Engineering Applications of Artificial Intelligence 133 (2024) 108340

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

A conceptual framework for machine learning algorithm selection for predictive maintenance

Simone Arena [a,*], Eleonora Florian [b], Fabio Sgarbossa [c], Endre Sølvsberg [c], Ilenia Zennaro [b]

[a] Department of Mechanical, Chemical, and Material Engineering, University of Cagliari, Via Marengo 2, 09123, Cagliari, Italy
[b] Department of Management and Engineering, University of Padua, Stradella San Nicola 3, 36100, Vicenza, Italy
[c] Department of Mechanical and Industrial Engineering, NTNU, Richard Birkelands Vei 2B, 7031, Trondheim, Norway

ARTICLE INFO

Keywords:
Machine learning
Predictive maintenance
Algorithm selection
Conceptual framework
Decision making

ABSTRACT

The Industry 4.0 paradigm enables advanced data-driven decision-making processes leading many manufacturers to a digital transformation. Within this context, Predictive Maintenance (PdM) - i.e. a maintenance strategy that predicts failures in advance - based on Machine Learning (ML) - i.e. a set of algorithms to analyze data for pattern recognition - emerged as one of the most prominent data-driven analytical approaches for maximizing availability and efficiency of industrial systems. Indeed, there exists a considerable body of literature dealing with ML-based PdM where a wide set of ML algorithms has been applied to a broad range of industrial settings. Whilst this results in extensive knowledge on the topic, the need to choose the right algorithm for a specific task arises as a challenging issue since it is considered an essential stage in the development and implementation of an ML-oriented approach. To respond to such a necessity, this work proposes a conceptual framework to guide practitioners as well as non-expert users in ML algorithm selection for PdM issues. The aim is to provide a set of guidelines and recommendations for the identification of which ML techniques are likely to achieve valuable performance for specific tasks or datasets. First, the most commonly applied ML algorithms in PdM are analyzed together with their core characteristics, advantages, and disadvantages. Then, several decision variables depending on dataset and ML characteristics, learning objectives, accuracy and interpretability are considered. Finally, illustrative case studies are presented to demonstrate how the proposed framework can be adopted in real industrial applications.

1. Introduction

In recent years, the technological revolution known as Industry 4.0 is driving companies toward a sustained change in the traditional perspective. This transition involves the gradual digitalization and automation of manufacturing practices through the adoption of a wide spectrum of disruptive technologies to manage companies' processes at all levels of production, management, service, and maintenance. In this scenario, the advance of new technologies and tools such as Big Data, Internet of Things (IoT), and cyber-physical system (CPS) allow the development of innovative approaches to industrial automation based on the digitalization process and cognitive production (Alabi et al., 2018). Thus, the result of this unavoidable tendency will be the conversion of enterprises into smart factories capable of designing integrative systems where every component communicates and cooperates with each other and with humans in real-time. This contributes to generate a large amount of data from several sources, such as characteristics of products, machines, production lines, materials, failures, or human resources (Dogan and Birant, 2021). From this perspective, data is increasingly considered as one of the most significant assets since it generates added value for companies providing a considerable potential in terms of useful information and knowledge (Carvalho et al., 2019). But to take full advantage of the data, it is necessary to know how to get the most value from it. Indeed, it is vital to adopt the right tools and technologies to aim at turning these information into beneficial outcomes. Machine Learning (ML), a branch of AI, has the potential to become one of the main driver in the fulfilment of these needs by providing a valuable decision support in a wide range of manufacturing and production applications (Rai et al., 2021).

Within the industrial field, a suitable application of these concepts concerns asset maintenance since the availability of huge amount of data may be helpful for planning maintenance activities aiming at avoiding

* Corresponding author.
E-mail address: simone.arena@unica.it (S. Arena).

https://doi.org/10.1016/j.engappai.2024.108340
Received 31 October 2023; Received in revised form 23 January 2024; Accepted 25 March 2024
Available online 10 April 2024

equipment failures (Mobley, 2002). Analysis of current and historical data that describe system states, events, and operations is a key component of failure prediction approaches. Indeed, the adoption of sensors mounted on machines enable the monitoring of proper health condition parameters during operation to provide significative knowledge of degradation processes and potential failures (Florian et al., 2019). This advanced ~~technique~~ strategy is commonly called Predictive Maintenance (PdM) which permits failure detection at an early stage before failures occur through the continuous monitoring of equipment performance or conditions. The adoption of PdM allows for several benefits such as a cost reduction, an increase in machinery lifetime, and the improvement of both safety and product quality among others (Ruiz-Sarmiento et al., 2020; Vogl et al., 2016). One of the most suitable methodologies, adopted in PdM, is based on Machine Learning (ML) algorithms which adaptively learn the diagnosis knowledge of monitored equipment aimed at preventing failures (Zenisek et al., 2019). ML is commonly categorized into four groups: (i) supervised learning in which the algorithm learns from the input data consisting of labeled training datasets; (ii) unsupervised learning in which the algorithm learns by using unlabelled datasets to detect patterns or dependencies within those data; (iii) semi-supervised learning in which the algorithm is trained by both labeled and unlabelled data; (iv) reinforcement learning that refers to a goal-oriented algorithm performed in an environment through a trial and error process (Gupta et al., 2016). Literature on ML covers a large variety of relevant studies addressing different approaches and techniques adopted for each specific application (Çinar et al., 2020; Dalzochio et al., 2020; Lei et al., 2020; Bertolini et al., 2021; Leukel et al., 2021; Nacchia et al., 2021; Rai et al., 2021; Jahani et al., 2023; Rolf et al., 2023).

Generally, the algorithm selection is a crucial step within the decision-making process since there is no unique ML algorithm that can efficiently handle the analysis of every dataset or all use cases. Often, the choice is established merely by the comparison between different algorithms whose final decision falls on the one that ensures the "best solution" only in terms of predictive and computational performance. As a consequence, this may frequently lead to results that are poor or too generic for the problem at hand. Therefore, a different approach is required by means of the analysis of specific features suitable for guiding users toward their choice. The criticality attending this step lies in defining a valid criterion capable of consistently determining the ML algorithm best suited for the requested task depending on such features.

Concerning maintenance issues, the adoption of ML-based models relies on their great potential in fault detection, so they are considered as the key elements in data-driven strategy to support a valuable real-time decision-making process. Consistent with this vision, their proper implementation is strongly influenced by several aspects. The first aspect is related to the ML input, i.e., the size and the type of available data. An appropriate amount of data is needed to ensure a good capability for the identification of the health status of the monitored machine aimed at achieving the maintenance goal of predicting potential failures. The type of data should meet the requirements in terms of maturity and quality, i.e., dataset should be complete, objective, consistent and relevant to ensure value-added and efficient maintenance activities (Florian et al., 2019). A second aspect concerns the ML algorithm performance. Indeed, the basic idea is to guide the choice towards the one that maximizes accuracy, precision and the training/prediction speed aims at ensuring shorter data processing times or better prediction outcomes. Another aspect emerges by considering the constraints existing in real industrial applications, such as the investment costs, the connection of physical assets, the storage of the data and specific user's priority.

Usually, user experience or prior research on similar content can be used to support the ability to select the best ML algorithm, but it appears evident that it is crucial to define the applicability domain in terms of technical, organizational, and economic assessment aiming at assisting users during the decision-making process.

Therefore, this work proposes a novel framework to guide companies, practitioners, and non-expert users in ML algorithm selection for predictive maintenance applications. This approach is based on the definition of the main learning characteristics, as well as on the maintenance objective aimed at providing proper support to the decision-making process. It provides a qualitative assessment of the achievable accuracy based on the maturity and quality of the data and the statistical characteristics of the dataset. It is worth underlining that the proposed framework does not claim to cover all the extant literature on the ML algorithms suggesting which is generally the best one should be implemented since the problem under study is complex and non-trivial. The main aim is to provide guidance to researchers and practitioners when designing strategies to successfully approach ML-based predictive maintenance issues. So, the basic idea is to develop an easy and responsive tool aiming at performing agile and informed decision-making in establishing a proper ML algorithm suitable for the specific case at hand if the above-mentioned aspects and characteristics for doing so are taken into consideration.

This paper is organized as follows: Section 2 illustrates the theoretical background, Section 3 reports the main ML algorithms usually adopted for PdM issues, Section 4 introduces the ML algorithms selection framework for PdM, and Section 5 presents the illustrative case studies of the proposed framework. Section 6 briefly describes the overall requirements needed for the proper implementation of a ML-oriented approach. Finally, in Section 6, the conclusions and future research are depicted.

2. Theoretical background

2.1. Predictive maintenance

Both Predictive Maintenance (PdM) and Condition-Based Maintenance (CBM) are preventive strategies that aim at reducing unexpected equipment failures and minimizing downtime (Prajapati et al., 2012; Quatrini et al., 2020; Zonta et al., 2020). CBM is a preventive maintenance approach that involves monitoring the actual condition of equipment using sensors and other data collection methods to assess the current state of components. PdM, instead, is a proactive maintenance strategy that uses data analytics, sensors, and predictive modelling to predict when equipment is likely to fail. The goal of PdM is to intercept in advance the symptoms of anomalous behaviours of a physical system through just-in-time maintenance actions aiming at preserving the availability, quality, and safety of the system, decreasing the risks of failure, and reducing the costs of unnecessary time-based maintenance activities.

This approach involves the development of predictive tools based on historical data, statistical inference methods, and engineering approaches (Nacchia et al., 2021). Indeed, implementing PdM means constructing a learning system that automatically improves through experience and consequently identifying the fundamental elements governing this system, which also involves humans and organizations (Jordan and Mitchell, 2015).

PdM success depends on the quality and robustness of the condition monitoring system (CMS) and it is developed according to the following key phases: data processing, and decision making (Jardine et al., 2006; Martin, 1994).

Data Processing comprehends Data Acquisition, Data Manipulation, State Detection, Health Assessment, Prognostic Assessment and Advisory Generation (ISO 13374-2, 2021). Data acquisition can be provided by different sources, and the data can be of different types, including monitoring data, maintenance event data, and process data. It is the initial step in the machinery prognostics process which provides basic condition monitoring knowledge for subsequent steps. Data Manipulation calculates the descriptors/indexes from collected data, while State Detection categorizes data and defines if the component or the system is in a state of "normal" or "abnormal", The Health Assessment phase, instead, defines the current health state of the component/system and

the associated diagnosis; the Prognostic Assessment process, instead, defines the future state of the component/system and the remaining useful life (RUL). Finally, the Advisory Generation process integrates different information to generate advisories for the decision-making process.

PdM objectives may be divided into two main categories – diagnostics and prognostics – with a different technique implemented for each (Jardine et al., 2006). The first category is a posterior event analysis, so the aims is to recognize patterns in the data, reveal specific fault conditions, and the fault is identified in real-time (Kiangala and Wang, 2018). The second category is a prior event analysis and it aims at estimating the Remaining Useful Life (RUL) of the monitored component, that is, at predicting how much time is left before a failure occurs, given the current machine condition and the past operation profile (Kim and Sohn, 2021). Finally, the decision-making process enables the identification of a potential set of maintenance tasks aiming at effectively achieving the defined objectives.

