

Self-Reconfiguration for Smart Manufacturing Based on Artificial Intelligence: A Review and Case Study



Yarens J. Cruz , Fernando Castaño , Rodolfo E. Haber ,
Alberto Villalonga , Krzysztof Ejsmont , Bartłomiej Gładysz ,
Álvaro Flores , and Patricio Alemany

1 Introduction

In the context of manufacturing systems, reconfiguration refers to the practice of changing a production system or process to meet new needs or to improve its performance. This might involve varying the structure of the production process, the order of the steps in which operations are executed, or the manufacturing process itself to make a different product.

Reconfiguration may be necessary for several reasons, including changes in raw material availability or price, changes in consumer demand for a product, the need to boost productivity, save costs, or improve product quality, among others. It is a complex process that requires careful planning and coordination to ensure that production is not disrupted and that the changes result in the desired outcomes. In return, it may offer substantial advantages including enhanced product quality, reduction of waste, and greater productivity, making it a crucial strategy for enterprises trying to maintain their competitiveness in a rapidly changing market.

Y. J. Cruz (✉) · F. Castaño · R. E. Haber · A. Villalonga
Centro de Automática y Robótica (CSIC-Universidad Politécnica de Madrid), Madrid, Spain
e-mail: y.cruz@car.upm-csic.es; fernando.castano@car.upm-csic.es;
rodolfo.haber@car.upm-csic.es; alberto.villalonga@car.upm-csic.es

K. Ejsmont · B. Gładysz
Institute of Production Systems Organization, Faculty of Mechanical and Industrial Engineering,
Warsaw University of Technology, Warsaw, Poland
e-mail: krzysztof.ejsmont@pw.edu.pl; bartlomiej.gladysz@pw.edu.pl

Á. Flores · P. Alemany
Rovimática SL, Córdoba, Spain
e-mail: alvaro.flores@rovimatica.eu; patricio.alemany@rovimatica.eu

Self-reconfiguration is the capacity of a manufacturing system to autonomously modify its configuration or structure to respond to dynamic requirements. This concept is frequently linked to the development of modular and adaptive manufacturing systems. These systems exhibit high flexibility, efficiency, and adaptability by allowing the self-reconfiguration of their assets. However, self-reconfiguration is not directly applicable to all manufacturing systems. To implement self-reconfiguration, a particular level of technological maturity is required, including the following requirements [1]:

- *Modularity*: The system is made up of a collection of standalone components.
- *Integrability*: The components have standard interfaces that facilitate their integration into the system.
- *Convertibility*: The structure of the system can be modified by adding, deleting, or replacing individual components.
- *Diagnosability*: The system has a mechanism for identifying the status of the components.
- *Customizability*: The structure of the system can be changed to fit specific requirements.
- *Automatability*: The system operation and modifications can be carried out without human intervention.

Additionally, self-reconfiguration may involve a variety of techniques and technologies, including IT infrastructure, robotic systems, intelligent sensors, and advanced control algorithms. These technologies enable machines to automatically identify and select the appropriate components or configurations needed to complete a given task, without requiring manual intervention or reprogramming. However, in some practical scenarios, human validation is still required before executing the reconfiguration.

Self-reconfiguration in manufacturing typically focuses on process reconfiguration and capacity reconfiguration with success stories in the automotive industry. Process reconfiguration involves changes in the manufacturing process itself, such as changing the sequence of operations or the layout of the production line, as well as modifications to the equipment. On the other hand, capacity reconfiguration involves adjusting the capacity of the manufacturing system to meet changes in demand. This may involve adding or removing production lines, or modifying the parameters of machines. It should be noted that modifying the parameters of existing equipment can increase production throughput without requiring significant capital investment; however, it may also require changes to the production process, such as modifying the material flow or introducing new quality control measures.

2 Reconfiguration in Manufacturing

2.1 Precursors of Reconfigurable Systems: Flexible Manufacturing Systems

Current self-reconfigurable manufacturing systems are the result of the evolution of ideas that emerged more than 50 years ago. During the 1960s and 1970s, the production methods were primarily intended for mass production of a limited range of products [2]. Due to their rigidity, these systems needed a significant investment of time and resources to be reconfigured for a different product. During that period, supported by the rapid advancements and affordability of computer technology, the concept of flexible manufacturing system (FMS) emerged as a solution to address this scenario [3]. FMSs are versatile manufacturing systems, capable of producing a diverse array of products utilizing shared production equipment. These systems are characterized by high levels of automation and computer control, enabling seamless adaptation for manufacturing different goods or products.

FMSs typically consist of a series of integrated workstations, each containing a combination of assets. These workstations are connected by computer-controlled transport systems that can move raw materials, workpieces, and finished products between workstations. When FMSs were introduced, they were primarily focused on achieving reconfigurability through the use of programmable controllers and interchangeable tooling. These systems may be configured to carry out a variety of manufacturing operations such as milling, drilling, turning, and welding. FMSs can also incorporate technologies such as computer-aided design/manufacturing (CAD/CAM) and computer numerical control (CNC) to improve efficiency and quality. This paradigm has been widely adopted in industries such as automotive [4], aerospace [5], and electronics [6] and continues to evolve with advances in technology.

However, despite the adaptability to produce different products, the implementation of FMSs has encountered certain drawbacks such as lower throughput, high equipment cost due to redundant flexibility, and complex design [7]. In addition, they have fixed hardware and fixed (although programmable) software, resulting in limited capabilities for updating, add-ons, customization, and changes in production capacity [3].

2.2 Reconfigurable Manufacturing Systems

Although FMSs can deal with the market exigence for new products or modifications of existing products, they cannot efficiently adjust their production capacity. This means that if a manufacturing system was designed to produce a maximum number of products annually and, after 2 years, the market demand for the product is reduced to half, the factory will be idle 50% of the time, creating a big financial loss.

On the other hand, if the market demand for the product surpasses design capability and the system is unable to handle it, the financial loss can be even greater [8]. To handle such scenarios, during the 1990s, a new type of manufacturing system known as reconfigurable manufacturing system (RMS) was introduced. RMSs adhere to the typical goals of production systems: to produce with high quality and low cost. However, additionally, they also aim to respond quickly to market demand, allowing for changes in production capacity. In other words, they strive to provide the capability and functionality required at the right time [3]. This goal is achieved by enabling the addition or removal of components from production lines on demand.

Design principles such as modularity, integrability, and open architecture control systems started to take more significance with the emergence of RMSs, given the relevance of dynamic equipment interconnection in these systems [9]. Considering their advantages, RMSs have been applied to the manufacturing of medical equipment [10], automobiles [11], food and beverage [12], and so on. Because they require less investments in equipment and infrastructure, these systems often offer a more cost-effective alternative to FMSs.

Although these systems can adapt to changing production requirements, the reconfiguration decisions are usually made or supervised by a human, which means the systems cannot autonomously reconfigure themselves. This gives more control to the plant supervisor or operator, but the downside is that it limits the response speed.

