Operations Research Perspectives 12 (2024) 100308

Contents lists available at ScienceDirect

Operations Research Perspectives

journal homepage: www.elsevier.com/locate/orp

Automated machine learning methodology for optimizing production processes in small and medium-sized enterprises

Yarens J. Cruz [a,*], Alberto Villalonga [a], Fernando Castaño [a], Marcelino Rivas [b], Rodolfo E. Haber [a]

[a] Centro de Automática y Robótica, CSIC-Universidad Politécnica de Madrid, Madrid 28500, Spain
[b] Centro de Estudios de Fabricación Avanzada y Sostenible, Universidad de Matanzas, Matanzas 40100, Cuba

ARTICLE INFO

Keywords:
Automated machine learning
Automl
Model selection
Hyperparameter optimization
R-NSGA-II
Multi-objective optimization

ABSTRACT

Machine learning can be effectively used to generate models capable of representing the dynamic of production processes of small and medium-sized enterprises. These models enable the estimation of key performance indicators, and are often used for optimizing production processes. However, in most industrial applications, modeling and optimization of production processes are currently carried out as separate tasks, manually in a very costly and inefficient way. Automated machine learning tools and frameworks facilitate the path for deriving models, reducing modeling time and cost. However, optimization by exploiting production models is still in infancy. This work presents a methodology for integrating a fully automated procedure that embraces automated machine learning pipelines and a multi-objective optimization algorithm for improving the production processes, with special focus on small and medium-sized enterprises. This procedure is supported on embedding the generated models as objective functions of a reference point based non-dominated sorting genetic algorithm, resulting in preference-based Pareto-optimal parametrizations of the corresponding production processes. The methodology was implemented and validated using data from a manufacturing production process of a small manufacturing enterprise, generating highly accurate machine learning-based models for the analyzed indicators. Additionally, by applying the optimization step of the proposed methodology it was possible to increase the productivity of the manufacturing process by 3.19 % and reduce its defect rate by 2.15 %, outperforming the results obtained with traditional trial and error method focused on productivity alone.

1. Introduction

Nowadays, generalized adoption of the new manufacturing paradigms involves the assimilation of key technologies such as intelligent data analysis and machine learning (ML), among others, aiming at the digital transformation of enterprises [46,56]. This digital transformation is crucial to keep up with the competition, especially in the case of small and medium-sized manufacturing enterprises (SMEs). One contemporary target is to create a highly reconfigurable, decentralized, dynamic, self-organizing, and real-time (or near real-time) decision-making infrastructure enabling to analyze customer expectations and reach their targets [20]. By applying these transformations, SMEs should be capable of monitoring and improving their key performance indicators (KPIs) [67]. However, in practice, SMEs face several difficulties in applying these technologies, mostly related to the transition and maintenance costs, innovation complexity, and personnel training [50]. Additionally, due to limited human and computational resources, time constraints and complexity of the optimization processes, SMEs usually focus their efforts on a single productivity objective, despite being more desirable to consider and optimize multiple objectives, which leads not only to more efficient but also a more sustainable and environmentally friendly production. Due to these obstacles, a large number of SMEs don't count yet with the necessary tools to continue with their digital transformation. In this context, the development of tools for generating useful information and smart recommendations of production systems in SMEs is almost mandatory in a high-competitive market and has a large number of potential adopters [12,42].

The interest for industrial ML applications has grown significantly in recent years. However, the design and deployment of ML-based solutions largely still follow traditional approaches, leading to high dependency of domain experts and time-consuming development processes [13,15,16]. For dealing with this situation, automated

* Corresponding author.
E-mail address: y.cruz@car.upm-csic.es (Y.J. Cruz).

https://doi.org/10.1016/j.orp.2024.100308
Received 14 November 2023; Received in revised form 25 April 2024; Accepted 10 June 2024
Available online 11 June 2024
2214-7160/© 2024 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

machine learning (AutoML) has emerged for saving time and effort on repetitive tasks during the creation of ML-based solutions [55]. The creation of an ML product usually involves operations such as data pre-processing, feature engineering, model training, model selection, and hyperparameter optimization [36]. A typical AutoML workflow is depicted in Fig. 1. By automating these tasks, an ML solution can be obtained in a short amount of time, and it has the added benefit that it allows non-experts to use these technologies [59]. Specifically, manufacturing SMEs can benefit from ongoing research and developments on AutoML given the current need for solutions that are easy to adopt and minimize the personnel qualification requirements. In contrast to large companies, manufacturing SMEs typically do not have an in-house team of ML experts and, without AutoML tools, they would be forced to rely on costly external talent. Additionally, the versatility of AutoML tools allows to apply them across different domains or production processes without the need to manually create a solution from scratch each time, contributing to accelerate the digital transformation of manufacturing SMEs.

Although some open source and commercial AutoML solutions are available such as auto-sklearn [23], Auto-Keras [32], H2O AutoML [34], Google Cloud AutoML, Microsoft Azure Machine Learning, TransmogrifAI, among others, AutoML is still a very active research field [71]. A recent study has proposed the use of an evolutionary algorithm for finding the best classifier ensemble and hyperparameter setting in an AutoML workflow [61]. Another reported strategy introduced a meta-feature-free meta-learning technique using a bandit strategy in budget allocation for building a portfolio of AutoML pipelines and carry out a greedy search to select the best candidate depending on the dataset [24]. A previous version of a tree-based AutoML software has been extended to include neural network estimators [48]. Similarly, an AutoML platform has been developed to build ML models subject to multiple objectives, as well as resource and hardware constraints [64]. An AutoML framework focused on deep learning has been developed to jointly optimize deep network architectures and training parameters [70]. The use of evolutionary algorithms for the automatic design of composite ML pipelines has also been explored [40]. Other recent studies have been more focused on the explainability of the models obtained with AutoML tools on Big Industrial Data [26].

Overall, up-to-date reported AutoML solutions generate ML models (or ensemble of models) with optimal hyperparameters fitted to the provided data with certain reliability and that, in some cases, are explainable. The problem they were designed to solve is commonly known as combined algorithm selection and hyperparameter optimization (CASH). In many industrial cases, this does not represent a complete automated solution to practical situations, since the yielded model should be also used to optimize the process that generated the data, and this task is currently performed manually. In other words, current approaches for applying AutoML solutions in manufacturing only allows to automatically obtain data-driven models with optimal hyperparameters.

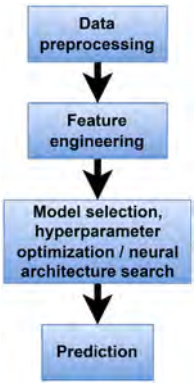

Fig. 1. Typical AutoML workflow.

In order to use them for optimizing the production processes is necessary the reliance on experts in the field to conduct optimization studies which are generally configured and triggered manually. This is a critical issue for companies facing small batch manufacturing and/or agile manufacturing strategies, requiring fast decision-making since they should avoid time-consuming manual intervention. Moreover, in most applications the preferences of the decision maker are not taken into consideration during the optimization process, which implies, for example, that after optimization the technologist or plant manager should search for an appropriate solution within the entire Pareto set. Another issue with the existing AutoML solutions is that although they simplify the modeling tasks, it is still required qualified personnel to use them, whether they are used as a cloud service or in local servers. Furthermore, in order to mirror the multifaceted nature of production environments, complex models are generally used, which demands significant computational power. Above mentioned issues, ultimately, represent a truly bottleneck for the democratization of AutoML, specially for SMEs. These companies, accounting for more than 90 % of all businesses in many countries, have to face challenges such as lack of personnel with skills in ML or data analytics, or limited access to computational resources [53].