Several methods for the realization of the PdM models are reported in the literature: model-based, data-driven, knowledge-based, and combination models (Ying et al., 2010). Currently, Machine Learning (ML) is a powerful tool for developing intelligent learning systems in many applications of PdM (Carvalho et al., 2019). In fact, ML techniques have the ability to handle high dimensional and multivariate data, as well as to extract hidden relations within the data in complex and dynamic environments (Wuest et al., 2016). In this work, we focus on a data-driven approach in which the analysis is conducted starting from the data and the relations within them; these relations are identified using statistical techniques or artificial intelligence (AI). The principal objectives of data-driven models are to detect the faults in advance using historical and on-time data, to estimate the RUL and condition indicators of the system, and consequently, to classify the fault with a diagnosis. These data-driven approaches are able to find highly complex and non-linear patterns in data of different types and from various sources and convert raw data into learning models, which are then applied to fault prediction, fault detection, fault classification, or forecasting. The quality and maturity of the data condition affect the feasibility of this approach (Pillai and Vadakkepat, 2021).

2.2. Machine learning

As a subfield of AI, ML enables computer programs to build a mathematical model based on historical data, known as "training data", in order to make a prediction, a diagnosis, or a plan; recognize behaviour patterns; or make decisions without *a priori* knowledge and without being explicitly programmed for the task (Burkov, 2019; Zhang, 2020). Mitchell (1999) defines the learning process as follows: *"A computer program is said to learn from experience E concerning some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E."*

The learning process involves the input variables X (predictors, independent variables, features, or just variables) and one or more output variables Y (target, response, or dependent variable). The purpose of ML is to define the relation f between X and Y, whose general expression is as follows:

$$Y = f(X) + \in, \tag{1}$$

where f is an unknown function representing the systematic information that X provides about Y, and $\in$ is a random error. ML refers to a series of algorithmic approaches to estimate f for the purpose of making a prediction or an inference. In the first case, the goal is the maximization of accuracy, that is, identifying f that minimizes the prediction error. In the second one, the goal is interpretability, that is, identifying which predictors are associated with the response, or which relation the predictors and the response are in, or how the relation between Y and the predictors can be summarized. The learning process involves using n observations, called training data, to teach an algorithm or method, how to estimate f and *fit* (or *train*) a model to the samples. Thus, the *training dataset* is defined as follows:

$$\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}, \text{where } x_i = \left(x_{i1}, x_{i2}, \ldots, x_{ip}\right)^T, \tag{2}$$

x_{ij} denotes the input (also called input or features), y represents the output (response variable), with $i = 1,2,\ldots,n$, n number of input and $j = 1,2,\ldots,p$, p number of output, and the learning process aims to estimate function $\widehat{f}$ (it represents the estimate for f), such that $Y \approx \widehat{f}(X)$ for each observation (X, Y). The training process aims to minimize the prediction error. No method is better than all the others for all possible datasets. Thus, an important step in developing an ML model is deciding which learning method provides the best results for each dataset. The measurement of the quality of the fit can be performed with different metrics and techniques, based on the learning activity. However, it is necessary to compare these metrics on a *test dataset* that has not been previously analyzed.

The training methods are classified as *parametrics* and *non-parametrics* (Hastie et al., 2017). Parametrics are model-based or rather, based on assumptions of the functional form of f. Non-parametrics make no explicit assumptions about the functional form of f, making these methods more flexible. However, each method has advantages and disadvantages. Parametric methods are more restrictive but need fewer observations to converge. Non-parametric methods are more easily adaptable to a wide range of functions but require a large number of observations to obtain an accurate estimate of f. Furthermore, these more complex methods can suffer from *overfitting*, a phenomenon where the estimated model fits the training data too much and fails to generalize the problem. In contrast, parametric methods can suffer from *underfitting*; in fact, if the chosen model vastly differs from the one hidden in the data, the estimate produces high bias and is unreliable. Parametric methods are computationally simpler; they are indicated in case of linear or smooth models and if the relation between input and output is well defined. Generally, restrictive methods are more interpretable and therefore particularly suitable when inference is the main objective. On the contrary, very flexible methods can lead to complicated estimates of f, making interpretation impossible. A model's interpretability and accuracy are equally desirable characteristics when estimating f but in contrast to each other. In fact, the restrictive (parametric) methods are characterized by high interpretability and low accuracy, while the more flexible (non-parametric) methods are characterized by low interpretability and high accuracy. While the advantage of applying simple methods when the aim is inference has been demonstrated, there is no uniqueness when the aim is accuracy (Hastie et al., 2017). Although the more flexible methods are designed to favour the accuracy of the estimate, it is uncertain that with them, analysts will always obtain more accurate estimates. In fact, if the relation is simple, using linear or spline models, analysts can obtain more accurate predictions, avoiding overfitting.

The literature reveals two main learning approaches: active learning and passive learning. The former refers to the learning processes where the ML algorithm learns from pre-selected data and is, therefore, able to perform better with less training (Settles, 2009). The latter refers to the learning approaches that use random samples from the dataset to build models for predicting the output (Mishra and Gupta, 2017). Based on the feedback to learning, three passive learning types are distinguished: *supervised learning*, *unsupervised learning*, and *reinforcement learning*. In supervised learning, the algorithm "observes some example input-output pairs and learns a function that maps from input to output" (Russell and Norvig, 2002).

This approach is described in Eq. (2), where for each observation $i = 1, 2, \ldots, n$, there is an input x_i, but a target variable y_i exists. Thus, the goal is to define a training data-mapping function that is able to make predictions (Silva and Zhao, 2016). The response variables are called labels, which constitute one or more tags that contain desirable

information on the data and favour their recognition. Based on the response type, two types of learning activity can be defined: regression and classification. The former aims to predict a continuous variable, the latter a categorical or binary one. The previous discussion about parametric and non-parametric models is related to supervised learning approaches. In *unsupervised learning*, the algorithm "learns patterns in the input even though no explicit feedback is supplied" (Russell and Norvig, 2002). It describes the situation where for each observation $i = 1, 2, \ldots, n$, there is an input x_i, but no response y_i is available. The main task involves discovering intrinsic patterns and extracting unknown insights from the data. In this case, no labels are available for the data analysis (Mitchell, 1999). The main unsupervised approach is the clustering analysis that aims to distinguish groups inside the data. Semi-supervised learning aims to combine these two tasks, attempting to improve the performance in one of them by utilizing information generally associated with the other (van Engelen and Hoos, 2020). For instance, in a classification problem, additional data points labeled with clustering methods might be used to aid in the classification process. In contrast, for unsupervised methods, knowing that certain data points belong to the same class might improve the clustering results. Finally, in *reinforcement learning*, the algorithm learns from a series of reinforcements – rewards or punishments. Its aim is to take suitable action in a given situation in order to maximize a reward (Sutton and Barto, 2018). The algorithm discovers outputs through a trial-and-error process. Typically, there is a sequence of states and actions where the learning algorithm interacts with its environment.

As introduced before, a priority step during ML model development is the method selection for the analyzed dataset, since no method is better than all the others for all possible datasets. The model is selected by evaluating the prediction accuracy achieved on a test dataset that has not been previously analyzed. For comparing predictive ability on a test dataset, the most popular metrics for regression are *mean squared error (MSE)* or *mean absolute error (MAE)*; for classification problems, the metrics are *accuracy, precision, recall, score, F1, ROC,* and *AUC* (Bishop, 2006; Strutz, 2011; Goodfellow et al., 2016; Hastie et al., 2017). The metric selection is a task assigned to data science that performs it based on the model goal. In case a test dataset is unavailable, the mean prediction error can be estimated via *cross-validation*, which is a method of estimating the test error on the training data. In each iteration, the cross-validation method involves dividing the dataset into a training set and a validation set. The former is used for fitting the model, the latter for estimating the error; the final error is calculated as the average of the errors observed in each iteration.

A key factor influencing training success is the number of available observations; thus, it must be considered in the training method selection. In fact, while model-based methods allow obtaining reliable predictions even with a reduced number of observations, non-parametric methods require a huge amount of data to converge, having to estimate many more coefficients (Hastie et al., 2017). There are statistical heuristic methods available that allow calculating suitable sample sizes. For instance, samples must be multiples of the number of classes to predict (e.g., tens, hundreds, or thousands). In another case, the samples must be a defined percentage greater than the number of inputs; in others, the number of samples should be related to the number of model parameters (Jain and Chandrasekaran, 1982).

Finally, algorithms can be characterized by their ability to manage outliers. Generally, data that do not belong to the sample are identified as outliers. These generate noise that reduces the accuracy of the prediction. Usually, these samples are removed (Grbić et al., 2013). However, the datasets related to maintenance problems may have few samples related to faults (Susto et al., 2015). Defining these samples as outliers would be a mistake.

In these situations, it is important to be able to use an algorithm that ensures good performance even with imbalanced datasets.

The analysis of the most relevant scientific contributions shows that most of the articles focus on the development of the ML model, based on specific cases and applications. Most of them propose ML approaches with a predefined maintenance objective, not considering what conditions lead to selecting one rather than another. While there are frameworks for selecting the most suitable algorithm to train the ML model, many of them focus on accuracy (Kotsiantis et al., 2007; Malhotra, 2015; Lee et al., 2016; Singh et al., 2016; Accorsi et al., 2017; Liu et al., 2018; Xu et al., 2019; Paturi and Cheruku, 2020; Xu and Saleh, 2021), prediction speed (Kotsiantis et al., 2007; Singh et al., 2016; Liu et al., 2018; Xu et al., 2019), or computational requirements (Kotsiantis et al., 2007; Javed et al., 2017; Saxena et al., 2017; Xu et al., 2019; Xu and Saleh, 2021). Other selection approaches require the computation of a specific value that is able to suggest the best statistical model to be used in an analysis, such as the Akaike information criterion (AIC) and the Bayesian information criterion (BIC) (Hastie et al., 2017). These criteria assume that the user knows the algorithm characteristics and can elaborate complex statistical analysis. Sala et al. (2018) address the problem by including decision variables, such as algorithm scalability or robustness to outliers. However, no tool relates algorithmic approaches to the maintenance process. This study aims to provide a tool for those who are approaching ML to guide them in effective algorithm selection, combining maintenance and analytical knowledge.

3. Algorithms

Covering this gap requires identifying the decision variables that affect the algorithm selection. In particular, it is necessary to understand the algorithms' behaviour as a function of the dataset's statistical characteristics. For this purpose, this section provides a description of the main algorithms used for PdM to allow a comprehensive understanding of problems. The list is not exhaustive and is limited to ML and does not deal with time-series forecasting, a topic that will be explored in future research, together with deep learning algorithms, semi-supervised algorithms and reinforcement learning algorithms. This allows the research method used to be analyzed in order to extend it to other algorithms or decision variables.

