Components monitoring and intelligent diagnosis tools for Prognostic Health Management approach

Calabrese F.*, Casto A.*, Regattieri A.*, Piana F.*

** Department of Industrial Engineering (DIN), Alma Mater Studiorum – University of Bologna*

Viale del Risorgimento 2, 40136 – Bologna – Italy (francesca.calabrese3@studio.unibo.it, andrea.casto2@unibo.it, alberto.regattieri@unibo.it, francesco.piana2@unibo.it)

Abstract:

The main goal of maintenance of complex systems is to minimize downtimes to make the system as much available as possible. Condition-Based Maintenance (CBM) is one of the most effective policies used by Companies nowadays, based on the monitoring of different parameters of machines that reflect its health status. CBM can be implemented by using the Prognostic Health Management approach, made up of four main steps: data collection, signal processing, diagnostic, and prognostic. It is a proactive process that requires the development of predictive models that can trigger the alarm for corresponding maintenance. The huge amount of data that need to be collected has suggested the use of models coming from statistic theory and data mining, in order to discover regular pattern in large data sets and generate knowledge that will be useful in the maintenance decision-making process. In this paper, different intelligent methods for diagnostic purpose, such as Decision trees, K-NN algorithm, Artificial Neural Networks and support Vector Machine, are used to classify the health condition of a rotating component. Collected signals are processed in the time-domain and in the time-frequency-domain in order to extract relevant features to give as input data for the intelligent methods. Such methods are finally compared by evaluating the related accuracy value for both training and testing. The main result of this work is that the time-frequency analysis improves accuracy in classifying the health condition of machines and that new intelligent models can perform in an effective way even in the time-domain.

Keywords: maintenance; prognostic health management; signal processing; intelligent diagnostic.

1. Introduction

Safety, quality, and productivity are three aspects that any production plant has to guarantee for a sustainable competitiveness. Nowadays, industries are managing high customization markets and an increasing complexity of both products and equipment. This means short setup times, small production batches and huge changes in the industrial field. In this process of change, maintenance become a strategic element, with its own identity, able to enhance quality products, performance efficiency, environment and safety requirements, availability, and total plant cost effectiveness (Alsyouf, 2007). Products and equipment deteriorate over time, causing unexpected failures and undesirable situations. That is the rationale of maintenance (Regattieri, Manzini and Battini, 2010); its goal is to improve machinery reliability during its useful life and to keep it as much available as possible. In this way, production downtimes are minimized and activities to restore them in a proper condition are carried on when needed.

Traditional policies are mainly corrective maintenance, carried out after a failure occurs, and preventive maintenance, carried out based on runtime hours of machinery or fixed calendar days. Preventive strategies represent a major expense, due to the complexity of modern technologies (Jardine, Lin and Banjevic, 2006).

To solve the aforementioned problem, the approach called Condition Based Maintenance (CBM) is implemented, which is based on data acquisition, for example, by sensors mounted on machinery. Then, data are processed through several condition monitoring (CM) techniques, such as vibration monitoring, temperature monitoring, oil analysis, acoustic emission, thermography and others. These analyses provide useful indicators of the health status of physical assets and support the maintenance decision-making process. The implementation of an effective CBM program allows unnecessary maintenance interventions, so reducing maintenance costs. However, it is mainly a diagnostic tool (Tinga and Loendersloot, 2014). Diagnostic is a reactive process which aims to automatically detect and classify a fault after its occurrence. A proactive strategy is needed to achieve the near-zero downtimes goal: predictive maintenance and prognostic aim to monitor the degradation behaviour of machinery, to predict the occurrence of a fault and to estimate the remaining useful life (RUL) of the monitored asset.

An evolved form of CBM is the Prognostic Health Management (PHM), which provides a complete knowledge about the health status of a component or system at any point in time, and enables to predict and prevent the eventual occurrence of a fault (Lee *et al.*, 2014). The major steps in PHM are signal processing, diagnostic and prognostic. All of them are deeply investigated in literature. Several methods and algorithms, coming from statistical theory and/or artificial intelligence (AI), are available as tools for *machine learning*. This is the extraction of information and knowledge, useful in the decision-making process, by discovering regular pattern in large data sets, with a supervised or unsupervised approach.

This paper wants to provide an easy and practical methodology for the signal processing and diagnostic by using some of the most popular supervised algorithms, in order to classify the health condition of a rotating component.