To face these challenges, this work introduces an end-to-end fully AutoML methodology that includes an evolutionary algorithm for the preference-based multi-objective optimization of production processes. The goal of this methodology is not only to generate optimal ML-based models of the production processes but also using these models to obtain and recommend different sets of parameters that the user can use to improve the performance of the production process by setting desired optimal regimes according to the Pareto optimality criterion, improving productivity and thus efficiency. The use of the preference-based optimization is especially beneficial because the results generated have been already refined considering the decision maker's interests, making them directly exploitable in the production process, which is an advantageous approach for highly dynamic environments. This implies that it is not necessary to manually explore and assess optimization results. Additionally, information about the most relevant data features for modeling is also provided. Furthermore, by simplifying the way of interacting with this methodology, it is guarantee that non-expert users can easily use it. This methodology is applied and validated using data from a manufacturing case study.

The paper consists of five sections. After this introduction, all the aspects of the methodology are explained in Section 2. Section 3 introduces a case study for validating the methodology where the AutoML pipeline is created and then used to carry out a two-objectives optimization. Later, Section 4 discusses the obtained results. Finally, Section 5 shows the conclusions and outlines future works.

2. Materials and methods

2.1. Dataset preparation, preprocessing and general description of the methodology

The proposed methodology starts by preprocessing a historical dataset of a production process and then perform three main steps: (1) determine which are the features with higher influence on predefined KPIs of a production system, (2) create regression models for each KPI of the production system, and (3) use these models to generate parametrizations that improve the KPIs through a multi-objective optimization algorithm.

A necessary step before applying this methodology consists of creating a historical dataset of the production system containing variables, features and KPIs values. Most ML libraries that implement the algorithms included in the methodology are designed to work only with numerical features. Since the methodology will be built on top of these libraries the datasets should be prepared in advance to comply with this requirement. If the data collected originally contains categorical

features they must be encoded using an ordinal codification scheme or a one-hot codification scheme depending on if they have inherent order or not, respectively. Also, the KPIs included in the dataset should be quantitative metrics related to the production process, which implies that the models that will be generated using the methodology are regression models. In addition to historical data, an optimization objective (i.e., minimization or maximization) for each KPI, reference or desired values (e.g., KPIs values that represent desired productivity), a feasible range for each feature, and any other constraints, if any, should be also defined to be able to carry out the optimization process.

There are no limits on the number of input features or the number of outputs (KPIs) to consider; although, the methodology is designed for being applied to SMEs, therefore the expected data should not contain as much information as the data acquired in a massive shop floor. Assessing the performance of the proposed methodology for big volumes of data is out of the scope of this study.

The inputs of the methodology are a historical dataset, and a file specifying the KPIs' objectives, reference values, features range, and constraints. These two files are the only requirements for interacting with the proposed methodology over a platform designed to execute all the processing and generate a file with the optimization results [4,41]. The idea behind this approach is to make it possible for even SMEs that do not have the required computational capacity or staff with ML or programming skills to benefit from this type of technology, in line with the concepts outlined for service-based platforms for the democratization of Artificial Intelligence [21,27].

Once the dataset is received, the first step carried out by the methodology is data preprocessing. Preprocessing is crucial for preparing data for machine learning models. For example, raw data is often inconsistent, containing missing values or errors that can affect model training. Removing poor-quality data can enhance model robustness and performance. Therefore, the received data it is examined to detect missing data and delete the corresponding samples, if applicable. Then, the integrity of the dataset is checked to comply with the specification of only containing numerical values. Once the dataset integrity has been checked, it is split using 60 % data for training, 20 % for validation, and 20 % for testing. Then, the training, validation, and test input vectors of each feature x are normalized using the following equation:

$$\bar{f}_{x\_std} = \frac{\bar{f}_x - \mu_{x,tr}}{\sigma_{x,tr}} \tag{1}$$

where $\bar{f}_x$ is a vector containing the feature values, $\mu_{x,tr}$ is the mean of the feature values in the training set, and $\sigma_{x,tr}$ is the standard deviation of the feature values on the training set.

Normalizing the features is a critical step in the methodology because later the features will be analyzed to determine which are the most relevant ones for each KPI. After data normalization, the methodology proposes a series of steps to accomplish its goals. Fig. 2 shows a high-level schema of the proposed methodology. The following subsections describe the most important operations of the methodology.

2.2. Feature selection

The feature engineering step is fundamental for creating high-quality ML models, since they depend, ultimately, on the data used for training. The goal of feature engineering is to improve the quality of the available data for making the training process more efficient. Feature engineering

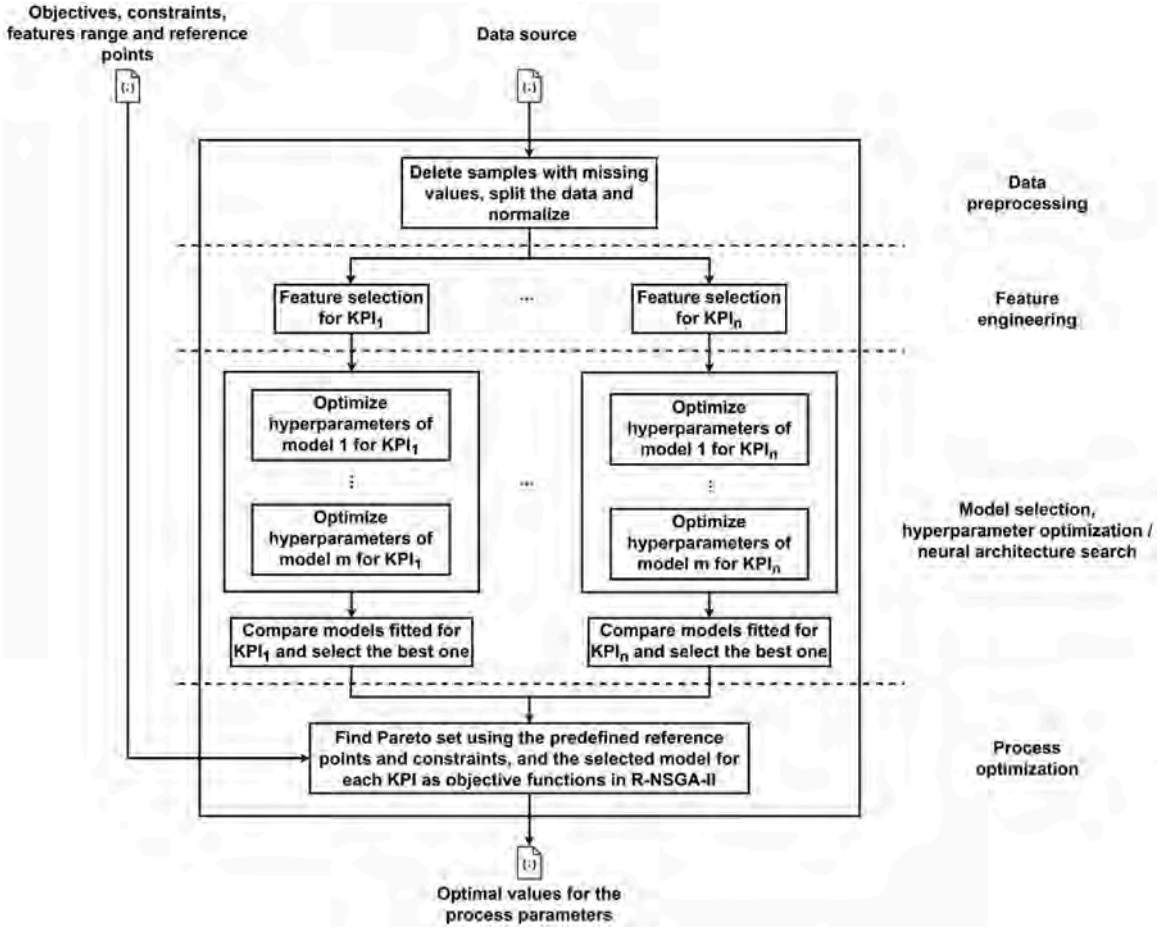

Fig. 2. High-level schema of the proposed methodology.

can include operations such as feature generation or feature selection. Particularly, the design of the proposed methodology does not consider feature generation since this operation adds more features to the dataset increasing the complexity of the models and the computational time. Additionally, the automated generation of new features can result in non-interpretable features or redundant information which can lead to overfitting of the models. Alternatively, feature selection is considered in the proposed methodology for dealing with datasets that already contain irrelevant or redundant features, which impact in the computational cost or result in biased models. Feature selection allows to choose the most useful features of a dataset for building models, resulting not only in a reduction of the dimensionality of the data; but also leads to more compact models with better generalization ability and reduced computational time [3,11,52]. Feature selection is also an enabler of explainability in ML since it helps identify which inputs have the stongest impact on the outputs.