The next section shows how this information has been categorized into a conceptual framework that fits the research aim. The following list reports the algorithms considered for supervised learning.

- Linear Least Squares Regression (LLSR). This algorithm is suitable for predicting the values of a continuous response variable Y based on predictors X, assuming a linear relation between X and Y, as in the following equation: $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon$. The algorithm's task is to estimate the regression coefficients by minimizing the sum of squares of the residuals (RSS) [30]. This algorithm has difficulty in finding a model that fits the data of inherently non-linear processes. Furthermore, while the LLSR method often provides optimal estimates of unknown parameters, it is very sensitive to the presence of unusual data points in the data used to fit a model. One or two outliers can sometimes seriously skew the results of a least squares analysis (Pham, 2006).
- Lasso Linear Regression (Lasso). This algorithm is suitable for regression. Lasso regression is very close to LLSR; however, the coefficients are estimated by minimizing a slightly different amount. In particular, the estimates of the coefficients of Lasso $\widehat{\beta}_\lambda^L$ are the coefficients that minimize $\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2 + \lambda\sum_{j=1}^{p}|\beta_j| = RSS + \lambda\sum_{j=1}^{p}|\beta_j|$, where λ is a tuning parameter. Introducing the term $|\beta_j|$, called the *shrinkage penalty*, has the effect of reducing the estimates of β_j towards zero and improving the accuracy of the model. The penalty has the effect of forcing some of the coefficient estimates to be exactly equal to zero when the λ tuning parameter is large enough, thus selecting the variables that facilitate the interpretation of these models (Agresti, 2003).

- Logistic Regression (Logit). This algorithm is suitable for binary classifications. Although they can be extended to more classes, in practice, these models are not used in multiple classification problems, which prefer the linear discriminant analysis. It models the probability that Y belongs to a specific category, where 0/1 encoding can be used for the response variable. For this reason, this logistic function is used: $p(X) = e^{\beta_0+\beta_1 X}/(1 + e^{\beta_0+\beta_1 X})$, and the coefficients can be estimated with the maximum likelihood method [30]. As with the LLSR method, it is best suited for linear processes, thus for data separated by a single linear boundary (Dreiseitl and Ohno-Machado, 2002).
- Linear Discriminant Analysis (LDA). This algorithm is suitable for multiple classifications. This classifier is derived from the assumption that the observations of each class follow a normal distribution. It tries to approximate the Bayes classifier (Mitchell, 1997) by introducing the estimates of mean, variance, and *a priori* probability of each class into the model that describes the discriminant functions of the classes, which are linear (Balakrishnama and Ganapathiraju, 1998). Compared with the Logit method, it is more stable in cases of well-separated classes, in cases where the distribution of predictors is approximately normal, and, as mentioned, in the case of multiple classifications (Hastie et al., 2017).
- Quadratic Discriminant Analysis (QDA). This algorithm is suitable for multiple classifications. Similar to LDA, the QDA classifier results from the assumption that the observations of each class are drawn from a Gaussian distribution and include the parameter estimates in Bayes' theorem to make the prediction. However, different from the LDA, the QDA assumes that each class has its own covariance matrix. Compared with the LDA, the QDA is a more flexible classifier, better in cases where the training set is very large or the assumption of a common covariance matrix for the classes is unsustainable (Duda et al., 1973; Hastie et al., 2017).
- Autoregressive Moving Average (ARIMA). It can be applied to many real-time series. These models are based on three parts: an autoregressive (AR) part, a contribution from a moving average (MA), and a part involving the first derivative of the time series. The AR part of the model has its origin in the theory that individual values of time series can be described by linear models based on preceding observations. The consideration leading to MA models is that time-series values can be expressed as dependent on the preceding estimation errors. Past estimation or forecasting errors are considered when estimating the next time-series value. The integrating part (I) is used when trend filtering is required. The algorithm fits the data in order to estimate parameters p,q,d, where parameter p denotes the order of the AR part, parameter q represents the order of the MA part, and d signifies the number of differentiation steps (Hamilton, 2020).
- Decision Tree (DT). This algorithm is suitable for both regression and classification. It is non-parametric. It has a flowchart structure, where each node represents a test and each leaf represents the answer. In the case of regression, in each step, the predictor space is divided into distinct and non-overlapping regions. For each observation of each region, a prediction equal to the mean of the response values is estimated. The goal is to minimize the value of the global *RSS*; therefore, among all candidates, the predictor and the division point are selected in such a way that leads to the maximum reduction of the *RSS* value. The tree-pruning techniques improve the DT's ability to generalize and its interpretability. In the classification, the response is qualitative, and for tree growth, the *Gini index* or *Entropy* is used. DT is suitable when the data have many features that interact in complicated and non-linear ways. DT methods are simple and useful for interpretation. However, they are generally not competitive with the best supervised learning approaches in terms of predictive accuracy since they are not robust as data change. While it can handle a wide variety of input data, this algorithm is unsuitable for large datasets (Breiman et al., 1984b).
- k-Nearest Neighbours (k-NN). This algorithm is suitable for both classification and regression. It is a non-parametric method. Regarding classification, given a positive integer K and a test observation x_0, the k-NN classifier first identifies the K points in the training data closest to x_0 (represented by N_0). Then, it estimates the conditional probability for class j as the fraction of the points in N_0 that have values of the response variable equal to j: $\Pr(Y = j|X = x_0) = 1/K \sum_{i \in N_0} I(y_i = j)$. Finally, the k-NN method applies Bayes' theory and classifies the test observation x_0 into the class with the highest probability. In case regression x_0 is a prediction point, and $f(x_0)$ is estimated using the mean of all training responses in N_0, $\widehat{f}(x_0) = 1/K \sum_{x_i \in N_0} y_i$, (Hastie et al., 2017). Consequently, the algorithm assumes that objects close to each other are similar. The algorithm can be trained using different distance metrics (e.g., Euclidean, Chebyshev, etc.). Its performance is strongly influenced by the size of the data and the presence of outliers and noise (Kotsiantis et al., 2007). The scale of features for k-NN regression influences the quality of the predictions. This method is particularly indicated in case of a nonlinear decisional boundary or when X and Y have a non-linear relation. A useful tool to consider in combination with ML methods like k-NN and XG-Boost (described below) is the statistical method for multivariate outlier detection called the Mahalanobis Distance (MD) (Cabana et al., 2021). MD is mainly adopted to identify and remove multivariate outliers and improve prediction (Dashdondov and Kim, 2023), but it is also applied as a method in complex manufacturing systems (Palacín et al., 2021; Sølvsberg et al., 2023). Although there are novel ML approaches where the MD is used in non-Euclidian distance learning (Tao et al., 2020; Yin et al., 2023), it's primary function in ML approaches is as a support tool for other ML methods. Methods like k-NN and others that can apply non-Euclidian distance metrics, can be trained with the MD distance to improve classification performance (Berrendero et al., 2020).
- Generalized Additive Models (GAM). This algorithm is suitable for both regression and classification. It provides a general framework for extending a standard linear model by allowing non-linear functions for each of the variables while maintaining additivity. Non-linear fits can potentially make more accurate predictions for the Y response, while retaining interpretability, so they are useful if analysts are interested in inference. However, the model is limited to being additive, with the risk of losing interactions between variables (Hastie and Tibshirani, 2017).
- Support Vector Machine (SVM). This algorithm is suitable for both regression and classification. In the classification, the algorithm looks for a linearly separable hyperplane that separates the values of one class from those of the other. If there is more than one, it looks for the one that has the highest margin with the support vectors to improve the accuracy of the model. If such a hyperplane does not exist, SVM uses non-linear mapping to transform the training data into a higher dimension (if there are two dimensions, it will evaluate the data in three dimensions). In this way, the data of two classes can always be separated by a hyperplane, which will be chosen for splitting the data. SVM uses a kernel to quantify the similarity between observations (Hofmann et al., 2008). In regression, SVM predicts a continuous response. In this case, it looks for a model that deviates from the measured data by a value not exceeding a small amount, having the values of the parameters as small as possible, in order to minimize sensitivity to error (Smola and Scholkopf, 2004). It is usually used for high-dimensional data, where there are a large number of predictive variables. Large datasets can be easily managed. The performance of the algorithm decreases in the presence of noise (Kotsiantis et al., 2007).
- Random Forest (RF). This algorithm is suitable for both regression and classification. It uses *ensemble learning*, combining many DTs to overcome their high variance problems. The insight behind this

approach, called *bagging*, is to take many population training sets, build a separate predictive model using each training set and calculate the average of the resulting predictions. In case few training data are available, *bootstrap* sampling can be carried out, extracting repeated samples from the single training dataset. Furthermore, whenever a division in a tree is considered, RF chooses a random sample of predictors as candidates for the division, not allowing the algorithm to consider most of the predictors. In this way, it is possible to decorrelate the trees, but above all, it involves a reduction in both the test error and the *out-of-bag (OOB)* error (Breiman et al., 1984a; Hastie et al., 2017).

- eXtreme Gradient Boosting (XGBoost). This algorithm is suitable for both regression and classification. It uses *ensemble learning*, combining many DTs. In contrast to RF, this algorithm uses *boosting*. It works similarly to *bagging*, except that trees are grown sequentially and it does not involve *bootstrap* sampling. Each tree is grown using information from previously grown trees, so each tree is estimated on a modified version of the original data. Therefore, learning does not make a complete estimate of the data but learns slowly; given the last estimated model, the next tree is estimated on the residuals of the model. Thus, the model slowly improves in areas where it does not perform well. Slow learning methods are highly accurate (Breiman et al., 1984a; Hastie et al., 2017). In particular, XGBoost was developed to increase speed and performance by introducing a penalty parameter to reduce overfitting. It uses gradient descent regression trees to calculate the optimal values for each leaf and the overall tree score. There is a regularization term, comprising the feature subsampling term and learning rate (Chen and Guestrin, 2016). It is a particularly interesting algorithm for computational speed and accuracy.
- Artificial Neural Network (ANN). This algorithm is suitable for both classification and regression. The central idea is to extract linear combinations of the inputs as derived features, and then model the target as a non-linear function of these features (Hastie et al., 2017). The algorithm consists of different layers of computation units called neurons. The neurons compute a defined function, and their connections are weighted. The first layer is called the inputs layer, the last is the outputs layer, and the others are hidden layers. The number of hidden layers defines the model's complexity and the types of neural networks. The training aims to optimize the weight for links in order to minimize the loss function. ANNs are particularly suitable for highly non-linear systems and require a huge amount of data to converge, proportional to the network complexity. The neural network algorithm can deal with noise and outliers in the dataset (Singh et al., 2016).

Tables 1 and 2 summarize the analysis and the comparison among the above-mentioned supervised (Table 1) and unsupervised (Table 2) algorithms according to their advantages and limitations in the context of PdM applications. Furthermore, in the same tables, some examples of ML algorithm applications for PdM are reported in a non-exhaustive way. The aim is to focus on the most commonly used ML methods applied to PdM, showing which are being explored in this field. This list is the synthesis of the results from several recent literature review papers where a complete overview of the different applications for ML-based PdM is available for readers (Carvalho et al., 2019; Çinar et al., 2020; Dalzochio et al., 2020; Nacchia et al., 2021).