The paper is organized as follows: in section 2 the PHM framework provided by (Lei, 2016) is described, with a detailed study on the signal processing techniques. Section 3 provides a brief theoretical background of some AI methods supporting the diagnostic. In section 4 a real application is presented: vibration signals are processed both in time and time-frequency domains. Then, supervised classification models are applied in MATLAB. Results are finally compared in terms of accuracy for both training and testing phases, in order to assess that in some cases, time-frequency domain analysis could improve diagnostic results.

2. Prognostic Health Management

PHM analysis uses past, present and future information of an equipment in order to assess its health, diagnose faults, predict and manage failures (Javed, Gouriveau and Zerhouni, 2017). PHM involves: prognostic, which refers to prediction, extrapolation, forecasting of process behaviour, based on current health state assessment and future operating conditions; and health management, which is defined as the decision process to intelligently perform maintenance, logistics and system configuration activities on the diagnostic/prognostic analyses. The main goals of PHM are therefore early detection and isolation of incipient faults, monitoring of their evolution and timely maintenance decisions. PHM consists in four processes: data acquisition, signal processing, or feature extraction, diagnostic; and prognostic.

Data acquisition is the process of acquiring, collecting and storing signals from machinery by different kinds of sensors and probes located on physical assets. The data acquisition system is composed by sensors, data transmission devices, and data storage devices. Many sensors are used for different measurement signals. The most common CM technique for the fault diagnosis purpose is the vibration analysis. Accelerometers are often used as sensors in this framework. Data is then transmitted to a portable device and stored into a memory location for further analysis (Javed, Gouriveau and Zerhouni, 2015).

Signals, and waveform data in general, are often characterized by noise: for this reason, they are difficult to analyse and understand directly. Signal processing techniques are needed for extracting information about the health condition of equipment. This process is also called feature extraction and is a critical element in the PHM. In literature, there are several feature extraction techniques, gathered in three main categories: time domain, frequency domain and time-frequency domain techniques. The extracted features are used as input for diagnostic, which involves:

- fault detection, to assess whether a fault has already occurred in the machine;
- fault isolation, to find the fault component and position;
- fault identification, to find the pattern and the severity of the fault.

Diagnostic is therefore defined as the process of identifying the relationship between the information obtained in the feature space and machine faults in the fault space. This analysis is also called pattern recognition. While diagnostic is only able to detect and analyse an occurred fault, prognostic deals with the fault prediction. Prognostic is combined by (Lei, 2016):

- state estimation, which quantitatively identifies the severity of the fault in order to estimate the degradation state of the machinery, based on the output of diagnostic;
- state prediction, which aims to predict the degradation trend according to the information of historical degradation curve;
- RUL prediction, which is carried out by calculating the time length of the degradation curve from current state to the final failure, based on a predetermined failure threshold.

Following, signal processing is deepen explained in paragraph 2.1, and in paragraph 3 several diagnostic models are presented.

2.1 Signal processing or feature extraction

Signal processing in the time – domain is directly based on the time waveform. One of the most popular approach is the extraction of statistical features, such as root mean square (RMS), crest factor, mean, variance, skewness, kurtosis, peak, clearance, root amplitude, shape factor and impulse factor. The RMS is a measure of the power content of the signal and is very effective especially for imbalance detection in rotating machinery (Vishwakarma *et al.*, 2017). Crest factor is used to detect changes due the impulse existing in vibration signals. It is a good indicator of incipient faults. Mean, variance, skewness and kurtosis are the first, second, third and fourth moment of probability distribution, respectively. They are used to represent the distribution of the acceleration values of the signal. In particular, kurtosis is a good indicator of incipient faults, but not of more severe faults (Lei, 2016).

Frequency – domain analysis is based on the signal transformation into the frequency domain, usually by the Fast Fourier Transform (FFT). The spectral analysis is carried out to easily identify and isolate the frequency components of interest and then reveal some information about the fault, not present in the time-domain. However, the frequency-domain analysis is not suitable for non-stationary signals, which are very common in rotating machinery. The frequency spectrum cannot reflect changes in the frequencies over time. By transforming signals into the time-frequency-domain, it is possible to obtain time-frequency distributions, which represent the energy or the power of waveform signals both in time and frequency domains. Another approach is to use time-scale representations of the signal, such as Wavelet Transforms (WT) (Jardine, Lin and Banjevic, 2006). WT can be seen as decompositions of the signal into a set of independent frequency channels: each of them carry useful information on the conditions of a specific mechanical element. They are obtained by applying filters simultaneously at different cut-off frequencies at each operation. A continuous wavelet transform is defined by

$$W(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} x(t) \psi^* \left(\frac{t-b}{a} \right) dt \qquad (1)$$

where, $x(t)$ is a waveform signal; a is the scale parameter; b is the time parameter; ψ is the wavelet function; and ψ^* is its complex conjugate. By the scale parameter dilation and the time parameter translation, the series of wavelets

with different frequencies at different times are obtained. The use of different wavelet functions leads to different results. Regardless the specific basis function, WT can present the signal with a limited number of coefficients, which can be directly used as the fault features. Different method can be used to select features from coefficients that best describe the fault (Peng and Chu, 2004).