2.3 Evolution Towards Self-Reconfiguration

As technology advanced and the demands of manufacturing increased, production systems began to incorporate more sophisticated sensing, control, and robotics capabilities. This allowed them to monitor and adjust production processes in real time, adapt to changes in the manufacturing environment, and even reconfigure themselves without human intervention. This shift from reconfigurable to self-reconfigurable systems was driven by several technological advancements:

- *Intelligent sensors*: sensors that are capable of not only detecting a particular physical quantity or phenomenon but also processing and analyzing the data collected to provide additional information about the system being monitored [13].
- *Adaptive control*: control systems that can automatically adjust the manufacturing process to handle changes in the production environment while maintaining optimal performance [14].
- *Autonomous robots*: robots that can move and manipulate objects, work collaboratively, and self-reconfigure. These robots can be used to assemble components, perform quality control checks, and generate useful data for reconfiguring production lines [15].

- *Additive manufacturing*: 3D printing and additive manufacturing techniques allow to create complex and customized parts and structures on demand, without the need for extensive changes in the production system. Additionally, this technique is very useful for quick prototyping [16].

Compared to conventional RMSs, self-reconfigurable manufacturing systems enable to carry out modifications to the production process in a faster and more autonomous way [17]. Today, these systems are at the cutting edge of advanced manufacturing, allowing the development of extremely complex, specialized, and efficient production systems that require little to no human involvement. Self-reconfiguration is receiving significant attention in the context of Industry 4.0, where the goal is to create smart factories that can communicate, analyze data, and optimize production processes in real time [18].

3 Current Approaches

Currently, there are several approaches for designing self-reconfiguration solutions including computer simulation, which is one of the most reported in the literature with proof-of-concepts based on simulation results. Other alternative techniques include those based on artificial intelligence (AI), which provide powerful methods and tools to deal with uncertainty, such as fuzzy and neuro-fuzzy approaches, machine learning and reinforcement learning strategies. These approaches are not mutually exclusive and, in many cases, are used in a complementary way.

3.1 Computer Simulation

Computer simulation is a particularly valuable tool for the design and optimization of self-reconfigurable manufacturing systems. In this context, these tools aim to enhance the system's responsiveness to changes in production requirements. The recent increase in computational capacities has enabled the testing of various configurations and scenarios before their actual implementation [19]. Currently, commercial applications such as AutoMod, FlexSim, Arena, Simio, and AnyLogic, among others, allow to create high-fidelity simulations of industrial processes [20], that even include three-dimensional recreations of factories for use in augmented/virtual reality applications. Computer simulation becomes a powerful tool when integrated with the production process it represents. Based on this idea, digital twins have gained significant attention in both industry and academia [21]. Digital twins enable real-time data integration from the production process into the simulation, replicating the actual production environment. By evaluating different options and identifying the optimal configuration for the new scenarios, digital twins provide feedback to the production process, facilitating real-time modifications.

3.2 *Fuzzy Systems*

Fuzzy logic is a mathematical framework that can be used to model and reason with imprecise or uncertain data. This capability makes fuzzy logic particularly useful in situations where the system may not have access to precise data or where the data may be subject to noise or other sources of uncertainty. In the context of self-reconfiguration, fuzzy systems can be used to model the behavior of the physical processes and make decisions about how to reconfigure them based on imprecise data. For instance, it is often very complex to assign a precise value to indicators such as expected market demand, product quality, or energy consumption [22]. These variables can be assigned to fuzzy membership functions and then, following predefined rules, combined using fuzzy operators to determine how the production system should be optimally reconfigured depending on the available data.

3.3 *Data-Driven Methods*

Data-driven methods deal with the collection and analysis of data, the creation of models, and their use for decision-making. This approach is extensively applied when historical data of the production process is available. By using data analytics, it is possible to identify bottlenecks or the inefficient use of assets in the production process. Also, data-driven methods make extensive use of machine learning algorithms for modeling the production process behavior [23]. Machine learning methods can be trained with datasets containing a large number of features and samples, learning to identify correlations, patterns, and anomalies that are beyond human perception [24]. Moreover, by collecting new data of the production process, machine learning models can be retrained or fine-tuned to improve their performance over time. Once the machine learning model has been trained with production data, it can be used as an objective function of an optimization algorithm to make decisions about how to reconfigure the manufacturing process to optimize desired indicators.

3.4 *Reinforcement Learning*

Reinforcement learning is a subfield of machine learning that has shown great capacity in the development of algorithms for autonomous decision-making in dynamic and complex environments. In reinforcement learning, an agent learns to make decisions based on feedback from the environment. The agent performs actions in the environment and receives feedback in the form of rewards or penalties. The goal of the agent is to maximize the cumulative reward over time by learning which actions are most likely to lead to positive outcomes. Self-

reconfigurable manufacturing systems present a unique challenge for reinforcement learning algorithms because the environment is constantly changing [25]. The agent should be able to adapt to changes in the production environment, such as changes in demand or changes in the availability of resources. The agent can learn which modules are more effective for specific tasks and reconfigure itself accordingly [18]. Another important benefit of using reinforcement learning is the ability to learn from experience. These algorithms can learn from mistakes and errors and try to avoid repeating them.

4 Lighthouse Demonstrator: GAMHE 5.0 Pilot Line

To evaluate how AI tools can be applied for self-reconfiguration in manufacturing processes and how they can be integrated with one another, an Industry 4.0 pilot line was chosen for demonstration. The selected pilot line was the GAMHE 5.0 laboratory, which simulates the slotting and engraving stages of the production process of thermal insulation panels. Figure 1 illustrates the typical workflow of the process. Initially, a robot picks up a panel and positions it in a machining center to create slots on all four sides. Subsequently, the same robot transfers the panel to a conveyor belt system that transports it to a designated location, where a second robot takes over the handling of the panel. Next, the panel is positioned in a visual inspection area by the robot. If the slotting is deemed correct, the panel is then moved to a second machining center for the engraving process. Finally, the robot transfers the panel to a stack of processed panels.

Occasionally, due to poor positioning in the slotting process, some sides of the panels are not slotted or the depth of the slot is smaller than required. In those cases, the visual inspection system should detect the irregularity and the workflow of the process should be modified to repeat the slotting process. Figure 2 illustrates this