The following list reports the algorithms considered for unsupervised learning.

- K-means clustering (K-MNS). It divides data into K mutually exclusive clusters. The distance from the cluster centre defines the items' probability of belonging to it. It starts by assigning the items to one of the predetermined K clusters and then calculating the K group centroids or by pre-specifying the K group centroids. It selects the average of a cluster's points as its centre (*mean*). Subsequently, to reduce the variation within the cluster, through an iterative procedure, the algorithm tries to minimize the sum of squares within groups (*WGSS*) on all variables, reassigning the items to the different clusters. The procedure stops when no *WGSS* improvements are achieved with further reassignments. It fits large datasets, but its performance decreases when outliers are present in the dataset (Pham et al., 2005; Pham, 2006; Hastie et al., 2017).
- K-medoids clustering (K-MDDS). It resembles the K-MNS clustering but differs in the cluster centre selection. It picks the actual data points from the clusters as their centres (e.g., *medoids*). The term K represents the number of medoids to be identified and the number of clusters that are required. It randomly chooses K points from the input data. K's value can be assessed using methods such as the *silhouette* method. Each data point is assigned to the cluster to which its nearest medoid belongs. For each data point of cluster i, its distance from all other data points is computed and added. The point of i^{th} cluster for which the computed sum of distances from other points is minimal is assigned as the medoid for that cluster. The steps are repeated until convergence is reached, that is, when the medoids stop moving. Since medoids do not become influenced by extremities, the K-MDDS algorithm is more robust to outliers and noise than the K-MNS algorithm (Park and Jun 2009).
- Hierarchical clustering (HC). It is represented as a dendrogram. Its root is the unique cluster that gathers all the samples, the leaves being the clusters with only one sample. It begins by defining a measure of the *dissimilarity* (or *distance* if the triangular inequality holds) of each item to all the others. Which definition of distance to use (*Euclidean, Manhattan, Canberra, Ward's method*, etc. (Ward Jr, 1963)) often depends on the specific application or is a subjective choice. Some distance measures are only appropriate for certain types of data, while others have been introduced to group features rather than observations. The *agglomerative approach* (e.g., *bottom-up*) proceeds iteratively. Starting from the bottom of the dendrogram, each of the n observations is treated as a cluster. The two most similar clusters are merged so that there are now $n-1$ clusters (Hastie et al., 2017; Rokach and Maimon, 2005).
- Density-based spatial clustering of applications with noise (DBSCAN). It estimates the density around each point (item) by counting the number of points in a neighbourhood ε specified by the user, and it applies thresholds to identify the *core*, *border,* and *noise* points. In the second step, the core points are gathered in a cluster if they are *density-reachable* (i.e., if there is a chain of core points where each point falls within the ε around the next). Finally, the edge points are assigned to the clusters. A cluster, therefore, satisfies two properties: all points of the cluster are mutually density-connected, and if a point is density-reachable from any point of the cluster, then it is part of the cluster itself. It is appropriate for data that contain clusters of similar density (Ester et al., 1996; Schubert et al., 2017).
- Gaussian Mixture Model (GMM). This probabilistic model assumes that all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. It generalizes k-means clustering to incorporate information about the covariance structure of the data, as well as the centres of the latent Gaussians. By adopting this approach, the clustering problem becomes that of estimating the parameters of the mixture assumed and then using the estimated parameters to calculate each item's *a posteriori* probability of belonging to the cluster. Furthermore, determining the number of clusters becomes a model selection problem for which objective procedures exist. GMM belongs to model-based clustering methods. Usually, density parameters are estimated with maximum likelihood using the *expectation-maximization* (EM) algorithm (Bishop, 2006; Fraley and Raftery, 2002).

As said, Table 2 reports the above-mentioned unsupervised algorithms according to their advantages, limitations, and application in

Table 1
Comparison of the supervised machine learning algorithms examined.

Algorithm	Type	Advantages	Limitations	Application, Equipment/Systems	Ref.
LLSR	Regression	• Predicting values of a continuous response variable. • It is easy to implement.	• Assumes linear relation. • Suboptimal for non-linear data • Sensitive to outliers. • Prone to underfitting.		
Lasso	Regression	• Regression analysis of continuous response variable. • Performs feature selection by shrinking coefficients towards zero. • Avoids overfitting.	• Assumes linear relation. • Suboptimal for non-linear data. • Assumes normal distributed data. • Selected features can be highly biased. • For different bootstrapped data, the features selected can be very different	RUL estimation; Fault prediction; Bearing; Jet engine blades; Oil analysis of gearbox; Nuclear power plant; CNC machine; Turbofan engine tool; Track; Aircrafts; Rotor bar; Printing machine;	Çinar et al. (2020)
Logit	Binary classification	• Dataset with dependent variable and one or more independent variables. • It is easy to implement. • LR-based models can be updated easily.	• Assumes linearity. • Sensitive to outliers. • Prone to overfitting. • Unless multinomial, generic LR can only classify variables that have two states (i.e., dichotomous)		
LDA	Multiple classification	• Suited for well-separated classes. • It is simple, fast, and portable algorithm.	• Assumes normal distribution and linearity. • Sometimes not good for few categories variables. • Sensitive to outliers. • Prone to overfitting.		
QDA	Multiple classification	• Large training sets with no common covariance matrix. • Robustness to Outliers: they are generally more robust to outliers compared to linear classifiers.	• Assumes Gaussian distribution and unique covariance matrix for each class.		
ARIMA	Regression/ Binary/ Multiple classification	• Stationary time series. •Easy to understand and interpret. • Can handle covariates information. • Suitable for small datasets	• Assumes linear relation and polynomial computational data. • Sensitive to outliers. • Only intended for univariate time series.	RUL estimation; Machine tool of CNC Machine;	(Çinar et al., 2020; Dalzochio et al., 2020)
DT	Regression/ Binary/ Multiple classification	• Non-parametric data; • Complex and non-linear variables. • Suitable for a wide variety of data (such as numeric, nominal, categorical) • Easy to understand and interpret, perfect for visual representation. • Data preparation is easy.	• Unsuitable for large datasets. • Less robust and accurate if other methods are applicable. • Prone to overfitting. • It is very sensitive (High variance) • Require classes to be mutually exclusive.	Time to fail class classification; Fault classification; Turbofan-engine; Rotor bar; Power Transformer; Engine equipped with a rotating shaft; Boiler and heat pump of (HVAC) system;	(Çinar et al., 2020; Dalzochio et al., 2020)
k-NN	Regression/ Binary/ Multiple classification	• It is easy to implement and the training is done in faster manner. • Non-parametric data; can be trained with Euclidian and non-Euclidian distance metrics; not reliant on training data. • One can plug in any distance metric even defined by the user. This allows working with complex objects, like time series, graphs, geographical coordinates …	• Sensitive to high feature number. • Prone to overfitting. • Testing is slow. • Sensitive to noise.	Time to fail class classification; Fault classification; Tungsten filament –Ion implantation; Engine equipped with a rotating shaft; Turbofan engine; Switchgear; Piping/ Structures;	(Carvalho et al., 2019; Çinar et al., 2020; Dalzochio et al., 2020; Nacchia et al., 2021)
GAM	Regression	• Handles linear and non-linear data. • Suitable when number of potential predictors are high.	• Prone to overfitting. • Additive properties can risk losing variable interactions.		
SVM	Regression/ Binary/ Multiple classification	• High-dimensional data with high number of predictors; • Handles large datasets. • Less vulnerability to overfitting issue compared to other techniques. • High accuracy and good generalization ability • Flexible selection of kernel for nonlinearity	• Performance can degrade with the presence of noise. • Computationally expensive for large and complex datasets. • May require deep knowledge while choosing appropriate kernel functions.	Fault classification; Condition monitoring; Tracking gauge deviation; Rotor bar; Aircraft; Track; Railways-Mile track; Rolling bearing; Gas turbine engine; Power Transformer; Electrical power systems; Tungsten filament –Ion implantation; Rotating machine-Gear Box; Building facilities; Nuclear power plant;	(Çinar et al., 2020; Dalzochio et al., 2020)
RF	Regression/ Binary/ Multiple classification	• Useful for high variance datasets (it takes the average value from the	• Feature set must contain signal to avoid guessing.	Time to fail class classification; Fault classification; Fault detection; Printing machine; Rotating machinery;	(Çinar et al., 2020; Dalzochio et al., 2020)

(*continued on next page*)

Table 1 (*continued*)

Algorithm	Type	Advantages	Limitations	Application, Equipment/Systems	Ref.
		outcomes of its constituent decision trees) • Empirically, this ensemble-based classifier performs better than its individual base classifiers	• Large number of trees slows down algorithm. • More complex and computationally expensive • Unsuitable for real-time prediction. • Prone to overfitting. • Less interpretability and sensitivity to outliers.	Turbofan-engine; Rotor bar; Aircraft; Production line and semiconductor; Industrial pumps; Cutting machine; Refrigeration system; Wind turbine; Trucks and buses air; compressors; Induction motor; HDD; Vending machine; Power Transformer; Gas turbine; Machine tool of CNC Machine; Oil analysis of gearbox; Woodworking industrial machines;	
XG-Boost	Regression/ Binary/ Multiple classification	• Suitable for modelling that require improvement over time. • Helps reduce overfitting. • Improved modelling by combining trees. • It is an easy to read and interpret algorithm.	• Can still overfit on small datasets or with high number of trees. • Parameters require proper tuning for optimal performance. • Sensitive to outliers	RUL estimation; Printing machine; Production line and semiconductor; Woodworking industrial machines;	(Çinar et al., 2020; Dalzochio et al., 2020)
ANN	Regression/ Binary/ Multiple classification	• Suitable for non-linear data. • Handles noise and outliers well. • Availability of multiple training algorithms • Suitable for tasks characterized by evolving patterns.	• Prone to overfitting. • Optimal network structure can be trial and error. • They may have characteristics of "black box". • Computationally expensive to train the network for a complex classification problem. • Predictor or independent variables require pre-processing	Fault prediction; Anomaly detection; Condition prediction; RUL estimation; Soft sensing; Predicting control; Tool wear monitoring; Cooling fan; Tracking gauge deviation; Wind turbine; Gas turbine; Building facilities; Packaging robot;	(Carvalho et al., 2019; Çinar et al., 2020; Dalzochio et al., 2020)

Table 2
Comparison of the unsupervised machine learning algorithms examined.