Given that the methodology is designed to work with a wide range of data and there is no a priori knowledge of how many features and samples will contain the dataset, a method based on Pearson's correlation coefficient was chosen for feature selection. This is a widely used approach when the data analyzed consists of numerical inputs and outputs, which is the case in the proposed methodology [25]. Additionally, by using this approach, computational cost remains low, which is necessary to obtain the desired information in a relatively short period, taking into consideration that there are other tasks in the methodology, besides feature selection, that requires more time to execute.

2.3. Modeling

Once the feature selection process is finished the next step is the modeling considering preestablished KPIs. The objective is to create models capable of representing the relationship among the production process parameters and variables (inputs) with KPIs as outputs. Among the available modeling approaches, ML is widely used in many research fields, often as an alternative approach [17]. Specifically, ML techniques are being used to predict KPIs taking advantage of the availability of large amounts of data generated by new digital systems [54]. Typically, these studies are focused on three main KPIs categories: economic, societal, and environmental. Some of the most simple, yet very used ML techniques for modeling KPIs are Multilayer Perceptron (MLP) [38,43,49], Support Vector Regression (SVR) [22,37,47], Ridge Regression (RR) [29], Least Absolute Shrinkage and Selection Operator regression (LASSO) [6], Random Forest (RF) (T. [57]), *k*-Nearest Neighbors (*k*-NN) [28], Gaussian Process Regression (GPR) [44,51] and Convolutional Neural Networks (CNNs) [14,66].

The previously mentioned techniques were selected for the sake of keeping certain level of diversity and according to the state-of-the-art techniques in smart manufacturing scenarios, which is crucial for determining the most appropriate model [15]. In the specific case of GPR, its inclusion in the set of modeling strategies relies on the size of the dataset. Specifically, if the dataset contains more than 1500 samples this technique is automatically discarded because of the high computational cost of GPR, which grows cubically with respect to the number of samples due to the need of calculating matrix inversions. Nevertheless, other techniques can be included, but adding more models implies more training effort, analysis and overall increasing of computational cost during execution time of the AutoML procedure.

2.4. Hyperparameter optimization

One of the key steps of the methodology consists of optimizing the hyperparameters of the models. During the creation of a model its hyperparameters must be specified; however, it is hard to know a priori which hyperparameters values will yield a good-fitting model. For this reason, ML practitioners often explore the feasible search space looking for the best hyperparameter combination. Typically, one of the following strategies is used for this task: grid search [9,65], random search [31,62], or Bayesian search (J. [68]).

Grid search implies carrying out an exhaustive search including all the possible combinations of hyperparameters. For this strategy, it is necessary to define a range to limit the search space of unbounded hyperparameters and to discretize the continuous hyperparameters. Random search explores hyperparameter combinations at random within the search space and, in this case, continuous hyperparameters do not necessarily have to be discretized. The main shortcoming of grid and random search is that they do not take advantage of previous evaluations of hyperparameter combinations. To deal with this, Bayesian optimization creates a probabilistic model of a function that maps hyperparameters values to a metric for evaluating the model fitness. On every iteration, a new promising hyperparameter combination based on the probabilistic model is chosen and evaluated; then, the probabilistic model is updated using this new information. In practice, Bayesian optimization has demonstrated to obtain better results in fewer evaluations than grid and random search ([69]). For this reason, it was selected as the hyperparameter optimization strategy in the proposed methodology for all the algorithms, except in the case of GPR, which only requires evaluating several kernels.

Table 1 summarizes parameters of the different ML-based models considered in the methodology.

2.5. Model selection criterion

The methodology explores different types of models for each KPI, but only one model for each KPI is used later and the others are discarded. In order to select the most appropriate model according to the corresponding KPI, a comparison based on a predefined metric is carried out. The performance index or figure of merit used for selecting a model was the mean squared error (MSE). MSE is differentiable and for this reason is commonly used as the default loss function in ML-related frameworks. MSE is also widely used for evaluating regression models in ML applications. This metric is particularly favored for its ability to penalize

Table 1
Hyperparameters/architecture of the models.

Model	Hyperparameter/architecture		Range/options
MLP	number of hidden layers		1–4
	number of units per layer		16–256
	initializer		glorot (normal), glorot (uniform), he (normal), he (uniform), lecun (normal), lecun (uniform)
	activation		ReLU, sigmoid, tanh, linear
SVR	kernel		linear, polynomial, rbf, sigmoid
	degree		2–5
	C		0–10
	ε		0.1–1
RR	alpha		0.1–10
	solver		svd, cholesky, lsqr, sparse_cg, sag, saga
LASSO	alpha		0.1–10
	maximum number of iterations		100–2000
	tolerance		0.00001–0.0001
RF	number of estimators		10–200
k-NN	number of neighbors		5–100
	algorithm		ball tree, kd tree, brute
GPR	kernel		constant, white, rbf, Matérn, rational quadratic, exp-sine-squared, dot-product
CNN	number of convolutional layers		1–2
	convolutional layer	number of filters	16–128
		activation	ReLU, sigmoid, tanh
	dense layer	number of units	16–128
		activation	ReLU, sigmoid, tanh
	dropout layer	rate	0–0.5

larger errors more severely than smaller ones [30]. This sensitivity to outliers is useful for selecting models that perform well on average but also in presence of extreme values.

2.6. Process optimization

After modeling considering KPIs, the next step is the optimization of the production process parameters. Many optimization methods are currently being used in research studies for process optimization including evolutionary algorithms [16,60], particle swarm optimization ([5,58]), cross-entropy method [7,8], among others. For optimizing the production processes in the proposed methodology, it was selected the reference point based non-dominated sorting genetic algorithm II (R-NSGA-II) as optimization procedure, which is a modification of the non-dominated sorting genetic algorithm II (NSGA-II) for taking into consideration the desired results in the optimization process.

NSGA-II is a multi-objective evolutionary algorithm for finding Pareto-optimal fronts [18]. NSGA-II was created to deal with some shortcomings of previous multi-objective optimization algorithms such as high computational complexity and non-elitist strategies. This algorithm implements a fast non-dominated sorting approach and a fast crowded distance computation that allows finding more efficiently a better spread of solutions and better convergence near the true Pareto-optimal front when compared to other strategies [63]. Its convenience for optimizing KPIs of industrial processes has been proved through many practical applications [35,39]. However, the solutions obtained by using NSGA-II can include non-desirable results, even if they are Pareto-optimal. In this case, it is necessary to select from the Pareto set the adequate solutions. This task can be automated by using the R-NSGA-II algorithm.

R-NSGA-II modifies NSGA-II to include a preference-based optimization strategy that allows to parallelly find a set of Pareto-optimal solutions near some reference points [19]. Its outline is very similar to NSGA-II, but it implements a modified survival selection. The individuals are first selected front wise; however, not all of them are allowed to survive. A second selection using a rank based on the normalized Euclidean distance to the reference points is carried out. The normalized distance d from an individual $f(x)$ to the reference point z is calculated using the following equation:

$$d = \sqrt{\sum_{i=1}^{M} \left(\frac{f_i(x) - z_i}{f_i^{max} - f_i^{min}} \right)^2} \tag{2}$$

where M is the number of objectives and f_i^{max} and f_i^{min} are the population maximum and minimum function values of the i-th objective. Finally, R-NSGA-II also implements the ϵ-based selection strategy to ensure a spread of solutions near the preferred Pareto-optimal regions.

A detailed description of the algorithm can be found in the documentation of the Pymoo library [10]. Table 2 summarizes the parameters assigned to this algorithm in the proposed methodology.

Table 2
R-NSGA-II parameters.