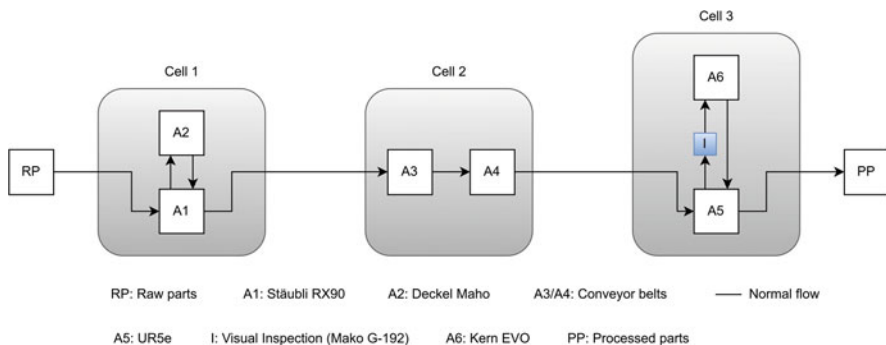


Fig. 1 Normal workflow of the pilot line

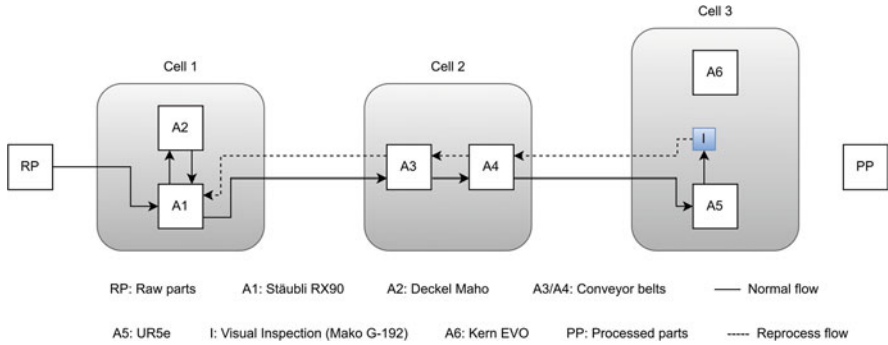


Fig. 2 Workflow for reprocessing in the pilot line

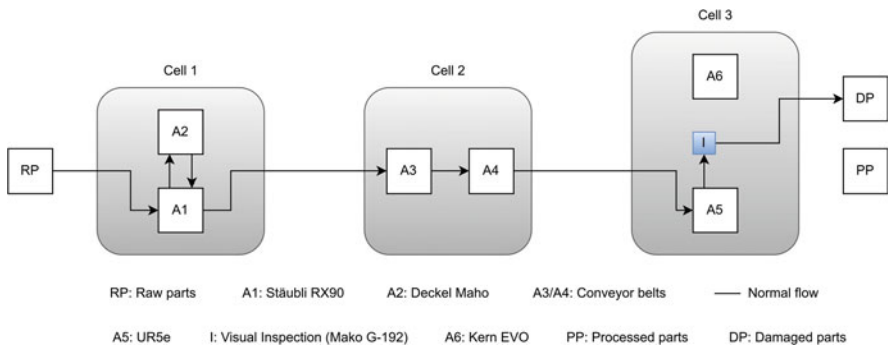


Fig. 3 Workflow for defective panels in the pilot line

situation. Once the slotting irregularities are corrected, the system continues with the normal workflow.

In some cases, the slotting process may cause damage to the panels. This can happen when working with new materials or previously unverified machining configurations. In those cases, the visual inspection system should detect that the panel is damaged and it should be sent directly to a stack of damaged parts. Figure 3 shows this situation.

Making accurate decisions about the process workflow depending on the quality of products, specifically on the result of the slotting process, has a direct impact on the productivity of the pilot line. For instance, in cases where a panel is damaged during slotting, it is crucial to remove it from the production line to prevent unnecessary time and resources from being spent on machining it during the engraving stage. To achieve this, the presence of a reliable visual inspection system becomes essential. Although a deep learning classifier could be used for this task, one drawback is that it is very hard to understand how the decision is made. For this reason, it is proposed a deep learning segmentation model, whose function is to separate the desired areas of the images from the unwanted regions.

The output of a segmentation model provides a pixel-level understanding of objects and their boundaries in an image, enabling a detailed visual interpretation of the model prediction. Then, using the segmentation result, a reasoned decision can be made, making the outcome of the system more interpretable. Section 4.1 deals with this situation.

On the other hand, a common situation is that the pilot line should deal with small batches of panels made of different materials and with different dimensions. Thus, the configuration of the assets for reaching an optimal performance varies frequently. For dealing with this situation a self-reconfiguration approach based on automated machine learning (AutoML) and fuzzy logic is proposed. Although the approach proposed in this work is generalizable to multiple objectives, for the sake of simplicity, the improvement of only one key performance indicator (KPI) will be considered. Sections 4.2 and 4.3 cover this topic.

4.1 Deep Learning-Based Visual Inspection

The segmentation model developed for application in the pilot line intends to separate the side surface of the panel from other elements within an image, allowing for later decisions on the panel quality and modifications of the process workflow. This model is based on a U-net architecture, which consists of an encoder path that gradually downsamples the input image and a corresponding decoder path that upsamples the feature maps to produce a segmentation map of the same size as the input image. This network also includes skip connections between the encoder and decoder paths that allow to retain and fuse both high-level and low-level features, facilitating accurate segmentation and object localization [26].

A dataset containing 490 images with their corresponding masks was prepared for training and evaluating the model. The image dataset was split into three subsets: training (70% of the data), validation (15% of the data), and testing (15% of the data). In this case, the validation subset serves the objective of facilitating early stopping during training. This means that if the model's performance evaluated on the validation subset fails to improve after a predetermined number of epochs, the training process is halted. By employing this technique, overfitting can be effectively mitigated and the training time can be significantly reduced.

A second version of the dataset was prepared by applying data augmentation to the training set while keeping the validation and test sets unchanged. The dataset was augmented using four transformations: horizontal flip, coarse dropout, random brightness, and random contrast. This helps increase the number of training examples, improving the model's prediction capability and making it more robust to noise. Using the two versions of the dataset, two models with the same architecture were trained. Table 1 presents the output of the two models for three examples taken from the test set. As can be observed, the predictions obtained with the model trained on the augmented dataset are significantly better than those obtained with the model

Table 1 Segmentation results for sample images




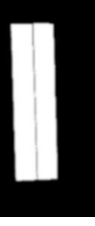




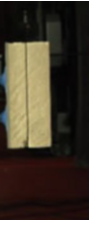



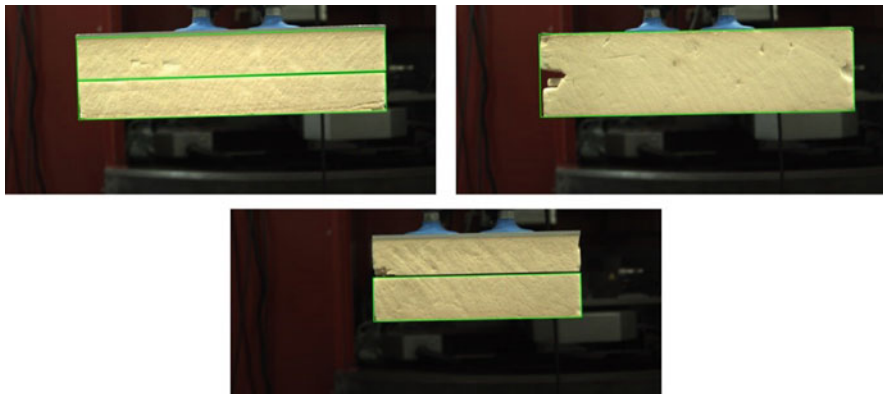
Sample	Original image	Ground truth	Prediction of the model trained on the original dataset	Prediction of the model trained on the augmented dataset
1				
2				
3				

Table 2 Metric values obtained by segmentation models on the test set

Model	Accuracy	F1 score	Jaccard index	Precision	Recall
Trained on the original dataset	0.892	0.639	0.563	0.997	0.564
Trained on the augmented dataset	0.995	0.992	0.984	0.99	0.993

**Fig. 4** Squared contours detection for a compliant panel, a panel without slot, and a damaged panel, respectively

trained on the original dataset. This is also confirmed by the values obtained in several metrics, which are shown in Table 2.