Algorithm	Type	Advantages	Limitations	Application, Equipment/Systems	Ref.
k-MNs	Clustering	• Suitable for large datasets. • Easy to implement.	• Sensitive to outliers. • Uncertainty of initial clustering centres leads to lack of stability. • Must specify number of clusters in advance.	Fault detection; RUL estimation; Exhaust fan; Laser melting; Oil-immersed power transformer; Turbofan engine; Semiconductor manufacturing; Machine motor	Çinar et al. (2020)
k-MDDs	Clustering	• Increased robustness to outliers and noise. • More suitable for smaller datasets.	• Not suitable for clustering non-spherical groups. • As the initial medoids are chosen randomly, the results might vary based on the choice in different runs.	Semiconductor manufacturing;	Çinar et al. (2020)
HC	Clustering	• Better suited for small dataset. • Less stringent assumptions about cluster shape. • Less sensitive to outliers.	• Specific types of data may require specific distance measures. • Sensitive to incomplete or mixed data. • Susceptible to misinterpretation. • Computationally expensive.	Semiconductor manufacturing;	Çinar et al. (2020)
DBSCAN	Clustering	• Appropriate for data with similar density clusters. • Suitable for noise cancellation. • Robust towards outliers' detection. • It performs well with arbitrary shaped clusters.	• Unsuitable for datasets with large variance in clusters. • Not suitable for high-dimension datasets.		
GMM	Clustering	• Assumes data generated from finite number of Gaussian distributions (this allows for more complex cluster shapes compared to K-means clustering). • Less sensitive to scale. • Suitable for small datasets.	• No measure of data point relation to clusters. • It assumes normal distribution. • Computationally expensive.		

PdM field.

4. Framework

The recent trends in the computer science field have made technologies available that were previously only accessible to specialists in the sector. In particular, the IT market provides platforms for data management and ML model development to exploit the knowledge arising from the data in corporate decision-making processes. Thus, the need for guiding those who approach ML during model development is a challenge. However, the theoretical accuracy and the computational speed do not ensure satisfactory results, making the algorithm selection a non-trivial task. This suggests the need to include other decision parameters during this stage of the process. In fact, in the literature, there are multiple algorithms, each of which is better suited to certain conditions of the dataset, as shown in the previous section. In the study of Sala et al. (2018), this challenge is faced by operations in general.

This work focuses on maintenance issues and aims to guide users

who do not know the limits and the opportunities of algorithms in their selection. To achieve this aim, a conceptual framework based on the literature review introduced in the previous section has been defined. The study deals with the main ML algorithms, deep learning algorithms are excluded except for ANN, semi-supervised algorithms, reinforcement learning algorithms and the algorithms for dealing with time series have not been analyzed, except for the ARIMA model. Furthermore, the study scope has been limited to *supervised* and *unsupervised learning*. Since the framework must support those who are approaching ML, drivers have been identified with attributes that are easy to extract and do not require elaborate analyses to be extracted. Decision drivers belonging to the three categories that influence the algorithm selection have been identified: the algorithm learning characteristics, the dataset statistical characteristics, and the PdM characteristics. The results are shown in Table 3, which contains the list of drivers and their descriptions.

In the previous section, it has been pointed out that estimating f has a twofold goal: to make a prediction or to make an inference. These goals are often at odds with each other, forcing the analyst to make a trade-off choice. Non-parametric techniques make the best prediction work; however, they are very difficult and complex to interpret. Algorithms adapt complex interactions between independent variables that are difficult to understand and may not always make sense. In the industrial context, maintenance operators expect to be able to interpret the model's output to justify the decision to be made. Furthermore, the ability to interpret the model builds confidence in these approaches. In this case, it is highlighted how model-based learning approaches meet this need. Hence, the framework must allow the evaluation of the accuracy–interpretability trade-off. Table 4 summarizes the attributes assigned to each algorithm for each decision variable considered. Furthermore, it provides qualitative ratings on the accuracy and the interpretability obtainable with each of them, based on the literature and author experience. As reported in the table, the ensemble methods (RF, XGBoost) are non-parametric, with good accuracy and low interpretability. However, some techniques can be applied to obtain the features important for prediction by improving the interpretability of the model. The Shapley Additive Explanations (ShAP) approach is considered the most robust for this goal (Shapley, 1953; Lundberg and Lee, 2017). Finally, accuracy is closely linked to data quality and maturity (Arena et al., 2022). In fact, even if all the other conditions would allow applying a non-parametric algorithm, with poor data quality, good performance is not achievable.

Deepening the decision variables related to ML and their relation to dataset ones, the type of learning represents the primary algorithm's characteristic. Supervised learning involves regression and binary or multiple classifications, while activity learning associated with unsupervised algorithms entails clustering. The response type is the dataset characteristic that guides the choice of the activity learning attribute. A quantitative response is closely associated with regression, a binary response is associated with binary classification, and a categorical response is associated with multiple classifications. If the response is unavailable (unlabelled dataset), it is necessary to use an algorithm for pattern recognition (i.e., an algorithm clustering).

Regarding the objective function, as explained in the previous section, a model-based approach requires a smaller amount of data for features and vice versa for the non-parametric approach. However, when input and output are in a linear relation, both the algorithms based on a linear model and those based on complex models allow high accuracy to be achieved.

This leads to preference for the former as they are easier to compute and interpret. In general, the algorithm's complexity must reflect the system's complexity. In identifying linearity in a dataset, applying a linear model is necessary for analyzing the residuals' graph. Ideally, the residuals' graph shows an unrecognizable pattern; on the contrary, the presence of a pattern can indicate a linearity problem. Finally, imbalanced datasets must deal with non-robust-to-outliers algorithms or

Table 3
Decision variables for each category identified.

Category	Decision variable	Attribute description
Machine learning	Learning type	*Supervised* and *unsupervised* approaches are analyzed. The former refers to a labeled dataset, the letter to an unlabelled one.
	Learning activity	is the task that the model must carry out. Four types of learning activities are considered: *regression, binary classification, multiple classification,* and *clustering*.
	Objective function	is the function type fitted by the algorithm. Two types of objective functions are considered: *model-based* (e.g., parametric) and *non-parametric*.
Dataset	Response type	is the output of the analysis (e.g., *Y*). Four response types are considered: *binary, categorical, continuous,* and *unavailable*. The binary response can assume two values (e.g., fault/non-fault). A categorical attribute has a finite number or a countable infinite set of values normally represented by integers or labels (e.g., generator failure, bearing failure, etc.). Quantitative or continuous data can assume any real value (e.g., bearing temperature, RUL, etc.).
	Variance	identifies the dataset balancing, that is, if the dataset includes values that are statistically identified as outliers since they are rare compared with the population; however, they represent a system state to be included in the analysis (e.g., representative of fault conditions). Two attributes are considered by variance: *balanced* and *imbalanced*. An imbalanced dataset means that the ratio between the majority class samples (non-fault) and the minority class samples is greater than 10 (Lee et al., 2016).
	Linearity	identifies the dataset linearity. Quantities are in a linear relation if there is some form of direct proportionality between them. Two attributes for this driver are considered: *linear* and *non-linear*.
	Observations/ Features ratio	identifies the dataset numerosity based on the predictors included as model inputs. Two attributes are considered: $n \approx p$ and $n \gg p$. Several studies provide statistical methods to estimate the samples number to be included in the model training for the model's accuracy; however, these methods require elaborate analyses. Hence, in this study, qualitative attributes are proposed to guide the initial decision in order to restrict the initial application scope.
Predictive maintenance	Fault type	two types of objective functions are considered: *incipient* and *critical*. See the previous section for details.
	Model objective	four types of model objectives are considered: *RUL estimation, fault detection, diagnosis,* and *state detection*
	States number	identifies the states number to discover based on the behaviour of the system. This driver is considered in the event of the unlabelled dataset when the *a priori* failures definition is not possible for fitting the model. The model aims to identify the different working conditions of the system, thus the fault. Two types of states number are considered: *known* and *missing*.

Table 4
ML and dataset characteristics of selected learning methods, accuracy, and interpretability. Key ▲=good, ♦=fair, ▼=poor.

ML Algorithm	ML Characteristics			Data Characteristics				Accuracy	Interpretability	Ref.
	Learning Activity	Learning Type	Objective Function	Response Type	Variance	Linearity	n/p			
LLSR	Supervised	Regression	Model-based	Continuous	Balanced	▲	▼	▼	▲	(Dreiseitl and Ohno-Machado, 2002; Pham, 2006)
Lasso	Supervised	Regression	Model-based	Continuous	Balanced	▲	▼	▼	▲	Agresti (2003)
Logit	Supervised	Binary Classification	Model-based	Binary	Balanced	▲	▼	▼	▲	(Agresti, 2003; Hastie et al., 2017)
LDA	Supervised	Multiple Classification	Model-based	Categorical	Balanced	▲	▼	▼	▲	(Mitchell, 1997; Balakrishnama and Ganapathiraju, 1998; Hastie et al., 2017)
QDA	Supervised	Multiple Classification	Model-based	Categorical	Balanced	▲	▼	▼	▲	(Duda et al., 1973; Hastie et al., 2017)
ARIMA	Supervised	Regression	Model-based	Continuous	Balanced	▲	▼	▼	▲	Hamilton (2020)
DT	Supervised	Regression/Binary Classification/ Multiple Classification	Non-parametric	Continuous/ Binary/ Categorical	Balanced	▼	♦	▼	▲	Breiman et al. (1984b)
k-NN	Supervised	Regression/Binary Classification/ Multiple Classification	Non-parametric	Continuous/ Binary/ Categorical	Imbalanced	♦	♦	♦	♦	(Kotsiantis et al., 2007; Hastie et al., 2017)
GAM	Supervised	Regression	Non-parametric	Continuous	Balanced	▲	▼	♦	♦	Hastie and Tibshirani (2017)
SVM	Supervised	Regression/Binary Classification/ Multiple Classification	Non-parametric	Continuous/ Binary/ Categorical	Imbalanced	▲	▲	▲	▼	(Smola and Scholkopf, 2004; Kotsiantis et al., 2007; Hofmann et al., 2008)
RF	Supervised	Regression/Binary Classification/ Multiple Classification	Non-parametric	Continuous/ Binary/ Categorical	Balanced	▼	▲	▲	▼	(Breiman et al., 1984a; Hastie et al., 2017)
XG-Boost	Supervised	Regression/Binary Classification/ Multiple Classification	Non-parametric	Continuous/ Binary/ Categorical	Balanced	▼	▲	▲	▼	(Breiman et al., 1984a; Chen and Guestrin, 2016; Hastie et al., 2017)
ANN	Supervised	Regression/Binary Classification/ Multiple Classification	Non-parametric	Continuous/ Binary/ Categorical	Balanced	▲	▲	▲	▼	(Singh et al., 2016; Hastie et al., 2017)
k-MNs	Unsupervised	Clustering	–	–	–	–	–	–	–	(Pham et al., 2005; Hastie et al., 2017)
k-MDDs	Unsupervised	Clustering	–	–	–	–	–	–	–	Park and Jun (2009)
GMM	Unsupervised	Clustering	–	–	–	–	–	–	–	(Fraley and Raftery, 2002; Bishop, 2006)
HC	Unsupervised	Clustering	–	–	–	–	–	–	–	(Ward Jr, 1963; Rokach and Maimon, 2005; Hastie et al., 2017)
DBSCAN	Unsupervised	Clustering	–	–	–	–	–	–	–	(Ester et al., 1996; Schubert et al., 2017)

instance-based ones, such as k-NN, while their application to balanced datasets needs outliers' removal in the pre-processing phase.

