Multi-Subpopulation Genetic Algorithm Heuristics
NSGA-II	Non-dominated Sorting Genetic Algorithm II
OP	Operation Parameters
PC	Production Capacities
pesgMESO	parallel evolution scenario generation-based Multi-objective Evolutionary Stochastic Optimization

 Table 10
 List of acronyms used in Table 1 (continued).

Acronym	Description
PiO	Stochastic Optimization
PO	Production Orders
PP	Production Planning
PPI	Picks per Inch
PPWP	Production Planning of Weaving Process
PR	Productive Resources
PSO	Particle Swarm Optimization
QAW	Quality Assurance Workflow
RiSP	Ring Spinning Process
$\mathbf{RMQ}$	Raw Material Quality
R-NSGA-II	Reference point based Non-dominated Sorting Genetic Algorithm II
RoSP	Rotor Spinning Process
SA	Simulated Annealing
SF	Sales Forecast
$\mathbf{SR}$	Subcontractor Resources
ST	Setup Times
STBD	Scheduling Textile Batch Dyeing
STC	Sewing Thread Consumption
SW	Sewing Workflow
TC	Tardiness cost
ThC	Thermal Conductivity
TR	Transportation Resources
TT	Total Tardiness
UPF	Ultraviolet Protection Factor
URDV	Utilization Rate of Dyeing Vats
WC	Water Consumption
WaC	Warp Count
WeC	Wert Count
WMP	Weaving Machine Parameters
WIC	Wastewater Treatment Costs
I VDO	Yarns Vere Desire Orless
YDS	Yarn Dyeing Orders
IDS VC	Varia Strangth
13	rarn Strength