After the image is segmented by the deep learning model, a second algorithm is used. Here, a convex hull is adjusted to each separate contour in the segmented image. Then, a polygonal curve is generated for each convex hull with a precision smaller than 1.5% of the perimeter of the segmented contour. Finally, if the polygonal curve has four sides, it is drawn over the original image. After this procedure, if two rectangles were drawn over the image it is assumed that the slotting was correct and the panel did not suffer any significant damage; thus, it can be sent to the next stage of the line. On the other hand, if only one rectangle was drawn, it is assumed that the slotting was not carried out or the panel was damaged during this process. Figure 4 shows the results obtained for illustrative cases of a compliant panel, a panel with missing slots, and a damaged panel, respectively. If only one rectangle was drawn, depending on its size and location, the panel will be sent to the slotting stage again or removed from the line. This method was applied to the test set images and in all the cases the output produced matched the expected output.

4.2 Automating the Machine Learning Workflow

As outlined in the previous sections, the working conditions of the pilot line are subject to rapid variations. To effectively address these variations and generate

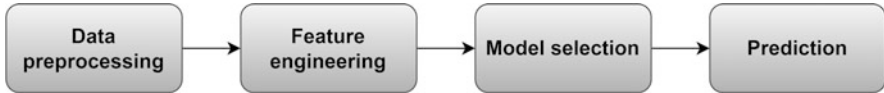


Fig. 5 General machine learning steps

optimal parametrizations for the assets, machine learning emerges as a promising tool.

The usual machine learning workflow is composed of a series of steps that are executed one by one by a team of specialists. However, this workflow can be automated. This research area is known as AutoML and recently has gained considerable attention. AutoML plays a crucial role in streamlining workflows, saving time, and reducing the effort required for repetitive tasks, thereby enabling the creation of solutions even for nonexperts. Noteworthy tools in this domain include Google Cloud AutoML, auto-sklearn, Auto-Keras, and Azure AutoML, among others. Typically, these tools encompass various stages, from data preprocessing to model selection. Moreover, in line with the automation philosophy of these systems, the process optimization step can also be integrated. This way the system would receive a dataset and return the parameter values that make the process work in a desired regime. Considering this idea, an end-to-end AutoML solution has been developed to be applied to GAMHE 5.0 pilot line. The following subsections describe the typical machine learning workflow, as well as the specificities of its different steps and how AutoML can be used for optimizing the production process.

4.2.1 Typical Machine Learning Workflow

Machine learning aims to create accurate and reliable models capable of identifying complex patterns in data. The creation and exploitation of these models is typically achieved through a series of steps that involve preparing the dataset, transforming the data to enhance its quality and relevance, selecting and training an appropriate machine learning model, evaluating the model's performance, and deploying the model in a real-world setting. Figure 5 depicts these steps. By following this workflow, machine learning practitioners can build models that harness the power of data-driven learning, enabling them to effectively derive meaningful insights and make accurate predictions in practical applications.

Data Preprocessing

Data preprocessing is the initial step in the creation of a machine learning system. The data to be used may have a variety of sources and formats, thus it should be prepared before being used by any algorithm. If data are originated from different sources, it must be merged into a single dataset. Furthermore, most methods are

not designed to work with missing data, so it is very common to remove samples with missing information. Preprocessing may also include filtering data to remove noise, which can result later in more robust models. In this stage, the data may be transformed to a format that is suitable for analysis, which can include operations such as normalization, bucketizing, and encoding. Finally, one common operation carried out in this stage is splitting. This refers to the partition of the dataset into two subsets, which will be used for training and evaluation purposes. Additionally, a third subset can be created if it is planned to carry out a hyperparameter optimization or neural architecture search over the model.

Feature Engineering

The goal of the feature engineering stage is to convert raw data into relevant features that contain the necessary information to create high-quality models. One of the most interesting techniques that can be used in this stage is feature selection. Feature selection aims to determine which features are the best predictors for a certain output variable. Then, when these features are selected, they can be extracted from the original dataset to build a lower dimensional dataset, allowing to build more compact models with better generalization ability and reduced computational time [27, 28]. Typically, for problems with numerical input and output variables, Pearson's [29] or Spearman's correlation coefficients [30] are used. If the input is numerical but the output is categorical, then the analysis of variance (ANOVA) [31] or Kendall's rank coefficient [32] are employed. Other situations may require the use Chi-squared test or mutual information measure [33].

Other techniques that can be applied in the feature engineering stage include feature creation and dimensionality reduction. Feature creation implies creating new features either by combining the existing ones or by using domain knowledge [34]. On the other hand, dimensionality reduction techniques such as principal component analysis (PCA) or t-distributed stochastic neighbor embedding (t-SNE) algorithms are used to map the current data in lower dimensional space while retaining as much information as possible [35].

Model Selection

The model selection step implies the creation, training, and evaluation of different types of models to, in the end, select the most suitable for the current situation. This practice is carried out since it does not exist a methodology for determining a priori which algorithm is better for solving a problem [36]. Therefore, the most adequate model may vary from one application to another as in the following cases: long short-term memory network (LSTM) [37], multilayer perceptron (MLP) [38], support vector regression (SVR) [39], Gaussian process regression (GPR) [40], convolutional neural network (CNN) [41], gradient boosted trees (GBT) [42]. The number and types of models to explore in this stage will depend on

the characteristics of the problem and the available computational resources. The selection of the model is carried out taking into consideration one or more metrics. For regression problems is common to rely on the coefficient of determination (R^2), mean squared error (MSE), and mean absolute percentage error ($MAPE$), among other metrics [43]. On the other hand, for classification problems, typical metrics are accuracy, recall, precision, F1-score, and so on.

Optionally, this stage can also include hyperparameter optimization. Hyperparameters determine a model's behavior during training and, in some cases, also how its internal structure is built. They are set before a model is trained and cannot be modified during training. The selection of these values can greatly affect a model's performance. However, finding an optimal or near-optimal combination of hyperparameters is not a trivial task and, usually, it is computationally intensive. The most commonly used techniques for this task include grid search, random search, Bayesian optimization, and so on.

4.2.2 Process Optimization

Once a model has been created for representing a process, it can be used for optimizing it. Assuming the model exhibits robust predictive capabilities and the constraints are accurately defined, various input values can be evaluated in the model to determine how the system would respond, eliminating the need for conducting exhaustive tests on the actual system. In other words, the model created can be embedded as the objective function of an optimization algorithm for finding the input values that would make the production process work in a desired regime. In this context, popular strategies such as particle swarm optimization [44], simulated annealing [45], evolutionary computation [46], and Nelder-Mead [47], among others, are commonly employed.

4.2.3 Application of AutoML to the Pilot Line

To apply an AutoML methodology to the selected pilot line, it is essential to collect operational data from the runtime system under varying asset parametrization. This data should include recorded measurements of variables and KPIs. Since not all the collected data have to be necessarily recorded using the same rate, it is necessary to transform the data to the same time base. This is commonly done by downsampling or averaging the data recorded with a higher rate to match the time base of the data recorded with a lower rate. In this case, averaging was used. Once the historical dataset has been prepared, an AutoML methodology can be applied. While typical AutoML methodologies automate the steps shown in Fig. 5, the proposed methodology also includes the process optimization procedure by embedding the selected model as objective function of an optimization algorithm for automatically finding the assets' configuration as depicted in Fig. 6.