The purpose of this study is to address the algorithm selection for PdM issues; thus, there is a need to analyze how the related decision variables vary based on those analyzed so far.

How to approach the maintenance objective is closely linked to the *fault type*. In fact, for failures due to wear and tear, the approach must consider the system's evolution over time. This means that during the feature selection and extraction phase, the analyst must include or build predictors representing this information; alternatively, the analyst must use algorithms for time-series forecasting (i.e., with memory algorithms). This study investigates the former requirement, reporting just the ARIMA example to cover the latter, postponing an in-depth analysis of time-series forecasting algorithms for future studies. The symptoms of wear-and-tear failures or incipient ones manifest themselves slowly, evolving, making these failures highly detectable through the analysis of weak signals. On the contrary, critical failures appear abruptly, and a prognostic approach would be technically unfeasible. In these cases, although it is not possible to accurately estimate the component/system RUL, the analysis and modelling of the CMS data allow increasing the detectability of faults.

Fault detection or diagnosis approaches can detect anomalous behaviour's in advance and associate them with anomalies already verified in the past, allowing intervention in advance and in a prompt manner, with preventive maintenance actions. The ML-based PdM strategy allows the greatest advantages for this type of failure.

Estimating the RUL means predicting the days/hours of the system's residual operation, which is a continuous variable; thus, the regression task is suitable. Fault detection and diagnosis are respectively associated with binary and multiple classifications. The literature cites several examples of fault detection conducted with semi-supervised approaches. However, this study does not investigate these types of approaches, postponing the demonstration of their general validity for these learning types for future studies.

In cases where a database of events associated with faults is unavailable, a supervised approach is not possible. For this reason, the objective of identifying the system states is associated with clustering activities. The study of the failure modes of the component/system could allow for identifying how many and which operating states, or degradation, should be identified in the dataset (*states number*). In this case, it is possible to apply algorithms that require defining the number of patterns in advance (K-MNS, K-MDDS, GMM, and DBSCAN). Alternatively, it is necessary to apply more interpretable algorithms in which it is possible to subsequently define the pruning level in order to assign a sample to a specific pattern. The variable *states number* is introduced to distinguish between these two cases only in unsupervised approaches.

What has been discussed so far has led to the definition of the framework in Fig. 1. The framework first classifies the supervised algorithms presented in the paper in a set of dimensional attributes to aid in decision-making concerning choosing an appropriate algorithm in different scenarios. The included dimensions are objective function or model-based vs. non-parametric methods, similar or much larger set of observations vs. number of features, algorithms are classified by the quality and maturity of available datasets, and methods are considered by output accuracy, linearity of variables and features and how easy it is to interpret algorithm output. The ShAP methods dimension describes how interpretability can be improved by applying methods like the Shapley Additive Explanations (Lundberg and Lee, 2017). The algorithms are then color-coded for fault type scenarios where they are applicable, with blue for RUL scenarios, orange for fault detection, and green for diagnosis. An underlined algorithm name describes variance imbalance and balanced variance with no underline. Naming in bold describes a linear method, and normal naming describes non-linear. The non-supervised algorithms are separated by interpretability, a square for known states and a circle for missing states. The yellow coloring describes the algorithms' role in status identification. The legend summarizes the associations among *learning activity, learning type, response type, fault type,* and *maintenance objective*. The next section shows two examples of the legend's usefulness in identifying the *maintenance objective*, that is, the color that identifies a limited selection of algorithms to choose from. The final selection is based on *observation/features ratio, data quality, and maturity* and *variance* considerations. The *states number* is the variable to consider if the dataset is unlabelled. Finally, accuracy and interpretability can also be assessed in accordance with the decision-making function of the model.

5. Illustrative case study

The purpose of this study is to identify a valid approach for maintenance issues, leaving the analyst the task to generalize the issue characteristic parameters he has to face. Following, two approaches of the framework usage are presented, with the aim of demonstrating how the proposed framework can guide the algorithm selection; however, there are no limits for its application scope based on specific data or assets.

Case A shows how the framework can support the technical evaluation of the ML-based PdM starting from the definition of the failure modes that affect the system. On the other hand, case B describes the assessments that can be made by the available data.

5.1. Case A

A company wants to evaluate the implementation of ML-based PdM in a production line. There is no CMS in the company, however, data relating to the maintenance carried out are available. The equipment analysis phase, based on Florian et al. (2019), is performed on a small number of critical components. The maintenance and intervention costs data allows for applying the economic model proposed by Florian et al. (2021) for evaluating the threshold investment cost and the performances that the ML model should ensure for the ML-based PdM to be sustainable. The proposed framework can guide the evaluation of the technical requirements required for achieving the desired performance. In the case described the type of fault and data on maintenance events are available. If the system is subject to wear or incipient failures, all maintenance objectives are achievable, however, since the labels are available, the supervised approaches are preferred. Among the three, the prognostic models allow a greater optimization of the maintenance process, in fact, the costs reduction is due to both the decrease in failure interventions (concerning the performance of the model) and the organizational aspects optimization (e.g. spare parts purchase, maintenance workers assignment, production stop, intervention procedures preparation) (Vogl et al., 2016). In the same way, the fault detection and diagnosis models avoid fault interventions, while the process optimization is not guaranteed as the time to the fault is not a model output.

The system failure modes engineering analysis allows to estimate the system linearity, the framework in these cases suggests that it is possible to apply algorithms that require a reduced number of observations, e.g. there is no need to waste resources for storing high-frequency sampling data or allocating excessive computational resources for model development (in accordance with the failure mode behaviour). On the contrary, for complex systems, there is a need of investing in IT infrastructures to ensure the transmission reliability of the good quality data, but above all, it is necessary to ensure a huge amount of data for the models training or re-training.

In general, using the proposed framework, a maintenance manager can carry out a technical evaluation of the ML-based PdM to support the appropriate economic evaluations, even without having in-depth data analysis knowledge.

5.2. Case B

A company wants to evaluate the implementation of the ML-based PdM on a production line whose data is collected in a CMS, with the maintenance data. As in the previous case, the critical components for which to prioritize the PdM can be identified.

Failure mode assessment allows the feature selection to be carried out for obtaining a labeled training dataset. Several scenarios can occur: (i) the time to failure is a reliable measure, thus a quantitative answer is available that allows for developing a model for the RUL estimation; (ii) the event data allows to accurately identify the different failure modes, thus a model for the diagnosis of the failures can be implemented; (iii) the data related to the faults allow to identify only if the system was functioning or not, therefore a model for fault detection can be implemented.

Furthermore, the observations/characteristics ratio and the balancing of the dataset can be identified for evaluating, firstly, the technical feasibility, secondly which algorithms to apply, thus avoiding the waste of resources that would occur with a try and error approach.

6. Conclusions

In recent years, data-driven approaches, thanks to the spread of ML techniques, have emerged as the most promising to accurately predict and diagnose the system's health conditions during operations. ML-based PdM provides many competitive benefits, such as maintenance cost reduction, higher product quality, system efficiency, and improved

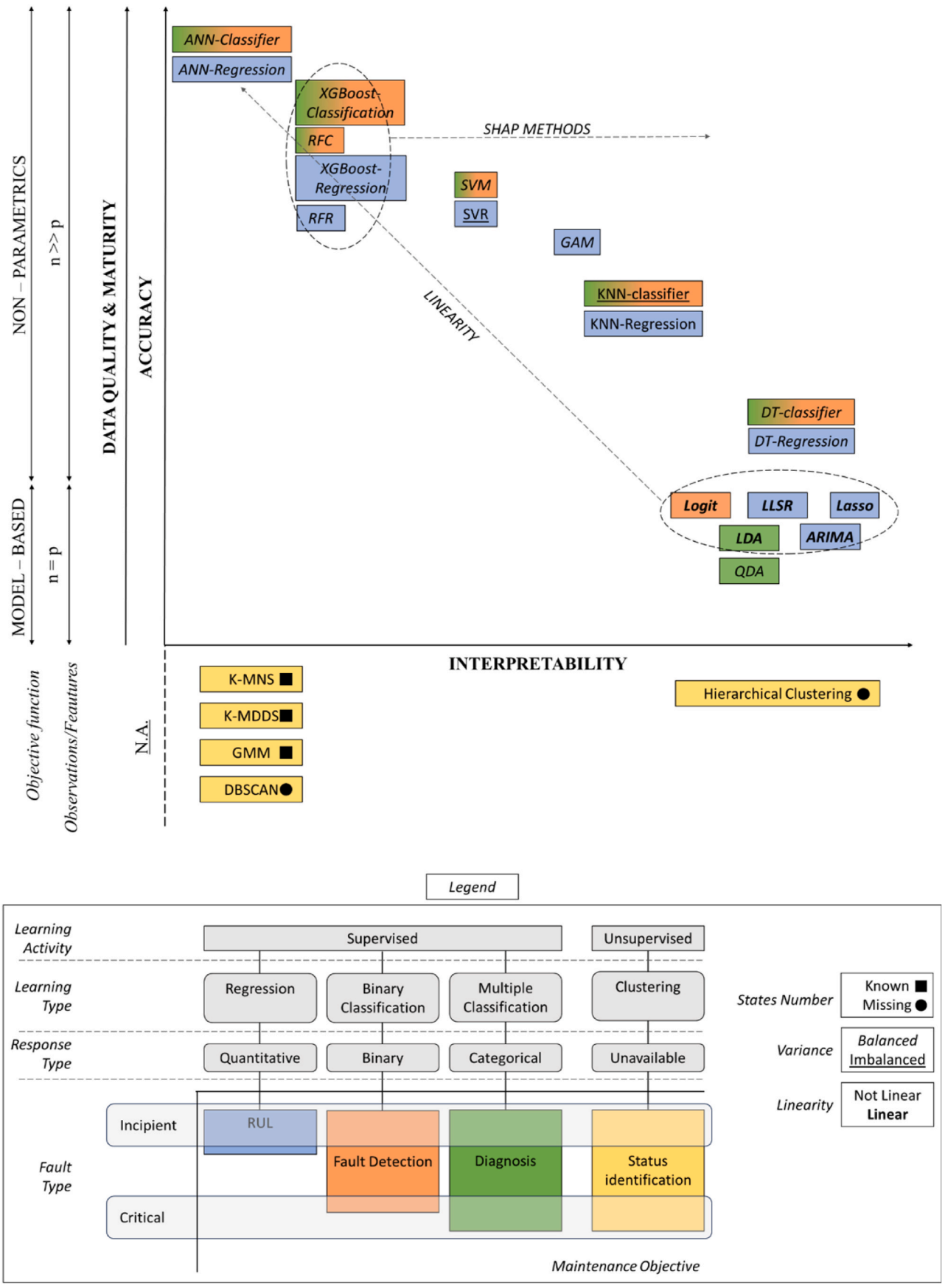

Fig. 1. ML-based PdM algorithm selection framework.

availability. However, when considering real industrial applications, implementing an ML-based PdM strategy presents multiple challenges, such as the investment costs, the connection of physical assets, the storage of the data, extraction of valuable information, and the development of precise ML algorithms. To overcome these constraints, it is essential to identify the applicability domain in terms of technical, organizational, and economic assessment of this maintenance strategy. Thus, this work provides a conceptual framework aimed at guiding users in the selection of one or more suitable ML algorithms to use in their analyses, with the purpose of establishing the technical ML-based PdM feasibility.