3. Intelligent diagnosis tools for maintenance of complex systems

Pattern recognition is realized in either a supervised or unsupervised way, depending on whether the variable state (value) to predict is taken into account at the prediction moment. Thus, supervised learning methods require a pre-specified target variable and many samples with the target variable provided, to learn which values of the target variable are associated with which values of the predictor variables. Unsupervised models do not require a target variable: patterns and structures are therefore identified among all variables. Among different supervised models, classification ones are widely adopted in diagnostic. They require a categorical target variable, also called class or label, which can assume a finite number of values. A classification problem consists of defining an appropriate hypothesis space, made up of all the hypothetical relationships between the class and the predictors; then an algorithm, called classifier, will identify the function in the hypothetical space that optimally describes such relationships. First, the training set is built: it contains samples from the dataset, extracted according to an unknown probability distribution; then, the classifier is applied to derive classification rules that match the corresponding target class to each sample of the training set. Finally, remained samples, which form the test set, are classified by the classification rules. The actual target class and the predicted class are compared and the accuracy of the model is assessed. A comparison among different classification models can be made based on their accuracy and generalization capabilities. The accuracy evaluates the ability of the model to predict the target class for future observations. The model has low generalization when the accuracy of training is lower than the accuracy of testing. This may happen because the model over-fits the training set. Overfitting occurs when the model tries to take into account every possible pattern in the training set in order to achieve a higher accuracy. Hence, the complexity of the model increases until it reaches a more-than-needed level, and its generalizability on the testing set significantly decreases (Lolli *et al.*, 2017). The most common classification models are discussed below.

Classification trees are heuristic methods that classify an object into a certain number of class, defined by the user, based on its attribute values. These trees start from a root node containing all samples, and divide samples in each node into two (binary trees) or more (multi-split trees) disjoint sets (nodes), based on a predefined splitting criterion. These nodes are characterized by a higher uniformity degree of the target class than that of the parent nodes. Nodes with no outgoing edges are called leaves: observations in each leaf are labelled with the class of the most representative target value (Figure 1). Each path from the root node to a leaf is called *decision rule* and is used to classify new observations. Basically, when building a classification tree, two parameters have to be set:

- the splitting rule identifies the best target attribute among those available and selects the best splitting criterion for making descendent nodes;
- the stopping criterion is evaluated for each node to establish whether it is a leaf.

Splitting criteria can be univariate or multivariate, depending on whether the splitting is based on a single attribute value or on a function of different attribute values. Univariate criteria are often adopted: they are based on the function evaluation, called impurity, which expresses the degree of homogeneity of the observations in a node (Vercellis, 2009). Even though large decision trees are more accurate, they may lead to over-fit the data. Furthermore, a large number of leaves reduces the interpretability of the decision rule. For these reasons, an appropriate stopping criterion has to be defined. The algorithm can terminate when: (1) all instances in the training set belong to a single value of the target attribute; (2) the maximum depth of the tree has been reached; (3) the number of cases in the terminal node is less than the minimum number of cases for parent nodes; (4) if the node are split, the number of cases in one or more child nodes would be less than the minimum number of cases for child nodes; (5) the best splitting criteria is not greater than a certain threshold (Rokach & Maimon, 2015). Classification trees are non-parametric models, easy to understand and require low computational speed. However, their heuristic procedure can get stuck in local minima and may over-fit data (Khan and Yairi, 2018). In the fault diagnosis process, classification trees can also be used for the selection of the most representative features (Sugumaran, Muralidharan and Ramachandran, 2007).