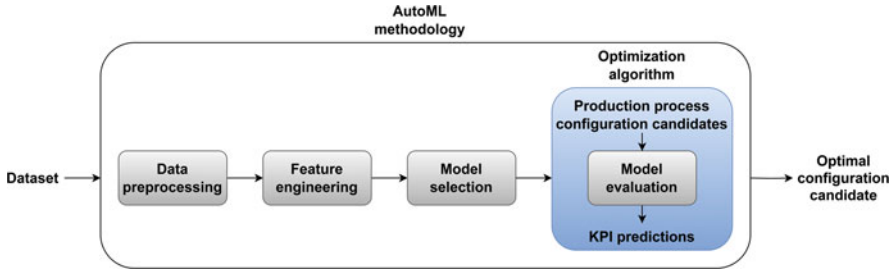


Fig. 6 Overall description of the proposed AutoML methodology

First, in the data preprocessing step, the dataset is inspected searching for missing values. If any are found, the corresponding sample is eliminated. Next, the features’ values are standardized and the dataset is divided into training and validation sets. In this case, hyperparameter optimization was not implemented for making the methodology applicable in scenarios with low computational resources. For this reason, a test set is not required. Following that, feature selection is carried out by computing the Pearson’s correlation coefficient (r) individually between each feature and the output variable on the training set, using the following equation:

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}$$

where n is the number of samples, x_i represents the value of the i -th sample of feature x , y_i represents the value of the i -th sample of the output variable, and \bar{x} and \bar{y} represent the mean of the respective variables.

Pearson’s correlation coefficient is a univariate feature selection method commonly used when the inputs and outputs of the dataset to be processed are numerical [48]. By using this method, it is possible to select the features with higher predictive capacity, resulting not only in a reduction of the dimensionality of data but also leads to more compact models with better generalization ability and reduced computational time [27, 28]. In the proposed approach, the features for which $|r| > 0.3$ are selected as relevant predictors, and the rest are discarded from both, the training and validation sets. Typically, a value below the 0.3 threshold is considered an indicator of low correlation [49]. During the application of the AutoML methodology to the data of the pilot line for improving the throughput, the number of features was reduced from 12 to 7. This intermediate result is important to guide the technicians on which parameters they should focus on while looking for a certain outcome.

The next step involves model selection. Among the different models to evaluate in the proposed approach are MLP, SVR, GPR, and CNN, which have been previously used for modeling industrial KPIs [40, 50–52]. Table 3 presents details of these models. Each one of these models is trained on the training set and then

Table 3 Details of the evaluated models

Model	Details
MLP	Architecture: Fully connected layer (128 units, ReLU activation) + Fully connected layer (64 units, ReLU activation) + Fully connected layer (1 unit, linear activation), Optimizer: RMSprop, Learning rate: 0.001, Epochs: 5000
SVR	Kernel: rbf, C: 1.0, Epsilon: 0.2, Tolerance: 0.001
GPR	Kernel: Dot Product + White Kernel, Alpha: 1e-10
CNN	Architecture: 1-D Convolution layer (64 filters, kernel size: 3, strides: 1, padding: same) + 1-D Max pooling layer (pool size: 2, strides: 1, padding: valid) + Flatten layer + Fully connected layer (64 units, ReLU activation) + Dropout layer (dropout rate: 0.1) + Fully connected layer (32 units, ReLU activation) + Fully connected layer (1 unit, linear activation), Optimizer: RMSprop, Learning rate: 0.001, Epochs: 5000

they are evaluated on the validation set. The metric used for comparison was the coefficient of determination (R^2). After this process is finished, the model that produced the best result is selected. The model selected during the application of the methodology to the pilot line was MLP with $R^2 = 0.963$ during validation. The R^2 value for the remaining candidate models was 0.958 for GPR, 0.955 for CNN, and 0.947 for SVR. One of the enablers of these results was the feature selection process, which allowed to retain the relevant predictors.

Finally, an optimization method is applied for determining the most favorable parametrization of the production process to minimize or maximize the desired KPI using the selected model as the objective function. In this case, the goal is to maximize throughput. The optimization is carried out using random search, which is a simple, low-complexity, and straightforward optimization method [53]. This method can be applied to optimizing diverse types of functions, even those that are not continuous or differentiable. It has been proven that random search is asymptotically complete, meaning that it converges to the global minimum/maximum with probability one after indefinitely run-time computation and, for this reason, it has been applied for solving many complex problems [54]. One aspect to consider before executing the optimization is that the feasible range of the parameters must be carefully decided to prevent the result of the optimization from being invalid. In the case analyzed, where the objective is to maximize the throughput of the pilot line, the obvious choice is to make the assets work at the maximum speed within the recommended ranges. To evaluate if the proposed methodology was capable of inferring this parametrization, during the preparation of the dataset the samples where all the assets were parametrized with the maximum speed were intentionally eliminated. As desired, the result of the methodology was a parametrization where all the assets were set to the maximum speed, yielding an expected throughput value of 163.37 panels per hour, which represents an expected improvement of 55.1% with respect to the higher throughput value present in the dataset. It is noticeable that the higher throughput value of the samples that were intentionally eliminated from the dataset is 158.52. The reason why the proposed methodology slightly overestimates this value is that the model is not perfectly accurate.

4.3 Fuzzy Logic-Based Reconfigurator

Once the parametrization of the assets has been determined by the AutoML methodology to meet a desired KPI performance, it is important to ensure that the system will continue to work as desired. Unfortunately, some situations may prevent the system from functioning as intended. For instance, a degradation in one of the assets may result in a slower operation, reducing the productivity of the entire line. For such cases, a fuzzy logic-based reconfigurator is developed. The intuition behind this component is that if the behavior of some assets varies from their expected performance, the reconfigurator can modify the parameters of the assets to make them work in the desired regime again, as long as the modification of the parameters is within a predefined safety range. Additionally, if the deviation from the expected performance is significant, the component should be able to detect it and inform the specialists that a problem needs to be addressed.

The proposed reconfigurator has two inputs and generates three outputs using the Mamdani inference method [55]. These variables are generic, so the reconfigurator can use them without any modification to try to keep each asset’s throughput level constant. The first input is the deviation from nominal production time (ΔT) and its safety range was defined as $\pm 50\%$ of the nominal production time. The second input is the change in the trend of the deviation from nominal production time (ΔT^2) and its safety range was defined as $\pm 20\%$ of the nominal production time. There is an instance of these two variables for each asset in the line and they are updated whenever a panel is processed. These values are normalized in the interval $[-1, 1]$ before being used by the reconfigurator. Figure 7 presents the membership functions defined for the two inputs.