The conceptual framework allows for systematizing the knowledge, about the algorithmic approach patterns and failure characteristics that could affect their choice and for defining the variables that affect algorithmic choice. The parameters were classified into three main categories based on their nature: data set, ML, and maintenance. The decision variables presented in the selection framework constitute a solid base for the selection of proper ML algorithms since they are easily recognizable and do not necessitate deep analysis to be identified. The associations between decision variables (learning activity, learning type, response type, fault type, and maintenance objective) are analyzed and designed. The final selection is based on observation/features ratio, data quality, and maturity and variance considerations. The state's number is the variable to consider if the dataset is unlabelled. Finally, accuracy and interpretability can also be assessed in accordance with the decision-making function of the model.

The proposed selection framework does not claim to be complete with respect to the available literature on the topic and/or to suggest the best algorithm to the user due to the vast complexity characterizing the ML field of research and application. Thus, the basic idea is to develop a simple and easy tool to choose the right approach that fits the specific purpose at hand through an agile and informed decision-making process.

The research presented leaves possible future improvements. First, the current pool of ML algorithms lacks Semi-Supervised approaches. Furthermore, deep-learning algorithms, reinforcement learning algorithms and time-series forecasting could be also considered in the framework. Future work could encompass multiple aspects not considered, starting from the validation strategy, which should be chosen carefully to be effective, and continuing with the selection framework extension and refinement.

As part of future work, there are factors out of scope for this article that are still important to consider for users of ML-oriented approaches. This includes a holistic and comprehensive view on the multifaceted domain of ML that takes into account the role of humans, social interactions, and close interrelations with cutting-edge technologies and decision-making. One of these factors is a thorough description of the requirements needed to implement ML methods. Bokrantz et al. (2020) describe the overall requirements by defining Smart Maintenance as « *an organizational design for managing maintenance of manufacturing plants in environments with pervasive digital technologies*» and point to the dimensions of data-driven decision-making, human capital, and both internal and external integration as required for Smart Maintenance transition. Both Smart Maintenance and ML can be further connected to the ongoing efforts towards digitalization and Industry 4.0 (I4.0). Acatech (2020) describes this development as a 6-stage process, where digitalization is shown through computerization and connectivity, and I4.0 through the steps of visibility, transparency, predictive capability, and adaptability. This stepwise depiction of I4.0 is useful to develop further to ensure improved transitioning from classic maintenance towards Smart Maintenance and to fully utilize the untapped potential of ML methods.

CRediT authorship contribution statement

Simone Arena: Conceptualization, Investigation, Methodology, Writing – original draft, Writing – review & editing. **Eleonora Florian:** Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Visualization, Writing – original draft, Writing – review & editing. **Fabio Sgarbossa:** Conceptualization, Investigation, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. **Endre Sølvsberg:** Conceptualization, Investigation, Methodology, Writing – original draft, Writing – review & editing. **Ilenia Zennaro:** Conceptualization, Investigation, Methodology, Supervision, Validation, Visualization, Writing – original draft, 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

No data was used for the research described in the article.

References

ACATECH, 2020. Industrie 4.0 Maturity Index - Managing the Digital Transformation of Companies.
Accorsi, R., Manzini, R., Pascarella, P., Patella, M., Sassi, S., 2017. Data mining and machine learning for condition-based maintenance. Procedia Manuf. 11, 1153–1161. https://doi.org/10.1016/j.promfg.2017.07.239.
Agresti, A., 2003. Categorical Data Analysis. John Wiley & Sons, Ltd.
Alabi, M.O., Nixon, K., Botef, I., 2018. A survey on recent applications of machine learning with big data in additive manufacturing industry. Am. J. Eng. Appl. Sci. 11, 1114–1124. https://doi.org/10.3844/ajeassp.2018.1114.1124.
Arena, S., Florian, E., Zennaro, I., Orrù, P.F.F., Sgarbossa, F., 2022. A novel decision support system for managing predictive maintenance strategies based on machine learning approaches. Saf. Sci. 146, 105529 https://doi.org/10.1016/j.ssci.2021.105529.
Balakrishnama, S., Ganapathiraju, A., 1998. Linear discriminant analysis-a brief tutorial. Institute for Signal and information Processing 18, 1–8.
Berrendero, J.R., Bueno-Larraz, B., Cuevas, A., 2020. On mahalanobis distance in functional settings. J. Mach. Learn. Res. 21, 1–33.
Bertolini, M., Mezzogori, D., Neroni, M., Zammori, F., 2021. Machine Learning for industrial applications: a comprehensive literature review. Expert Syst. Appl. 175, 114820 https://doi.org/10.1016/j.eswa.2021.114820.
Bishop, C.M., 2006. Pattern recogniton and machine learning. Information Science and Statistics. Springer-Verlag, New York, New York.
Bokrantz, J., Skoogh, A., Berlin, C., Wuest, T., Stahre, J., 2020. Smart Maintenance: a research agenda for industrial maintenance management. Int. J. Prod. Econ. 224, 107547 https://doi.org/10.1016/j.ijpe.2019.107547.
Breiman, L., Friedman, J.H., Olshen, A., Stone, C.J., 1984a. Classification and Regression Trees. Wadsworth, Belmont.
Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J., 1984b. Classification and Regression Trees. Wadsworth, Belmont.
Burkov, A., 2019. The Hundred-Page Machine Learning Book, first ed. Publishing Kindle Direct.
Cabana, E., Lillo, R.E., Laniado, H., 2021. Multivariate outlier detection based on a robust Mahalanobis distance with shrinkage estimators. Stat. Pap. 62, 1583–1609. https://doi.org/10.1007/s00362-019-01148-1.
Carvalho, T.P., Soares, F.A.A.M.N., Vita, R., Francisco, R. da P., Basto, J.P., Alcalá, S.G.S., 2019. A systematic literature review of machine learning methods applied to predictive maintenance. Comput. Ind. Eng. 137, 106024 https://doi.org/10.1016/j.cie.2019.106024.
Chen, T., Guestrin, C., 2016. XGBoost: a scalable tree boosting system. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 13-17-Augu, pp. 785–794. https://doi.org/10.1145/2939672.2939785.
Çinar, Z.M., Nuhu, A.A., Zeeshan, Q., Korhan, O., Asmael, M., Safaei, B., 2020. Machine learning in predictive maintenance towards sustainable smart manufacturing in industry 4.0. Sustainability 12. https://doi.org/10.3390/su12198211.
Dalzochio, J., Kunst, R., Pignaton, E., Binotto, A., Sanyal, S., Favilla, J., Barbosa, J., 2020. Machine learning and reasoning for predictive maintenance in Industry 4.0: current status and challenges. Comput. Ind. 123 https://doi.org/10.1016/j.compind.2020.103298.
Dashdondov, K., Kim, M.H., 2023. Mahalanobis distance based multivariate outlier detection to improve performance of hypertension prediction. Neural Process. Lett. 55, 265–277. https://doi.org/10.1007/s11063-021-10663-y.
Dogan, A., Birant, D., 2021. Machine learning and data mining in manufacturing. Expert Syst. Appl. 166, 114060 https://doi.org/10.1016/j.eswa.2020.114060.
Dreiseitl, S., Ohno-Machado, L., 2002. Logistic regression and artificial neural network classification models: a methodology review. J. Biomed. Inf. 35, 352–359. https://doi.org/10.1016/S1532-0464(03)00034-0.