Parameter		Value
population size		500
offspring size		500
sampling		random
crossover	operator	simulated binary crossover
	probability	0.9
	distribution index	15
mutation	operator	polynomial mutation
	probability	1.0
	distribution index	20
number of generations		100
ϵ		0.00001

3. A manufacturing case study

For validating the proposed methodology, a production process of an SME specialized in manufacturing aerospace components is considered as case study. The complexity of aerospace systems poses a unique challenge in manufacturing due to their high-performance requirements related to safety concerns. The analyzed company is currently facing a challenge that relates to transitioning towards small batch and agile manufacturing paradigms, which are both crucial strategies for modern businesses. This shift requires a high degree of flexibility and responsiveness to accommodate varying customer demands and to manage frequent changes in product designs and specifications. Therefore, it is necessary the use of tools for generating actionable insights, enabling a faster and more informed decision-making. In this context, data-driven solutions enable the creation of proactive management tools for enhancing the process efficiency and meet production requirements in a timely manner. Specifically, ML presents several advantages over other approaches for representing the complex relationships underlying data from manufacturing processes including its capacity for handling large multi-dimensional volumes of data beyond human ability, its generalization capacity, and scalability, among others. However, this company does not have personnel with ML knowledge. Although existing AutoML tools could be helpful in this case since they simplify the workflow for creating models, there would still be unresolved challenges if they were used. Firstly, once the models are created they must be used for generating optimal parametrizations of the production process but current AutoML methodologies lack this stage. In other words, they allow to create the models and optimize their hyperparameters but do not automatically exploit these models. Secondly, while developing a custom solution for exploiting the models generated by existing AutoML tools is possible, on the one hand, this implies a long development period, which would delay the transition of the SME to the new manufacturing paradigms, and on the other hand, an in-depth knowledge of ML and multi-objective optimization is required, which the company does not have. Under these circumstances, the proposed methodology is a suitable candidate since it allow to create the models and automatically use them to generate optimal parametrizations of the production process for supporting the decision-making and, additionally, it only requires minimal training of the staff to teach them how to structure the data and configuration file, how to send the information to the methodology, and how to interpret the results, avoiding having to train them in machine learning subjects.

The process starts by machining an aluminum AL7075-T6 (UNS A97075) workpiece in a Kondia HS1000 machining center equipped with a Siemens 840D open-architecture CNC. The workpiece is machined according to four parameters: radial depth of cut (*ae*), tool diameter (*Diam*), feed rate (*fz*) and spindle rotation speed (*ssp*). During the machining process six signals are measured: vibrations in x axis (*AcelX*), vibrations in y axis (*AcelY*), resulting vibrations (*AcelR*), force in x axis (*Fx*), force in y axis (*Fy*) and resulting force (*Fr*). For measuring the vibrations, it was used a PCB Piezotronics WJT 352B sensor, while a Kistler 9257B dynamometer was used for measuring the forces. After machining, the quality of the component is assessed using a Carl Zeiss Surfcom 130 stylus profilometer for measuring its roughness average, which is the most used index to characterize the surface roughness [7]. Depending on the measured roughness, the operator classifies the component into non-defective (compliant) or defective (non-compliant) and places it in the corresponding stack. Then, the production cycle starts again. In addition to the machining parameters and signals measured, the operator's fatigue (*ftg*) is also monitored. Every hour the operator should indicate in a questionnaire the perceived level of fatigue in a range from 1 to 8, being 1 the lowest level and 8 the highest.

Two KPIs were used to assess the behavior and performance of the manufacturing process. The two KPIs selected were throughput (*tp*) that denotes the rate at which components are processed and scrap (*sc*) that denotes the proportion of defective components produced. Both KPIs

were recorded hourly. The following equations shows how to calculate these two variables:

$$tp = \frac{t_c}{l} \tag{3}$$

$$sc = \frac{d_c}{t_c} \tag{4}$$

where t_c is the total number of components produced in a predefined time window, d_c is the number of defective components produced in the same time window, and l is the length of the time window, whose value has been set at one hour for this study.

Data comprising 15 working days with different machining parametrizations were collected. For running and implementing the proposed methodology in a computational procedure, a personal computer equipped with an Intel Core i7–10,750 H Central Processing Unit operating at 2.6 GHz, with 16 GB DDR4 Random Access Memory and an NVIDIA GeForce RTX 2060 Graphics Processing Unit with 6 GB GDDR6 of capacity was used. The methodology was implemented on top of Scikit-learn [45], TensorFlow [1], Optuna [2], and Pymoo [10] libraries.

4. Results and discussion

After collecting and conditioning data from the manufacturing process described in Section 3, the proposed methodology and the corresponding computational procedure was applied. Fig. 3 shows the distribution of features values before and after data normalization, where it can be seen how the difference in the scales is drastically reduced.

The first outcomes obtained were the selected features for each KPI which are summarized in Table 3 where the *F*-values are derived from Pearson's correlation coefficient. This measure is implemented in Scikit-learn resulting always in a non-negative value and it is used to rank the correlation of the features with the target. The *F*-value in regression is the result of a null hypothesis test where the null hypothesis is that all the regression coefficients are equal to zero, except the intercept. An *F*-test compares this model with a model that includes coefficients different than zero and decides whether these improved the predictions or not. Then, it is assumed that the larger the value obtained for a feature, the more relevant it is for the model fitting.

It is intuitive that for *tp* the feature with the most relevant one is the feed rate (*fz*) which is directly related to the machining speed, having, consequently, a very large F value compared to the other features. The selection of tool diameter (*Diam*) was also likely since it has a direct impact on the process speed given that a larger tool diameter is traduced in fewer operations to cover the same surface of a working piece. It is also comprehensible that the vibration signals (*AcelR*, *AcelX,* and *AcelY*) were selected given that a higher machining speed will be expectedly reflected as an increase in vibrations. As can be deducted from the selection of fatigue (*ftg*), the human factor is also relevant for *tp* because a fatigued operator is more likely to slower the work, which decreases the productivity of the process. In the case of *sc* is remarkable that the four parameters of the machining process (*ssp*, *ae*, *fz,* and *Diam*) have been selected, denoting the importance of a good parametrization for achieving a good quality product. Also, two process signals (*Fy* and *AcelR*) were selected, indicating that machining effects are probably better reflected in these two signals than in the others.

Table 3
Selected features for each KPI.

Features	*F*-values for *tp*	Selected for *tp* modeling	*F*-values for *sc*	Selected for *sc* modeling
AcelR	424.005	x	294.846	x
AcelX	339.367	x	147.229	
AcelY	286.907	x	234.186	
ae	25.130		466.176	x
Diam	4449.780	x	278.234	x
ftg	30.591	x	8.124	
Fr	1.017		103.049	
Fx	2.860		13.034	
Fy	0.221		449.584	x
fz	14,850.700	x	291.062	x
ssp	15.777		885.122	x

After selecting the features for each KPI, the next step of the methodology is to generate a set of models, optimize corresponding hyperparameters and then evaluate these models. Table 4 summarizes the results of the best instance of each model type obtained after completing the previously mentioned operations. The values shown for the coefficient of determination (R^2) were computed using the most general definition of R^2, also referred to as pseudo-R^2. In the case of GPR, this model was not evaluated due to the number of samples which exceeded 1500, as described in Section 2.3. The model with the lowest MSE value was selected for each KPI, which in the case of *tp* was an MLP and for *sc* was a *k*-NN. The architecture/hyperparameters of these models are summarized in Table 5.

Table 4
Test results obtained for the different models after hyperparameter optimization.