On the other hand, the first output is the operation that the reconfigurator must apply to the current asset’s working speed (*Reco1*). If the operation is *Increase* or *Decrease*, the values in the interval $[-1, 1]$ are denormalized to a range comprising $\pm 50\%$ of the nominal asset speed. The second output represents the timing when the modifications should be applied (*Reco2*), and the third output represents the operation mode (*Reco3*), which specifies if the previous reconfigurator outputs

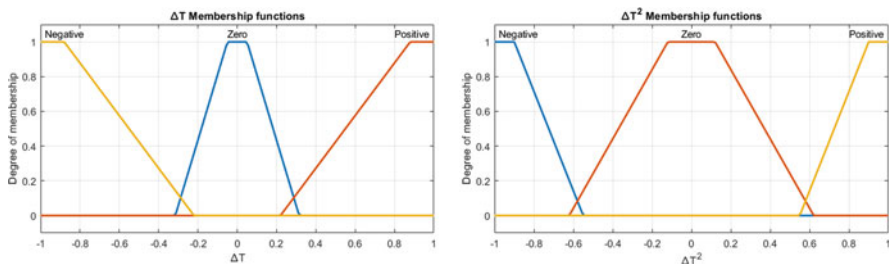


Fig. 7 Membership functions for inputs ΔT and ΔT^2

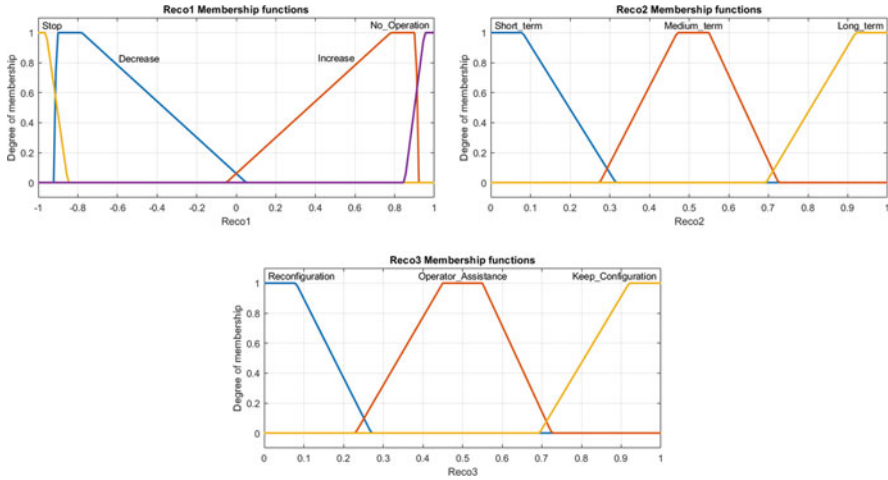


Fig. 8 Membership functions for outputs *Reco1*, *Reco2*, and *Reco3*

should be applied automatically, presented as recommendations for the operators, or ignored. Figure 8 shows the membership functions of the three outputs.

Once the membership functions of the input and output variables were defined, a rule base was created for each output variable. Each rule base is formed by nine If–Then rules that associate a combination of the input membership functions with an output membership function, as in the following example:

If ΔT is *Negative* **And** ΔT^2 is *Negative* **Then** *Reco1* is *Increase*

The defined rule bases allow to obtain the output surfaces illustrated in Fig. 9 for the fuzzy inference systems corresponding to each output variable.

To evaluate the reconfigurator, the nominal speed of each asset was set to 70% of its maximum speed and several disturbances were emulated. The first one was reducing the speed of all assets to 50% of their maximum speed, the second increasing the speed of all assets to their maximum speed, and finally, the speed of all assets was set to 30% of their maximum speed. As expected after the first disturbance the system recommended increasing the speed, after the second it recommended decreasing the speed, and after the third it recommended stopping the production. The results are shown in Table 4.

5 Conclusions

This work has addressed self-reconfigurable manufacturing systems from both theoretical and practical points of view, emphasizing how AI is applied to them. The emergence and evolution until the current state of these systems have been

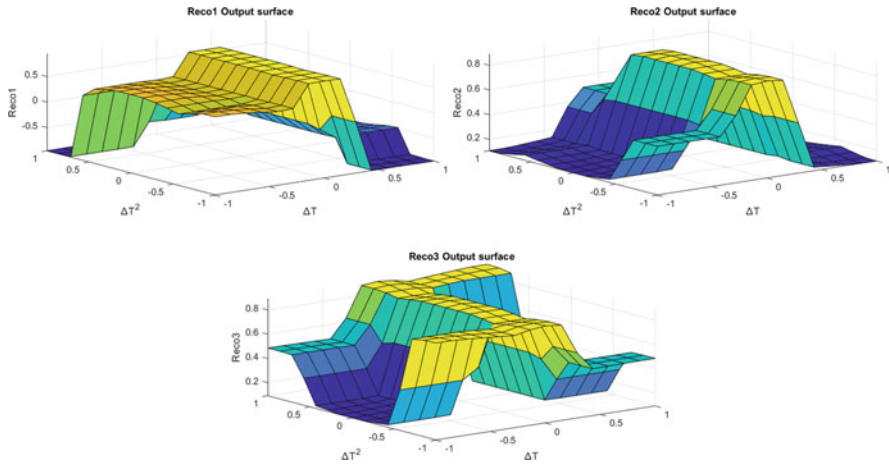


Fig. 9 Output surfaces of *Reco1*, *Reco2*, and *Reco3*

presented. Likewise, their potential benefits such as improved responsiveness, flexibility, and adaptability have been analyzed. Current approaches for implementing self-reconfiguration in manufacturing have also been discussed. Additionally, the application of self-reconfiguration and AI techniques to a pilot line was tested. First, the integration in the pilot line of an AI-based solution for visual inspection was evaluated. This component has a direct relation with the workflow of the pilot line, thus influencing the productivity. Two segmentation models were trained for the visual inspection task and the best one, with an accuracy of 0.995 and a F1 score of 0.992, was deployed in the pilot line, enabling the correct handling of products. Furthermore, an AutoML approach that includes generating the models and optimizing the production process was used for determining the optimal parametrization of the line. This way, a model with $R^2 = 0.963$ was obtained and the expected improvement in throughput with respect to the data seen during training is 55.1%, which matches the values reached in real production at maximum capacity. Then, a fuzzy logic-based reconfigurator was used for dealing with the degradation in performance. This component demonstrated a correct behavior and showed robustness when tested against three different perturbations. The findings of this study suggest that self-reconfiguration is a key area of research and development in the field of advanced manufacturing. Future research will explore additional applications of self-reconfiguration in different manufacturing contexts.

Table 4 Fuzzy logic reconfigurator results for disturbances

Asset	Nominal speed	Disturbance	<i>Reco1</i>	<i>Reco2</i>	<i>Reco3</i>
Stäubli RX90	7.7 m/s	1	Increase to 7.05 m/s	Medium term	Keep configuration
		2	Decrease to 8.94 m/s	Medium term	Keep configuration
		3	Stop	Short term	Operator assistance
Deckel Maho	7200 mm/min	1	Increase to 7027.8 mm/min	Medium term	Keep configuration
		2	Decrease to 8132.68 mm/min	Medium term	Keep configuration
		3	Stop	Short term	Operator assistance
Conveyor belts	5040 rpm	1	Increase to 4013.6 rpm	Medium term	Keep configuration
		2	Decrease to 5904 rpm	Medium term	Keep configuration
		3	Stop	Short term	Operator assistance
UR5e	126°/s	1	Increase to 115.28°/s	Medium term	Keep configuration
		2	Decrease to 146.1°/s	Medium term	Keep configuration
		3	Stop	Short term	Operator assistance
Kern EVO	11,200 mm/min	1	Increase to 10,250.2 mm/min	Medium term	Keep configuration
		2	Decrease to 12,992 mm/min	Medium term	Keep configuration
		3	Stop	Short term	Operator assistance

Acknowledgments This work was partially supported by the H2020 project “platform-enabled KITS of artificial intelligence FOR an easy uptake by SMEs (KITT4SME),” grant ID 952119. The work is also funded by 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,” Spain, grant ID PID2021-127763OB-I00, and supported by MICINN and NextGenerationEU/PRTR. This result is also part of the TED2021-131921A-I00 project, funded by MCIN/AEI/10.13039/501100011033 and by the European Union “NextGenerationEU/PRTR.”