Duda, R.O., Hart, P.E., Stork, D.G., 1973. Pattern Classification and Scene Analysis. Wiley, New York.
Ester, M., Kriegel, H.P., Sander, J., Xu, X., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD-96 Proceedings, pp. 226–231.
Florian, E., Sgarbossa, F., Zennaro, I., 2021. Machine learning-based predictive maintenance: a cost-oriented model for implementation. Int. J. Prod. Econ. 236, 108114 https://doi.org/10.1016/j.ijpe.2021.108114.
Florian, E., Sgarbossa, F., Zennaro, I., 2019. Machine learning for predictive maintenance: a methodological framework. Proceedings of the Summer School Francesco Turco 1, 194–200.
Fraley, C., Raftery, A.E., 2002. Model-based clustering, discriminant analysis, and density estimation. J. Am. Stat. Assoc. 97, 611–631. https://doi.org/10.1198/016214502760047131.
Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT press, Cambridge.
Grbić, R., Slišković, D., Kadlec, P., 2013. Adaptive soft sensor for online prediction and process monitoring based on a mixture of Gaussian process models. Comput. Chem. Eng. 58, 84–97. https://doi.org/10.1016/j.compchemeng.2013.06.014.
Gupta, P., Sharma, A., Jindal, R., 2016. Scalable machine-learning algorithms for big data analytics: a comprehensive review. WIREs Data Min. Knowl. Discovery 6, 194–214. https://doi.org/10.1002/widm.1194.
Hamilton, J.D., 2020. Time Series Analysis. Princeton university press.
Hastie, T., Tibshirani, R., Friedman, J., 2017. The Elements of Statistical Learning. Springer Series in Statistics.
Hastie, T.J., Tibshirani, R.J., 2017. Generalized Additive Models. Routledge, London.
Hofmann, T., Schölkopf, B., Smola, A.J., 2008. Kernel methods in machine learning. Ann. Stat. 36, 1171–1220.
ISO 13374-2, 2021. Condition Monitoring and Diagnostics of Machines — Data Processing, Communication, and Presentation — Part 2: Data Processing ([WWW Document]).
Jahani, H., Jain, R., Ivanov, D., 2023. Data science and big data analytics: a systematic review of methodologies used in the supply chain and logistics research. Ann. Oper. Res. https://doi.org/10.1007/s10479-023-05390-7.
Jain, A.K., Chandrasekaran, B., 1982. 39 Dimensionality and sample size considerations in pattern recognition practice. Handb. Stat. 2, 835–855. https://doi.org/10.1016/S0169-7161(82)02042-2.
Jardine, Andrew K.S., Lin, D., Banjevic, D., 2006. A review on machinery diagnostics and prognostics implementing condition-based maintenance. Mech. Syst. Signal Process. 20, 1483–1510. https://doi.org/10.1016/j.ymssp.2005.09.012.
Javed, K., Gouriveau, R., Zerhouni, N., 2017. State of the art and taxonomy of prognostics approaches, trends of prognostics applications and open issues towards maturity at different technology readiness levels. Mech. Syst. Signal Process. 94, 214–236. https://doi.org/10.1016/j.ymssp.2017.01.050.
Jordan, M.I., Mitchell, T.M., 2015. Machine learning: trends, perspectives, and prospects. Science 349, 255–260.
Kiangala, K.S., Wang, Z., 2018. Initiating predictive maintenance for a conveyor motor in a bottling plant using industry 4.0 concepts. Int. J. Adv. Des. Manuf. Technol. 97, 3251–3271.
Kim, T.S., Sohn, S.Y., 2021. Multitask learning for health condition identification and remaining useful life prediction: deep convolutional neural network approach. J. Intell. Manuf. 32, 2169–2179. https://doi.org/10.1007/s10845-020-01630-w.
Kotsiantis, S.B., Zaharakis, I., Pintelas, P., 2007. Supervised machine learning: a review of classification techniques. Emerging artificial intelligence applications. Comput. Eng. 160, 3–24.
Lee, T., Lee, K.B., Kim, C.O., 2016. Performance of machine learning algorithms for class-imbalanced process fault detection problems. IEEE Trans. Semicond. Manuf. 29, 436–445. https://doi.org/10.1109/TSM.2016.2602226.
Lei, Y., Yang, B., Jiang, X., Jia, F., Li, N., Nandi, A.K., 2020. Applications of machine learning to machine fault diagnosis: a review and roadmap. Mech. Syst. Signal Process. 138, 106587 https://doi.org/10.1016/j.ymssp.2019.106587.
Leukel, J., González, J., Riekert, M., 2021. Adoption of machine learning technology for failure prediction in industrial maintenance: a systematic review. J. Manuf. Syst. 61, 87–96. https://doi.org/10.1016/j.jmsy.2021.08.012.
Liu, R., Yang, B., Zio, E., Chen, X., 2018. Artificial intelligence for fault diagnosis of rotating machinery: a review. Mech. Syst. Signal Process. 108, 33–47. https://doi.org/10.1016/j.ymssp.2018.02.016.
Lundberg, S.M., Lee, S.I., 2017. A unified approach to interpreting model predictions. Adv. Neural Inf. Process. Syst. 4766–4775, 2017-December.
Malhotra, R., 2015. A systematic review of machine learning techniques for software fault prediction. Applied Soft Computing Journal 27, 504–518. https://doi.org/10.1016/j.asoc.2014.11.023.
Martin, K.F., 1994. A review by discussion of condition monitoring and fault diagnosis in machine tools. Int. J. Mach. Tool Manufact. 34, 527–551. https://doi.org/10.1016/0890-6955(94)90083-3.
Mishra, C., Gupta, D.L., 2017. Deep machine learning and neural networks: an overview. IAES Int. J. Artif. Intell. 6, 66. https://doi.org/10.11591/ijai.v6.i2.pp66-73.
Mitchell, T., 1997. Machine Learning. McGraw-Hill Education.
Mitchell, T.M., 1999. Machine learning and data mining. Commun. ACM 42, 30–36. https://doi.org/10.1145/319382.319388.
Mobley, R.K., 2002. Introduction to Predictive Maintenance, second ed. Elsevier Science, Woburn, MA.
Nacchia, M., Fruggiero, F., Lambiase, A., Bruton, K., 2021. A systematic mapping of the advancing use of machine learning techniques for predictive maintenance in the manufacturing sector. Appl. Sci. 11, 1–34. https://doi.org/10.3390/app11062546.
Palacín, I., Gibert, D., Planes, J., Arena, S., Orrù, P.F., Melis, M., Annis, M., 2021. Anomaly detection for diagnosing failures in a centrifugal compressor train. Frontiers in Artificial Intelligence and Applications 339, 217–220. https://doi.org/10.3233/FAIA210137.
Park, H.S., Jun, C.H., 2009. A simple and fast algorithm for K-medoids clustering. Expert Syst. Appl. 36, 3336–3341. https://doi.org/10.1016/j.eswa.2008.01.039.
Paturi, U.M.R., Cheruku, S., 2020. Application and performance of machine learning techniques in manufacturing sector from the past two decades: a review. Mater. Today: Proc. 38, 2392–2401. https://doi.org/10.1016/j.matpr.2020.07.209.
Pham, D.T., Dimov, S.S., Nguyen, C.D., 2005. Selection of K in K-means clustering. Proc. IME C J. Mech. Eng. Sci. 219, 103–119. https://doi.org/10.1243/095440605X8298.
Pham, H., 2006. Springer Handbook of Engineering Statistics. Springer Science & Business Media, London.
Pillai, S., Vadakkepat, P., 2021. Deep learning for machine health prognostics using Kernel-based feature transformation. J. Intell. Manuf. https://doi.org/10.1007/s10845-021-01747-6.
Prajapati, A., Bechtel, J., Ganesan, S., 2012. Condition based maintenance: a survey. J. Qual. Mainten. Eng. 18, 384–400.
Quatrini, E., Costantino, F., Di Gravio, G., Patriarca, R., 2020. Condition-based maintenance-An extensive literature review. Machines 8. https://doi.org/10.3390/MACHINES8020031.
Rai, R., Tiwari, M.K., Ivanov, D., Dolgui, A., 2021. Machine learning in manufacturing and industry 4.0 applications. Int. J. Prod. Res. 59, 4773–4778. https://doi.org/10.1080/00207543.2021.1956675.
Rokach, L., Maimon, O., 2005. Clustering methods. In: Data Mining and Knowledge Discovery Handbook. Springer, Boston, pp. 321–352.
Rolf, B., Jackson, I., Müller, M., Lang, S., Reggelin, T., Ivanov, D., 2023. A review on reinforcement learning algorithms and applications in supply chain management. Int. J. Prod. Res. 61, 7151–7179. https://doi.org/10.1080/00207543.2022.2140221.
Ruiz-Sarmiento, J.R., Monroy, J., Moreno, F.A., Galindo, C., Bonelo, J.M., Gonzalez-Jimenez, J., 2020. A predictive model for the maintenance of industrial machinery in the context of industry 4.0. Eng. Appl. Artif. Intell. 87, 103289 https://doi.org/10.1016/j.engappai.2019.103289.
Russell, S., Norvig, P., 2002. Artificial Intelligence: a Modern Approach. Pearson, London.
Sala, R., Zambetti, M., Pirola, F., Pinto, R., 2018. How to select a suitable machine learning algorithm: a feature-based, scope-oriented selection framework. In: Proceedings of the Summer School Francesco Turco, pp. 87–93.
Saxena, A., Prasad, M., Gupta, A., Bharill, N., Patal, O.P., Tiwari, A., Er, M.J., Ding, W., Lin, C.-T., 2017. A review of clustering techniques and developments. Neurocomputing 267, 664–681. https://doi.org/10.1016/j.neucom.2017.06.053.
Schubert, E., Sander, J., Ester, M., Kriegel, H.P., Xu, X., 2017. DBSCAN revisited, revisited: why and how you should (still) use DBSCAN. ACM Trans. Database Syst. 42 https://doi.org/10.1145/3068335.
Settles, B., 2009. Active learning literature survey. Computer Sciences Technical Report 1648.
Shapley, L.S., 1953. 17. A value for n-person games. In: Contributions to the Theory of Games (AM-28) Volume II. Princeton university press, Princeton. https://doi.org/10.1515/9781400881970-018.
Silva, T.C., Zhao, L., 2016. Machine Learning in Complex Networks. Springer International Publishing, Cham, Switzerland.
Singh, A., Thakur, N., Sharma, A., 2016. A review of supervised machine learning algorithms. In: 3rd International Conference on Computing for Sustainable Global Development (INDIACom). IEEE, pp. 1310–1315.
Smola, A.J., Scholkopf, B., 2004. A tutorial on support vector regression. Stat. Comput. 14, 199–222.
Sølvsberg, E., Arena, S., Sgarbossa, F., Schjølberg, P., 2023. Identifying customer returns in a printed circuit board production line using the mahalanobis distance. In: IFIP International Conference on Advances in Production Management Systems. Springer, pp. 426–438.
Strutz, T., 2011. Data Fitting and Uncertainty: A Practical Introduction to Weighted Least Squares and beyond. Vieweg Teubner, Wiesbaden, Germany.
Susto, G.A., Schirru, A., Pampuri, S., McLoone, S., Beghi, A., 2015. Machine learning for predictive maintenance: a multiple classifier approach. IEEE Trans. Ind. Inf. 11, 812–820.
Sutton, R.S., Barto, A.G., 2018. Reinforcement learning: an introduction. Bradford Books. MIT Press Cambridge.
Tao, F., Wang, T., Wu, J., Lin, X., 2020. A novel KA-STAP method based on Mahalanobis distance metric learning. Digit. Signal Process.: A Review Journal 97, 102613. https://doi.org/10.1016/j.dsp.2019.102613.
van Engelen, J.E., Hoos, H.H., 2020. A survey on semi-supervised learning. Mach. Learn. 109, 373–440. https://doi.org/10.1007/s10994-019-05855-6.
Vogl, G.W., Weiss, B.A., Helu, M., 2016. A review of diagnostic and prognostic capabilities and best practices for manufacturing. J. Intell. Manuf. 30, 79–95. https://doi.org/10.1007/s10845-016-1228-8.
Ward Jr, J.H., 1963. Hierarchical grouping to optimize an objective function. J. Am. Stat. Assoc. 58, 236–244. Taylor & Francis.
Wuest, T., Weimer, D., Irgens, C., Thoben, K.D., 2016. Machine learning in manufacturing: advantages, challenges, and applications. Production and Manufacturing Research 4, 23–45. https://doi.org/10.1080/21693277.2016.1192517.
Xu, Y., Sun, Y., Liu, X., Zheng, Y., 2019. A digital-twin-assisted fault diagnosis using deep transfer learning. IEEE Access 7, 19990–19999. https://doi.org/10.1109/ACCESS.2018.2890566.
Xu, Z., Saleh, J.H., 2021. Machine learning for reliability engineering and safety applications: review of current status and future opportunities. Reliab. Eng. Syst. Saf. 221 https://doi.org/10.1016/j.ress.2021.107530.

Yin, L., Lv, L., Wang, D., Qu, Y., Chen, H., Deng, W., 2023. Spectral clustering approach with K-nearest neighbor and weighted mahalanobis distance for data mining. Electronics 12. https://doi.org/10.3390/electronics12153284.

Ying, P., Ming, D., Zuo, M.J., 2010. Current status of machine prognostics in condition-based maintenance: a review. Int. J. Adv. Des. Manuf. Technol. 50, 297–313.

Zenisek, J., Holzinger, F., Affenzeller, M., 2019. Machine learning based concept drift detection for predictive maintenance. Comput. Ind. Eng. 137, 106031 https://doi.org/10.1016/j.cie.2019.106031.

Zhang, X. Da, 2020. A Matrix Algebra Approach to Artificial Intelligence. Springer, Singapore.

Zonta, T., da Costa, C.A., da Rosa Righi, R., de Lima, M.J., da Trindade, E.S., Li, G.P., 2020. Predictive maintenance in the Industry 4.0: a systematic literature review. Comput. Ind. Eng. 150, 106889 https://doi.org/10.1016/j.cie.2020.106889.