Algorithm	R^2 for *tp*	MSE for *tp*	R^2 for *sc*	MSE for *sc*
SVR	0.985	5.192	0.567	1×10^{-4}
GPR	-	-	-	-
RF	0.981	5.319	0.625	9×10^{-5}
RR	0.956	6.943	0.397	1.2×10^{-4}
LR	0.949	8.136	0.201	1.5×10^{-4}
k-NN	0.979	6.737	0.735	6×10^{-5}
MLP	0.999	2.585	0.698	7×10^{-5}
CNN	0.998	3.189	0.731	6×10^{-5}

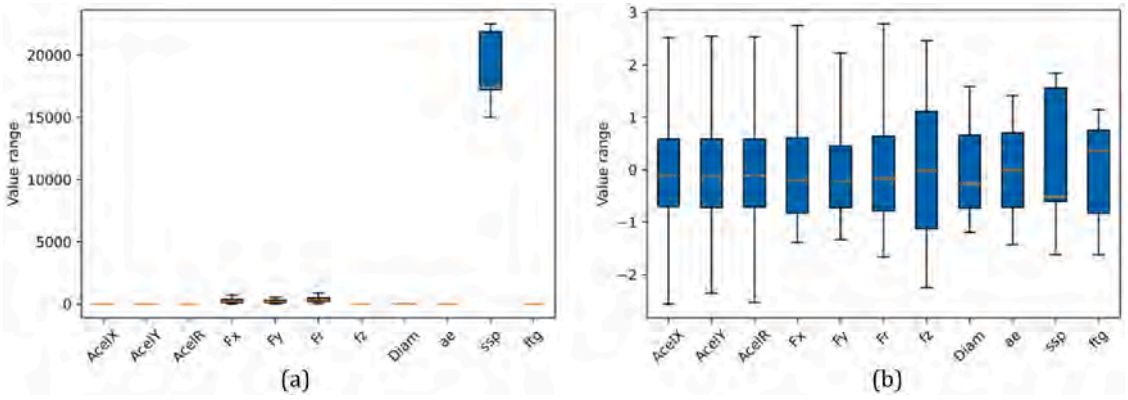

Fig. 3. Data distribution. (a) before normalization, (b) after normalization.

Table 5
Selected models hyperparameters.

KPI	Algorithm	Architecture/Hyperparameters
tp	MLP	Number of hidden layers:2 Number of units in the 1st hidden layer: 128 Activation function of the 1st hidden layer: tanh Number of units of the 2nd hidden layer: 64 Activation function of the 2nd hidden layer: tanh Number of units in the output layer: 1 Activation function of the output layer: linear Initializer: he (uniform)
sc	*k*-NN	Number of neighbors: 99 Algorithm: kd tree

In order to analyze the effectiveness of the proposed methodology with regard to other state-of-the-art AutoML frameworks, a comparison is conducted and results are shown in Table 6. For both KPIs, the results obtained are very similar. It is important to remark the complexity of the solutions offered by some of these frameworks, for instance, the stacked ensemble generated by H2O for *tp* including up to 74 base models or the ensemble generated by auto-sklearn for *sc* including up to seven base models. In the cases of auto-sklearn, TPOT and H2O, the frameworks were configured to have an execution time of approximately 15 min, similar to the execution time of the computational procedure that implements the proposed methodology.

Finally, the last step of the proposed methodology consists of using the generated models in a multi-objective optimization of the production process. In this step, only the adjustable parameters are optimized, while non-modifiable parameters are constrained to be equal to their mean values in the training set. It should be noticed that the definition of the optimization problem is one of the first steps of the methodology; which allows to carry out the process optimization automatically without any intervention after selecting the models for all the KPIs. The following equations describe the optimization problem for the case study:

$$\mathrm{Max} M_1(x[F_1]) \tag{5}$$

$$\mathrm{Min} M_2(x[F_2]) \tag{6}$$

$$\text{subject to} AcelR = \overline{AcelR_{tr}} \tag{7}$$

$$AcelX = \overline{AcelX_{tr}} \tag{8}$$

$$AcelY = \overline{AcelY_{tr}} \tag{9}$$

$$ftg = \overline{ftg_{tr}} \tag{10}$$

$$Fr = \overline{Fr_{tr}} \tag{11}$$

$$Fx = \overline{Fx_{tr}} \tag{12}$$

$$Fy = \overline{Fy_{tr}} \tag{13}$$

$$ae \in \{1,\ 2,\ 3,\ 4,\ 5\}\ mm \tag{14}$$

$$Diam \in \{8,\ 10,\ 12,\ 16,\ 20\}\ mm \tag{15}$$

$$0.025\ m/min \leq fz \leq 0.13\ m/min \tag{16}$$

$$15,000\ rpm \leq ssp \leq 22,500\ rpm \tag{17}$$

$$M_1(x[F_1]) \geq 0 \tag{18}$$

$$M_2(x[F_2]) \geq 0 \tag{19}$$

where M_1 and M_2 represent the models previously selected by the methodology for *tp* and *sc*, respectively. In this case, $x[F_1]$ and $x[F_2]$ represent the selected features for each KPI. Eqs. (5) and (6) are the objectives of the problem and the remaining Equations represent the constraints. Eqs. (7)–(13) are equality constraints for the non-controllable parameters of the process, Eqs. (14)–(17) are inequality constraints for the controllable parameters where their feasible range is specified, and Eqs. (18) and (19) are inequality constraints for the models since the two KPIs of the case study are non-negative.

The Pareto-optimal parametrizations of the production process are generated via the R-NSGA-II algorithm. The obtained results are shown in Fig. 4. For illustration purposes on convergence, an optimization of the production process by means of the NSGA-II algorithm was also carried out, using the same hyperparameters than R-NSGA-II, except for population and offspring sizes, which were set to 1000. Additionally, the performance of the process under the parametrization recommended by the plant manager before the development of this study was also included as a baseline.

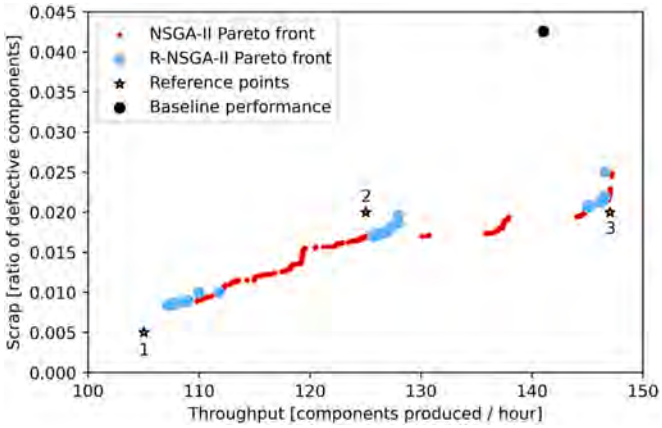

Fig. 4. Results for reference values at (105, 0.005), (125, 0.02) and (147, 0.02).

Table 6
Comparison of the modeling task.

AutoML framework	Best model type for *tp*	R^2 for *tp*	MSE for *tp*	Best model type for *sc*	R^2 for *sc*	MSE for *sc*
auto-sklearn [23]	ensemble: 3 gradient boosting regressors +1 automatic relevance determination regressor	0.989	5.036	ensemble: 2 gradient boosting regressors + 1 extremely randomized trees regressor + 1 adaptive boosting regressor + 1 *k*-NN regressor + 1 MLP model + 1 automatic relevance determination regressor	0.756	6×10^{-5}
TPOT [33]	extreme gradient boosting regressor	0.995	2.609	extremely randomized trees regressor	0.727	6×10^{-5}
H2O [34]	stacked ensemble: (74 base models)	0.986	6.048	stacked ensemble: 1 gradient boosting regressor + 1 extreme gradient boosting regressor + 1 distributed RF regressor	0.748	6×10^{-5}
Proposed methodology	MLP model	0.999	2.585	*k*-NN regressor	0.735	6×10^{-5}

Based on the results achieved, the production process could be parametrized to work in desired regimes near the reference points. The solutions associated to reference 1 in Fig. 4 reduce *sc* by 3.36 % on average compared to the baseline performance, at the cost of decreasing *tp* by 22.69 % on average. In the case of the solutions associated to reference 2, they enable reducing *sc* by 2.46 % on average compared to the baseline performance, at the cost of decreasing *tp* by 9.57 % on average. Finally, it can be seen how the solutions associated to reference 3 dominate the baseline performance. In this case, *sc* is reduced by 2.15 % on average compared to the baseline performance, while increasing *tp* by 3.19 % on average. With these results, a technologist, plant manager or any designated person in the SME can decide on whether to adopt a solution that improves both KPIs or, if reduction of defects is critical, to adopt a solution that implies decrease the productivity but minimize the number of defective components. For testing purposes, one of the parametrizations of the Pareto set associated to reference 3 was applied to the production process, resulting in the following operating conditions: $tp = 144.6$ component produced/hour and $sc = 0.024$. These results are in line with the values predicted in the Pareto front.