References

1. Hees, A., Reinhart, G.: Approach for production planning in reconfigurable manufacturing systems. *Proc. CIRP*. **33**, 70–75 (2015). <https://doi.org/10.1016/j.procir.2015.06.014>
2. Yin, Y., Stecke, K.E., Li, D.: The evolution of production systems from Industry 2.0 through Industry 4.0. *Int. J. Prod. Res.* **56**, 848–861 (2018). <https://doi.org/10.1080/00207543.2017.1403664>
3. Mehrabi, M.G., Ulsoy, A.G., Koren, Y., Heytler, P.: Trends and perspectives in flexible and reconfigurable manufacturing systems. *J. Intell. Manuf.* **13**, 135–146 (2002). <https://doi.org/10.1023/A:1014536330551>
4. Cronin, C., Conway, A., Walsh, J.: Flexible manufacturing systems using IIoT in the automotive sector. *Proc. Manuf.* **38**, 1652–1659 (2019). <https://doi.org/10.1016/j.promfg.2020.01.119>
5. Parhi, S., Srivastava, S.C.: Responsiveness of decision-making approaches towards the performance of FMS. In: 2017 International Conference of Electronics, Communication and Aerospace Technology (ICECA), pp. 276–281 (2017)
6. Bennett, D., Forrester, P., Hassard, J.: Market-driven strategies and the design of flexible production systems: evidence from the electronics industry. *Int. J. Oper. Prod. Manag.* **12**, 25–37 (1992). <https://doi.org/10.1108/01443579210009032>
7. Singh, R.K., Khilwani, N., Tiwari, M.K.: Justification for the selection of a reconfigurable manufacturing system: a fuzzy analytical hierarchy based approach. *Int. J. Prod. Res.* **45**, 3165–3190 (2007). <https://doi.org/10.1080/00207540600844043>
8. Koren, Y.: The emergence of reconfigurable manufacturing systems (RMSs) BT - reconfigurable manufacturing systems: from design to implementation. In: Benyoucef, L. (ed.), pp. 1–9. Springer International Publishing, Cham (2020)
9. Koren, Y., Shpitalni, M.: Design of reconfigurable manufacturing systems. *J. Manuf. Syst.* **29**, 130–141 (2010). <https://doi.org/10.1016/j.jmsy.2011.01.001>
10. Epureanu, B.I., Li, X., Nassehi, A., Koren, Y.: An agile production network enabled by reconfigurable manufacturing systems. *CIRP Ann.* **70**, 403–406 (2021). <https://doi.org/10.1016/j.cirp.2021.04.085>
11. Koren, Y., Gu, X., Guo, W.: Reconfigurable manufacturing systems: principles, design, and future trends. *Front. Mech. Eng.* **13**, 121–136 (2018). <https://doi.org/10.1007/s11465-018-0483-0>
12. Gould, O., Colwill, J.: A framework for material flow assessment in manufacturing systems. *J. Ind. Prod. Eng.* **32**, 55–66 (2015). <https://doi.org/10.1080/21681015.2014.1000403>
13. Shin, K.-Y., Park, H.-C.: Smart manufacturing systems engineering for designing smart product-quality monitoring system in the Industry 4.0. In: 2019 19th International Conference on Control, Automation and Systems (ICCAS), pp. 1693–1698 (2019)
14. Arıcı, M., Kara, T.: Robust adaptive fault tolerant control for a process with actuator faults. *J. Process Control.* **92**, 169–184 (2020). <https://doi.org/10.1016/j.procont.2020.05.005>
15. Ghofrani, J., Deutschmann, B., Soorati, M.D., et al.: Cognitive production systems: a mapping study. In: 2020 IEEE 18th International Conference on Industrial Informatics (INDIN), pp. 15–22 (2020)

16. Scholz, S., Mueller, T., Plasch, M., et al.: A modular flexible scalable and reconfigurable system for manufacturing of microsystems based on additive manufacturing and e-printing. *Robot. Comput. Integr. Manuf.* **40**, 14–23 (2016). <https://doi.org/10.1016/j.rcim.2015.12.006>
17. Cedeno-Campos, V.M., Trodden, P.A., Dodd, T.J., Heley, J.: Highly flexible self-reconfigurable systems for rapid layout formation to offer manufacturing services. In: 2013 IEEE International Conference on Systems, Man, and Cybernetics, pp. 4819–4824 (2013)
18. Lee, S., Ryu, K.: Development of the architecture and reconfiguration methods for the smart, self-reconfigurable manufacturing system. *Appl. Sci.* **12** (2022). <https://doi.org/10.3390/app12105172>
19. Mourtzis, D.: Simulation in the design and operation of manufacturing systems: state of the art and new trends. *Int. J. Prod. Res.* **58**, 1927–1949 (2020). <https://doi.org/10.1080/00207543.2019.1636321>
20. dos Santos, C.H., Montevechi, J.A.B., de Queiroz, J.A., et al.: Decision support in productive processes through DES and ABS in the Digital Twin era: a systematic literature review. *Int. J. Prod. Res.* **60**, 2662–2681 (2022). <https://doi.org/10.1080/00207543.2021.1898691>
21. Guo, H., Zhu, Y., Zhang, Y., et al.: A digital twin-based layout optimization method for discrete manufacturing workshop. *Int. J. Adv. Manuf. Technol.* **112**, 1307–1318 (2021). <https://doi.org/10.1007/s00170-020-06568-0>
22. Abdi, M.R., Labib, A.W.: Feasibility study of the tactical design justification for reconfigurable manufacturing systems using the fuzzy analytical hierarchical process. *Int. J. Prod. Res.* **42**, 3055–3076 (2004). <https://doi.org/10.1080/00207540410001696041>
23. Lee, S., Kurniadi, K.A., Shin, M., Ryu, K.: Development of goal model mechanism for self-reconfigurable manufacturing systems in the mold industry. *Proc. Manuf.* **51**, 1275–1282 (2020). <https://doi.org/10.1016/j.promfg.2020.10.178>
24. Panetto, H., Jung, B., Ivanov, D., et al.: Challenges for the cyber-physical manufacturing enterprises of the future. *Annu. Rev. Control.* **47**, 200–213 (2019). <https://doi.org/10.1016/j.arcontrol.2019.02.002>
25. Schwung, D., Modali, M., Schwung, A.: Self -optimization in smart production systems using distributed reinforcement learning. In: 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC), pp. 4063–4068 (2019)
26. Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation BT - medical image computing and computer-assisted intervention – MICCAI 2015. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.), pp. 234–241. Springer International Publishing, Cham (2015)
27. Al-Tashi, Q., Abdulkadir, S.J., Rais, H.M., et al.: Approaches to multi-objective feature selection: a systematic literature review. *IEEE Access.* **8**, 125076–125096 (2020). <https://doi.org/10.1109/ACCESS.2020.3007291>
28. Solorio-Fernández, S., Carrasco-Ochoa, J.A., Martínez-Trinidad, J.F.: A review of unsupervised feature selection methods. *Artif. Intell. Rev.* **53**, 907–948 (2020). <https://doi.org/10.1007/s10462-019-09682-y>
29. Jebli, I., Belouadha, F.-Z., Kabbaj, M.I., Tilioua, A.: Prediction of solar energy guided by Pearson correlation using machine learning. *Energy.* **224**, 120109 (2021). <https://doi.org/10.1016/j.energy.2021.120109>
30. González, J., Ortega, J., Damas, M., et al.: A new multi-objective wrapper method for feature selection – accuracy and stability analysis for BCI. *Neurocomputing.* **333**, 407–418 (2019). <https://doi.org/10.1016/j.neucom.2019.01.017>
31. Alassaf, M., Qamar, A.M.: Improving sentiment analysis of Arabic Tweets by One-way ANOVA. *J. King Saud Univ. Comput. Inf. Sci.* **34**, 2849–2859 (2022). <https://doi.org/10.1016/j.jksuci.2020.10.023>
32. Urkullu, A., Pérez, A., Calvo, B.: Statistical model for reproducibility in ranking-based feature selection. *Knowl. Inf. Syst.* **63**, 379–410 (2021). <https://doi.org/10.1007/s10115-020-01519-3>
33. Bahassine, S., Madani, A., Al-Sarem, M., Kissi, M.: Feature selection using an improved Chi-square for Arabic text classification. *J. King Saud Univ. Comput. Inf. Sci.* **32**, 225–231 (2020). <https://doi.org/10.1016/j.jksuci.2018.05.010>