As can be seen in this case study, the proposed methodology allowed an SME without in-house ML expertise to create accurate models for two KPIs of its manufacturing process and automatically use them to find optimal parametrizations of the production.This was possible thanks to the holistic approach of the methodology that extends the classic AutoML workflow to include an automated optimization stage where the previously created models are used as objective functions without the need for manual intervention. In addition, the capabilities of the optimization algorithm allowed to consider the operator preferences during the generation of the optimal parametrizations, streamlining the decision-making process.

5. Conclusions

This work presents an automated machine learning methodology for optimizing manufacturing processes in SMEs by combining the standard tasks of AutoML tools, such as data preprocessing, feature selection, model training, and hyperparameter optimization, with preference-based multi-objective optimization. For this purpose, the basic AutoML workflow is used to generate models for each of the KPIs of the production process and, then, a new automated optimization step is introduced for using the generated models as objective functions, resulting in optimal parametrizations of the production process. By simplifying the way of interacting with this methodology, it is possible that manufacturing SMEs with low availability of highly-skilled personnel or limited computing power can benefit from advanced technologies making easier the digitalization and application of Industry 4.0 paradigm.

The methodology was implemented and validated in a production process where, firstly, the most relevant features for modeling each key performance indicator were automatically selected based on the Pearson's correlation coefficient, allowing to reduce the dimensionality of the data. Then, models of key performance indicators were generated and their architecture/hyperparameters optimized. Generated models were compared to models obtained through other AutoML frameworks offering similar results, with values of $MSE = 2.585$ and $R^2 = 0.999$, and $MSE = 6 \times 10^{-5}$ and $R^2 = 0.735$, respectively. Finally, the models were used as objective functions in the R-NSGA-II algorithm for finding optimal parametrizations of the production process, yielding an improvement in both KPI, reducing scrap by 2.15 % and increasing throughput by 3.19 %, with regard to the baseline of conventional parametrization considering only a single productivity target. These improvements contribute to a higher production rate while, at the same time, the number of defective components is reduced, which underscore the potential of the proposed methodology to significantly boost overall efficiency and profitability for SMEs by optimizing their production processes more holistically.

In the future, the study will be extended by including a meta-heuristic algorithm for initializing the architecture and hyper-parameters of the models. By exploring promising hyperparameters and architectures first, derived from similar problems, the model selection process will be more efficient, thus positively impacting the performance of the entire AutoML workflow.

Furthermore, we intend to apply the proposed methodology to other domains beyond manufacturing SMEs. Potential domains include but are not limited to construction, infrastructure services, logistics, healthcare, and finance. Many problems in these fields also require the creation of models and its use for optimization, which paves the way for applying the proposed methodology. However, its suitability to different domains must be carefully evaluated, recognizing that each field present unique challenges and constraints.

Funding

This work has been funded by European Commission trough the Horizon Europe project "Multi-Modal and Multi-Aspect Holistic Human-Robot Interaction (FORTIS)", grant ID 101135707. This work was also partially funded by Ministerio de Ciencia, Innovación y Universidades (MICIU) of Spain trough the project "Self-reconfiguration for Industrial Cyber-Physical Systems based on digital twins and Artificial Intelligence. Methods and application in Industry 4.0 pilot line (SELFRECO)", grant ID PID2021–127763OB-100". This work was also supported in part by the project "Digitalization of Power Electronic Applications within Key Technology Value Chains (PowerizeD)", grant ID 101096387, funded by the European Union HORIZON Framework Programme and Chips JU.

CRediT authorship contribution statement

Yarens J. Cruz: Conceptualization, Investigation, Methodology, Software, Writing – original draft. **Alberto Villalonga:** Formal analysis, Software, Validation, Visualization. **Fernando Castaño:** Conceptualization, Data curation, Investigation, Validation. **Marcelino Rivas:** Conceptualization, Formal analysis, Writing – original draft. **Rodolfo E. Haber:** Funding acquisition, Methodology, Project administration, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data that has been used is confidential.

References

[1] Abadi M., Agarwal A., Barham P., Brevdo E., Chen Z., Citro C., Corrado G. S., Davis A., Dean J., Devin M., Ghemawat S., Goodfellow I., Harp A., Irving G., Isard M., Jozefowicz R., Jia Y., Kaiser L., Kudlur M., Levenberg J., Mané D., Schuster M., Monga R., Moore S., Murray D., Olah C., Shlens J., Steiner B., Sutskever I., Talwar K., Tucker P., Vanhoucke V., Vasudevan V., Viégas F., Vinyals O., Warden P., Wattenberg M., Wicke M., Yu Y., & Zheng X. (2015). TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. Retrieved from https://www.tensorflow.org/.

[2] Akiba T, Sano S, Yanase T, Ohta T, Koyama M. Optuna: a next-generation hyperparameter optimization framework. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining (KDD '19); 2019. p. 2623–31. https://doi.org/10.1145/3292500.3330701.

[3] Al-Tashi Q, Abdulkadir SJ, Rais HM, Mirjalili S, Alhussian H. Approaches to multi-objective feature selection: a systematic literature review. IEEE Access 2020;8: 125076–96. https://doi.org/10.1109/ACCESS.2020.3007291.

[4] Alonso R, Haber RE, Castaño F, Reforgiato Recupero D. Interoperable software platforms for introducing artificial intelligence components in manufacturing: a

meta-framework for security and privacy. Heliyon 2024;10(4):e26446. https://doi.org/10.1016/j.heliyon.2024.e26446.
[5] Ang KM, Lim WH, Isa NAM, Tiang SS, Wong CH. A constrained multi-swarm particle swarm optimization without velocity for constrained optimization problems. Expert Syst Appl 2020;140:112882. https://doi.org/10.1016/j.eswa.2019.112882.
[6] Astolfi D, Pandit R. Multivariate wind turbine power curve model based on data clustering and polynomial LASSO regression. Appl Sci 2022;12(1). https://doi.org/10.3390/app12010072.
[7] Beruvides G, Castaño F, Haber Guerra RE, Quiza R, Villalonga A. Coping with complexity when predicting surface roughness in milling processes: hybrid incremental model with optimal parametrization. Complexity 2017;2017:1–11. https://doi.org/10.1155/2017/7317254.
[8] Beruvides G, Castaño F, Quiza R, Haber RE. Surface roughness modeling and optimization of tungsten-copper alloys in micro-milling processes. Measurement 2016;86:246–52. https://doi.org/10.1016/j.measurement.2016.03.002.
[9] Bianco GM, Giuliano R, Mazzenga F, Marrocco G. Multi-slope path loss and position estimation with grid search and experimental results. IEEE Trans Signal Inf Process Netw 2021;7:551–61. https://doi.org/10.1109/TSIPN.2021.3106693.
[10] Blank J, Deb K. Pymoo: multi-objective optimization in python. IEEE Access 2020;8:89497–509. https://doi.org/10.1109/ACCESS.2020.2990567.
[11] Cai J, Luo J, Wang S, Yang S. Feature selection in machine learning: a new perspective. Neurocomputing 2018;300:70–9. https://doi.org/10.1016/j.neucom.2017.11.077.
[12] Castano F, Cruz YJ, Villalonga A, Haber RE. Data-driven insights on time-to-failure of electromechanical manufacturing devices: a procedure and case study. IEEE Trans Industr Inform 2022:1–11. https://doi.org/10.1109/TII.2022.3216629.
[13] Chauhan K, Jani S, Thakkar D, Dave R, Bhatia J, Tanwar S, Obaidat MS. Automated machine learning: the new wave of machine learning. In: Proceedings of the 2020 2nd international conference on innovative mechanisms for industry applications (ICIMIA); 2020. p. 205–12. https://doi.org/10.1109/ICIMIA48430.2020.9074859.
[14] Cruz YJ, Rivas M, Quiza R, Beruvides G, Haber RE. Computer vision system for welding inspection of liquefied petroleum gas pressure vessels based on combined digital image processing and deep learning techniques. Sensors 2020;20(16). https://doi.org/10.3390/s20164505.
[15] Cruz YJ, Rivas M, Quiza R, Haber RE, Castaño F, Villalonga A. A two-step machine learning approach for dynamic model selection: a case study on a micro milling process. Comput Ind 2022;143:103764. https://doi.org/10.1016/j.compind.2022.103764.
[16] Cruz YJ, Rivas M, Quiza R, Villalonga A, Haber RE, Beruvides G. Ensemble of convolutional neural networks based on an evolutionary algorithm applied to an industrial welding process. Comput Ind 2021;133:103530. https://doi.org/10.1016/j.compind.2021.103530.
[17] Dastile X, Celik T, Potsane M. Statistical and machine learning models in credit scoring: a systematic literature survey. Appl Soft Comput 2020;91:106263. https://doi.org/10.1016/j.asoc.2020.106263.
[18] Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 2002;6(2):182–97. https://doi.org/10.1109/4235.996017.
[19] 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; 2006. p. 635–42. https://doi.org/10.1145/1143997.1144112.
[20] Dragicevic N, Ullrich A, Tsui E, Gronau N. A conceptual model of knowledge dynamics in the industry 4.0 smart grid scenario. Knowl Manag Res Pract 2020;18(2):199–213. https://doi.org/10.1080/14778238.2019.1633893.
[21] Ejsmont K, Gladysz B, Roczon N, Bettoni A, Barut ZM, Haber R, Minisci E. Multisided Business Model for Platform Offering AI Services. In: Soldatos J, editor. Artificial Intelligence in Manufacturing: Enabling Intelligent, Flexible and Cost-Effective Production Through. AI; 2024. p. 105–20. https://doi.org/10.1007/978-3-031-46452-2_7.
[22] Engelmann B, Schmitt S, Miller E, Bräutigam V, Schmitt J. Advances in machine learning detecting changeover processes in cyber physical production systems. J Manuf Mater Process 2020;4(4). https://doi.org/10.3390/jmmp4040108.
[23] Feurer M., Klein A., Eggensperger K., Springenberg J., Blum M., & Hutter F. (2015). Efficient and Robust Automated Machine Learning. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, & R. Garnett (Eds.), Advances in Neural Information Processing Systems 28 (NIPS 2015). Retrieved from https://proceedings.neurips.cc/paper_files/paper/2015/file/11d0e6287202fced83f79975ec59a3a6-Paper.pdf.
[24] Feurer M., Eggensperger K., Falkner S., Lindauer M., & Hutter F. (2022). Auto-Sklearn 2.0: Hands-free AutoML via Meta-Learning. Journal of Machine Learning Research, 23(261), 1-61. Retrieved from https://jmlr.org/papers/volume23/21-0992/21-0992.pdf.
[25] Gao X, Li X, Zhao B, Ji W, Jing X, He Y. Short-term electricity load forecasting model based on EMD-GRU with feature selection. Energies 2019;12(6). https://doi.org/10.3390/en12061140 (Basel).
[26] Garouani M, Ahmad A, Bouneffa M, Hamlich M. AMLBID: an auto-explained automated machine learning tool for big industrial data. SoftwareX 2022;17:100919. https://doi.org/10.1016/j.softx.2021.100919.
[27] Gladysz B, Matteri D, Ejsmont K, Corti D, Bettoni A, Haber Guerra R. Platform-based support for AI uptake by SMEs: guidelines to design service bundles. Cent Eur Manag J 2023. https://doi.org/10.1108/CEMJ-08-2022-0096. , ahead-of-p (ahead-of-print).
[28] He S, Li Z, Wang J, Xiong NN. Intelligent detection for key performance indicators in industrial-based cyber-physical systems. IEEE Trans Industr Inform 2021;17(8):5799–809. https://doi.org/10.1109/TII.2020.3036168.
[29] Imane M, Aoula ES, Achouyab EH. Using bayesian ridge regression to predict the overall equipment effectiveness performance. In: Proceedings of the 2022 2nd international conference on innovative research in applied science, engineering and technology (IRASET); 2022. p. 1–4. https://doi.org/10.1109/IRASET52964.2022.9738316.
[30] Statistical learning bt - James G, Witten D, Hastie T, Tibshirani R, Taylor J, James G, Witten D, Hastie T, Tibshirani R, Taylor J. An Introduction to statistical learning: with applications in python. Statistical learning bt -. Springer International Publishing; 2023. p. 15–67. 10.1007/978-3-031-38747-0_2.
[31] Javeed A, Zhou S, Yongjian L, Qasim I, Noor A, Nour R. An intelligent learning system based on random search algorithm and optimized random forest model for improved heart disease detection. IEEE Access 2019;7:180235–43. https://doi.org/10.1109/ACCESS.2019.2952107.
[32] Jin H, Song Q, Hu X. Auto-Keras: an efficient neural architecture search system. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery \& data mining; 2019. p. 1946–56.
[33] Le TT, Fu W, Moore JH. Scaling tree-based automated machine learning to biomedical big data with a feature set selector. Bioinformatics 2020;36(1):250–6. https://doi.org/10.1093/bioinformatics/btz470.
[34] LeDell E, Poirier S. H2O AutoML: scalable automatic machine learning. In: Proceedings of the 7th ICML workshop on automated machine learning (AutoML); 2020. https://www.automl.org/wp-content/uploads/2020/07/AutoML_2020_paper_61.pdf.
[35] Lin YK, Chang PC, Yeng LCL, Huang SF. Bi-objective optimization for a multistate job-shop production network using NSGA-II and TOPSIS. J Manuf Syst 2019;52:43–54. https://doi.org/10.1016/j.jmsy.2019.05.004.
[36] López-Ibáñez M, Dubois-Lacoste J, Pérez Cáceres L, Birattari M, Stützle T. The irace package: iterated racing for automatic algorithm configuration. Oper Res Perspect 2016;3:43–58. https://doi.org/10.1016/j.orp.2016.09.002.
[37] Manimuthu A, Venkatesh VG, Shi Y, Sreedharan VR, Koh SCL. Design and development of automobile assembly model using federated artificial intelligence with smart contract. Int J Prod Res 2022;60(1):111–35. https://doi.org/10.1080/00207543.2021.1988750.
[38] Mezzogori D, Romagnoli G, Zammori F. Defining accurate delivery dates in make to order job-shops managed by workload control. Flex Serv Manuf J 2021;33(4):956–91. https://doi.org/10.1007/s10696-020-09396-2.
[39] Mukkoti VV, Mohanty CP, Gandla S, Sarkar P, P SR, B D. Optimization of process parameters in CNC milling of P20 steel by cryo-treated tungsten carbide tools using NSGA-II. Prod Manuf Res 2020;8(1):291–312. https://doi.org/10.1080/21693277.2020.1790436.
[40] Nikitin NO, Vychuzhanin P, Sarafanov M, Polonskaia IS, Revin I, Barabanova IV, Maximov G, Kalyuzhnaya AV, Boukhanovsky A. Automated evolutionary approach for the design of composite machine learning pipelines. Future Gener Comput Syst 2022;127:109–25. https://doi.org/10.1016/j.future.2021.08.022.
[41] : enabling intelligent, flexible and cost-effective production through ai Nizamis A, Schlake G, Siachamis G, Dimitriadis V, Patsonakis C, Beecks C, Ioannidis D, Votis K, Tzovaras D, Soldatos J. Designing a marketplace to exchange ai models for industry 5.0 bt. Artificial Intelligence in manufacturing. Springer Nature Switzerland; 2024. p. 27–41. enabling intelligent, flexible and cost-effective production through ai, 10.1007/978-3-031-46452-2_2.
[42] Omar YM, Minoufekr M, Plapper P. Business analytics in manufacturing: current trends, challenges and pathway to market leadership. Oper Res Perspect 2019;6:100127. https://doi.org/10.1016/j.orp.2019.100127.
[43] Saleeshya PG, Binu M. A neuro-fuzzy hybrid model for assessing leanness of manufacturing systems. Int J Lean Six Sigma 2019;10(1):473–99. https://doi.org/10.1108/IJLSS-05-2017-0040.
[44] Pai KN, Prasad V, Rajendran A. Experimentally validated machine learning frameworks for accelerated prediction of cyclic steady state and optimization of pressure swing adsorption processes. Sep Purif Technol 2020;241:116651. https://doi.org/10.1016/j.seppur.2020.116651.
[45] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E. Scikit-learn: machine learning in python. J Mach Learn Res 2011;12:2825–30.
[46] Peres RS, Dionisio Rocha A, Leitao P, Barata J. IDARTS – Towards intelligent data analysis and real-time supervision for industry 4.0. Comput Ind 2018;101:138–46. https://doi.org/10.1016/j.compind.2018.07.004.
[47] Petridis K, Tampakoudis I, Drogalas G, Kiosses N. A support vector machine model for classification of efficiency: an application to M&A. Res Int Bus Finance 2022;61:101633. https://doi.org/10.1016/j.ribaf.2022.101633.
[48] Romano JD, Le TT, Fu W, Moore JH. TPOT-NN: augmenting tree-based automated machine learning with neural network estimators. Genet Program Evolvable Mach 2021;22(2):207–27. https://doi.org/10.1007/s10710-021-09401-z.
[49] Schaefer JL, Nara EOB, Siluk JCM, Baierle IC, Costa MBDa, Furtado JC. Competitiveness metrics for small and medium-sized enterprises through multi-criteria decision making methods and neural networks. Int J Process Manag Benchmarking 2022;12(2):184–207. https://doi.org/10.1504/IJPMB.2022.121599.
[50] Sevinç A, Gür Ş, Eren T. Analysis of the Difficulties of SMEs in Industry 4.0 Applications by Analytical Hierarchy Process and Analytical Network Process. Processes 2018;6(12). https://doi.org/10.3390/pr6120264.
[51] Shi Y, Liu D. Relationship between urban new business indexes and the business environment of chinese cities: a study based on entropy-TOPSIS and a gaussian process regression model. In Sustainability 2020;12(24). https://doi.org/10.3390/su122410422.