34. Lu, Z., Si, S., He, K., et al.: Prediction of Mg alloy corrosion based on machine learning models. *Adv. Mater. Sci. Eng.* **2022**, 9597155 (2022). <https://doi.org/10.1155/2022/9597155>
35. Anowar, F., Sadaoui, S., Selim, B.: Conceptual and empirical comparison of dimensionality reduction algorithms (PCA, KPCA, LDA, MDS, SVD, LLE, ISOMAP, LE, ICA, t-SNE). *Comput. Sci. Rev.* **40**, 100378 (2021). <https://doi.org/10.1016/j.cosrev.2021.100378>
36. Cruz, Y.J., Rivas, M., Quiza, R., et al.: A two-step machine learning approach for dynamic model selection: a case study on a micro milling process. *Comput. Ind.* **143**, 103764 (2022). <https://doi.org/10.1016/j.compind.2022.103764>
37. Castano, F., Cruz, Y.J., Villalonga, A., Haber, R.E.: Data-driven insights on time-to-failure of electromechanical manufacturing devices: a procedure and case study. *IEEE Trans. Ind. Inform.* 1–11 (2022). <https://doi.org/10.1109/TII.2022.3216629>
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.* **33**, 956–991 (2021). <https://doi.org/10.1007/s10696-020-09396-2>
39. Luo, J., Hong, T., Gao, Z., Fang, S.-C.: A robust support vector regression model for electric load forecasting. *Int. J. Forecast.* **39**, 1005–1020 (2023). <https://doi.org/10.1016/j.ijforecast.2022.04.001>
40. Pai, K.N., 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.* **241**, 116651 (2020). <https://doi.org/10.1016/j.seppur.2020.116651>
41. Cruz, Y.J., Rivas, M., Quiza, R., et al.: Computer vision system for welding inspection of liquefied petroleum gas pressure vessels based on combined digital image processing and deep learning techniques. *Sensors*. **20** (2020). <https://doi.org/10.3390/s20164505>
42. Pan, Y., Chen, S., Qiao, F., et al.: Estimation of real-driving emissions for buses fueled with liquefied natural gas based on gradient boosted regression trees. *Sci. Total Environ.* **660**, 741–750 (2019). <https://doi.org/10.1016/j.scitotenv.2019.01.054>
43. Chicco, D., Warrens, M.J., Jurman, G.: The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ. Comput. Sci.* **7**, e623 (2021). <https://doi.org/10.7717/peerj-cs.623>
44. Eltamaly, A.M.: A novel strategy for optimal PSO control parameters determination for PV energy systems. *Sustainability*. **13** (2021). <https://doi.org/10.3390/su13021008>
45. Karagul, K., Sahin, Y., Aydemir, E., Oral, A.: A simulated annealing algorithm based solution method for a green vehicle routing problem with fuel consumption BT - lean and green supply chain management: optimization models and algorithms. In: Weber, G.-W., Huber, S. (eds.) *Paksoy T*, pp. 161–187. Springer International Publishing, Cham (2019)
46. Cruz, Y.J., Rivas, M., Quiza, R., et al.: Ensemble of convolutional neural networks based on an evolutionary algorithm applied to an industrial welding process. *Comput. Ind.* **133**, 103530 (2021). <https://doi.org/10.1016/j.compind.2021.103530>
47. Yildiz, A.R.: A novel hybrid whale–Nelder–Mead algorithm for optimization of design and manufacturing problems. *Int. J. Adv. Manuf. Technol.* **105**, 5091–5104 (2019). <https://doi.org/10.1007/s00170-019-04532-1>
48. Gao, X., Li, X., Zhao, B., et al.: Short-term electricity load forecasting model based on EMD-GRU with feature selection. *Energies*. **12** (2019). <https://doi.org/10.3390/en12061140>
49. Mu, C., Xing, Q., Zhai, Y.: Psychometric properties of the Chinese version of the Hypoglycemia Fear SurveyII for patients with type 2 diabetes mellitus in a Chinese metropolis. *PLoS One*. **15**, e0229562 (2020). <https://doi.org/10.1371/journal.pone.0229562>
50. Schaefer, J.L., Nara, E.O.B., Siluk, J.C.M., et al.: Competitiveness metrics for small and medium-sized enterprises through multi-criteria decision making methods and neural networks. *Int. J. Proc. Manag. Benchmark.* **12**, 184–207 (2022). <https://doi.org/10.1504/IJPMB.2022.121599>
51. Manimuthu, A., Venkatesh, V.G., Shi, Y., et al.: Design and development of automobile assembly model using federated artificial intelligence with smart contract. *Int. J. Prod. Res.* **60**, 111–135 (2022). <https://doi.org/10.1080/00207543.2021.1988750>

52. Zagumennov, F., Bystrov, A., Radaykin, A.: In-firm planning and business processes management using deep neural network. *GATR J. Bus. Econ. Rev.* **6**, 203–211 (2021). [https://doi.org/10.35609/jber.2021.6.3\(4\)](https://doi.org/10.35609/jber.2021.6.3(4))
53. Ozbey, N., Yeroglu, C., Alagoz, B.B., et al.: 2DOF multi-objective optimal tuning of disturbance reject fractional order PIDA controllers according to improved consensus oriented random search method. *J. Adv. Res.* **25**, 159–170 (2020). <https://doi.org/10.1016/j.jare.2020.03.008>
54. Do, B., Ohsaki, M.: A random search for discrete robust design optimization of linear-elastic steel frames under interval parametric uncertainty. *Comput. Struct.* **249**, 106506 (2021). <https://doi.org/10.1016/j.compstruc.2021.106506>
55. Mamdani, E.H.: Application of fuzzy algorithms for control of simple dynamic plant. *Proc. Inst. Electr. Eng.* **121**, 1585–1588 (1974). <https://doi.org/10.1049/piee.1974.0328>

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