[52] Solorio-Fernández S, Carrasco-Ochoa JA, Martínez-Trinidad JF. A review of unsupervised feature selection methods. Artif Intell Rev 2020;53(2):907–48. https://doi.org/10.1007/s10462-019-09682-y.
[53] Stühler H, Zöller MA, Klau D, Beiderwellen-Bedrikow A, Tutschku C. Benchmarking automated machine learning methods for price forecasting applications. In: Proceedings of the 12th international conference on data science, technology and applications - DATA; 2023. p. 30–9. https://doi.org/10.5220/0012051400003541.
[54] Sun Q, Ge Z. Deep learning for industrial KPI prediction: when ensemble learning meets semi-supervised data. IEEE Trans Industr Inform 2021;17(1):260–9. https://doi.org/10.1109/TII.2020.2969709.
[55] Truong A, Walters A, Goodsitt J, Hines K, Bruss CB, Farivar R. Towards automated machine learning: evaluation and comparison of AutoML approaches and tools. In: Proceedings of the 2019 IEEE 31st international conference on tools with artificial intelligence (ICTAI); 2019. p. 1471–9. https://doi.org/10.1109/ICTAI.2019.00209.
[56] Usuga Cadavid JP, Lamouri S, Grabot B, Pellerin R, Fortin A. Machine learning applied in production planning and control: a state-of-the-art in the era of industry 4.0. J Intell Manuf 2020;31(6):1531–58. https://doi.org/10.1007/s10845-019-01531-7.
[57] Wang T, Wang X, Ma R, Li X, Hu X, Chan FTS, Ruan J. Random forest-bayesian optimization for product quality prediction with large-scale dimensions in process industrial cyber–physical systems. IEEE Internet Things J 2020;7(9):8641–53. https://doi.org/10.1109/JIOT.2020.2992811.
[58] Wang ZJ, Zhan ZH, Kwong S, Jin H, Zhang J. Adaptive granularity learning distributed particle swarm optimization for large-scale optimization. IEEE Trans Cybern 2021;51(3):1175–88. https://doi.org/10.1109/TCYB.2020.2977956.
[59] Waring J, Lindvall C, Umeton R. Automated machine learning: review of the state-of-the-art and opportunities for healthcare. Artif Intell Med 2020;104:101822. https://doi.org/10.1016/j.artmed.2020.101822.
[60] Wei G, Wu Q, Zhou M. A Hybrid Probabilistic multiobjective evolutionary algorithm for commercial recommendation systems. IEEE Trans Comput Soc Syst 2021;8(3):589–98. https://doi.org/10.1109/TCSS.2021.3055823.
[61] Xavier-Júnior JC, Freitas AA, Ludermir TB, Feitosa-Neto A, Barreto CAS. An evolutionary algorithm for automated machine learning focusing on classifier ensembles: an improved algorithm and extended results. Theor Comput Sci 2020;805:1–18. https://doi.org/10.1016/j.tcs.2019.12.002.
[62] Xu Y, Guan G, Song Q, Jiang C, Wang L. Heuristic and random search algorithm in optimization of route planning for Robot's geomagnetic navigation. Comput Commun 2020;154:12–7. https://doi.org/10.1016/j.comcom.2020.02.043.
[63] Yadav A, Mishra S, Sairam AS. A multi-objective worker selection scheme in crowdsourced platforms using NSGA-II. Expert Syst Appl 2022;201:116991. https://doi.org/10.1016/j.eswa.2022.116991.
[64] Yang Y, Nam A, Nasr-Azadani M, Tung T. Resource-aware pareto-optimal automated machine learning platform. In: Proceedings of the 2020 3rd international seminar on research of information technology and intelligent systems (ISRITI); 2020. p. 1–6. https://doi.org/10.1109/ISRITI51436.2020.9315336.
[65] Yao L, Fang Z, Xiao Y, Hou J, Fu Z. An intelligent fault diagnosis method for lithium battery systems based on grid search support vector machine. Energy 2021;214:118866. https://doi.org/10.1016/j.energy.2020.118866.
[66] Zagumennov F, Bystrov A, Radaykin A. In-firm planning and business processes management using deep neural network. GATR J Bus Econ Rev 2021;6(3):203–11. https://doi.org/10.35609/jber.2021.6.3(4.
[67] Zahar Djordjevic M, Djordjevic A, Klochkova E, Misic M. Application of modern digital systems and approaches to business process management. Sustainability 2022;14(3). https://doi.org/10.3390/su14031697.
[68] Zhang J, Wang Q, Shen W. Hyper-parameter optimization of multiple machine learning algorithms for molecular property prediction using hyperopt library. Chin J Chem Eng 2022. https://doi.org/10.1016/j.cjche.2022.04.004.
[69] Zhang K, Zheng L, Liu Z, Jia N. A deep learning based multitask model for network-wide traffic speed prediction. Neurocomputing 2020;396:438–50. https://doi.org/10.1016/j.neucom.2018.10.097.
[70] Zimmer L, Lindauer M, Hutter F. Auto-Pytorch: Multi-Fidelity MetaLearning for Efficient and Robust AutoDL. IEEE Trans Pattern Anal Mach Intell 2021;43(9):3079–90. https://doi.org/10.1109/TPAMI.2021.3067763.
[71] Zöller MA, Huber MF. Benchmark and survey of automated machine learning frameworks. J Artif Intell Res 2021;70:409–72. https://doi.org/10.1613/jair.1.11854.