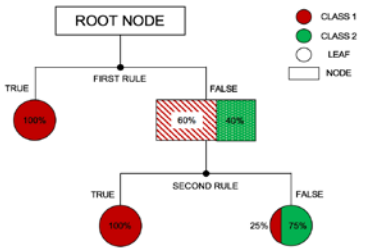

Figure 1 Binary classification tree

k-Nearest Neighbour algorithm is an instance-based learning algorithm, in which new observations are classified by comparing them with the most similar in the training set. The assumption is that input data of the same class should be closer in the features space. Therefore, for a new input data, the algorithm computes the distance between the data itself and all the observations of the training set: the new observation will be assigned to the class of its k nearest neighbours (Figure 2). Three decisions have to be taken when building a k-NN classifier: the distance metric; the number of neighbours; and the combination function for observations in the training set. The distance between two observations is a measure of their degree of similarity. Among the several distance metrics, the Euclidean distance is the most adopted. In any case, attribute values should be normalized before its computation. The number of neighbours, k, is a critical issue, because it highly

influences the classification accuracy. Large values of *k* reduce the effect of outliers, but reduce the boundaries evidence between two classes; on the other hand, small values of *k* make the algorithm particularly sensitive to the presence of anomalous values, causing the over-fitting of the data. Finally, similar samples in the training set can combine with each other in different ways to classify a new observation. The basic combination function is the so-called unweighted voting: it calls for establishing how many observations have to be involved in the classification, *k*, and comparing the new observation with its *k* nearest neighbours. An advanced combination function, is the weighted voting, which takes into account the major influence of the nearest neighbours by assigning a vote to each neighbour. The weight is inversely proportioned to the distance between neighbours and the observation to classify. k-NN shows an advantage of simple implementation. It requires less computation power during the training than other methods, but more effort during the classification process (Liu *et al.*, 2018).

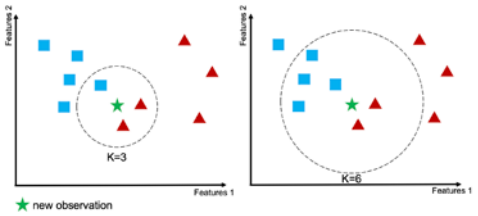

Figure 2 k-NN algorithm

Artificial neural networks (ANNs) are learning algorithms constituted by sets of nodes connected by weighted edges. The adjustment of weights is based on a pre-established rule and represents the learning process (Alsina *et al.*, 2016). Different connection modes and learning rules generate different kinds of neural networks. The most popular structure adopted in the fault diagnosis is the feedforward neural network (FFNN) (Alsina, Cabri and Regattieri, 2016) that is a layered, acyclic network, in which connections are allowed only between nodes belonging to two consecutive layers. It contains an input layer, one or more hidden layers, and an output layer, each having a certain number of nodes (Figure 3). The number of nodes in the input layer depends on the number and type of attributes of the dataset; in the output layer, it depends on the specific classification task; the number of hidden layers and the number of nodes in each of them is established by the user, who should consider that more nodes produce higher accuracy and flexibility but make the network more sensitive to the overfitting; few nodes may lead to an unacceptable training accuracy (Larose, 2005). Data inputs are collected in the input layer, which is only used for passing input values along the first hidden layer. Then, they are linearly combined with weights through a combination function, which gives a scalar value, named net:

$$net_j = \sum_i W_{i,j} x_{i,j} \qquad (2)$$

where $x_{i,j}$ is the i*th* input to node *j* and $W_{i,j}$ is the weight associated with the i*th* input to node *j*. Within each node, the net function is then passed as input to an activation function, that usually is the Sigmoid function:

$$y = \frac{1}{1 + e^{-net_j}} \qquad (3)$$

The output of the Sigmoid function in then passed to the nodes of the next level and the procedure is repeated until the activation function for the output node is computed. The obtained value is the output of the learning process, that is the predicted value for the target variable of the first sample. It is compared to the actual value and the difference between them, called error, is estimated.

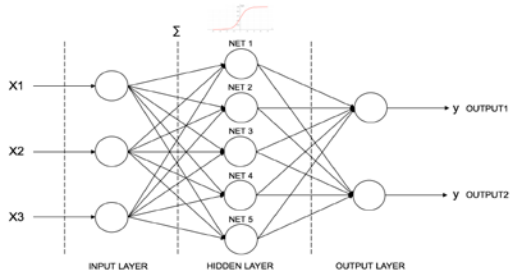

Figure 3 Feedforward neural network

Neural networks are in most cases trained by using the back-propagation algorithm. It takes the prediction error for a particular observation and filters it back to the network, assigning a portion of responsibility, δ_j, of that error to each node. Weights are adjusted in order to minimize the sum of squared errors (SSE). The new value of the weight is given by adding to the weight current value the amplitude of its adjustment, equal to:

$$\Delta w_{i,j} = x_{i,j} \delta_j \eta \qquad (4)$$

where

$$\delta_j = \begin{cases} output_j(1 - output_j)(actual_j - outpt_j) & \text{if } j \text{ is an output layer node} \\ output_j(1 - output_j) \sum_{downstream} W_{jk}\delta_j & \text{if } j \text{ is an hidden layers node} \end{cases}$$

and $\sum_{downstream} W_{jk}\delta_j$ is the weighted sum of error responsibilities for nodes downstream to node *j*. η is a constant ranging between 0 and 1, called learning rate: if small, the adjustment tends to be small and it would take more time before letting the new weight converge to the optimal value; on the other hand, if η is too large, an oscillation around the optimal value may occur. The tradeoff between convergence time and accuracy has to be found. Back-propagation algorithm terminates when: (1) the number of iterations or the total amount of time consumed by the algorithm is reached; or (2) a threshold level for the SSE on training data is set, considering that short training times may reduce the model efficacy and that ANNs are prone to overfitting. ANNs require a small number of parameters to optimize for training, but may require greater computational resources, and there is no standard to determine the network structure (Khan and Yairi, 2018).

Support Vector Machine finds its theoretical foundations in statistical learning theory (STL). It produces better results, in terms of accuracy and generalization capabilities, than any other classification models, especially when facing with large problems (Alsina *et al.*, 2018). Given a set of input features with the associated labels, SVM constructs a hyper-plane that separates the hyper-space into two or more classes, in

order to achieve the maximum separation between the classes and minimize the expected generalization error (Sugumaran, Muralidharan and Ramachandran, 2007). This process leads to the formation of two hyper-planes (bounding planes) parallel to the separating plane, and located at a certain distance (margin) from each other (Figure 4). Points on the bounding planes at the minimum distance from the separating hyperplane are the support vectors: they define the position of the separating plane generated by the classifier in the attribute space and correspond to the most representative observations for each target class (Vercellis, 2009). In case of binary classification and linearly separable data, the separating hyperplane is given by (Widodo and Yang, 2007):

$$f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b = 0 \tag{5}$$

where, $\boldsymbol{x}_i (i = 1 \dots M)$ are the input data and M is the number of samples. $\boldsymbol{w}$ (M-dimensional vector) and b (scalar) define the position of the separating plane. Supporting hyperplanes are given by

$$\boldsymbol{w}^T\boldsymbol{x} - b = 1\,, \quad \boldsymbol{w}^T\boldsymbol{x} - b = -1 \tag{6}$$

Denoting the margin with $\delta = \frac{2}{\|w\|}$, the optimization problem can be formulated as follows

$$\min_{w,b} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{M}\xi_i$$
$$\text{s.to} \begin{cases} y_i(\boldsymbol{w}^T\boldsymbol{x} - b) \geq 1 - \xi_i \\ \xi_i \geq 0 \end{cases} \quad i \in M \tag{7}$$

where C is the penalty for misclassifications and ξ_i is the distance between the margin and sample x_i which lie on the wrong side of the margin. The constraints force each point x_i to lie in the half-space corresponding to the class value y_i. The problem can be solved via Lagrangian duality. It also allows to identify the support vectors, which are associated with positive Lagrange multipliers in the optimal solution of the dual problem. Starting from the Lagrangian function of the model, the optimal solution can be found by setting the partial derivatives with respect to the variables $\boldsymbol{w}$ and b of the primal problem equal to zero. The objective of the dual problem can be obtained by replacing the obtained equality constraints into the Lagrangian function. The decision function used by SVM algorithm to classify a new observation **x** is given by

$$f(x) = sgn\left(\sum_{i,j=1}^{m} \alpha_i\, y_i \boldsymbol{x}_i \boldsymbol{x}_j + b\right) \tag{8}$$

which is not a linear function. SVM can also be used in non-linear classification tasks with application of Kernel functions, which not require explicit evaluation of the non-linear vector function that map the n-dimensional input vector onto l-dimensional feature space. There are several Kernel functions: their choice is very important as long as they define the feature space in which the training set samples will be classified. SVM has generalization capabilities, also with few training data and data from two or more categories can always be separated by a hyperplane. Thus it can produce high accuracy in classification tasks (Liu *et al.*, 2018). However, there are no standards for choosing the kernel function**.**

4. Case study

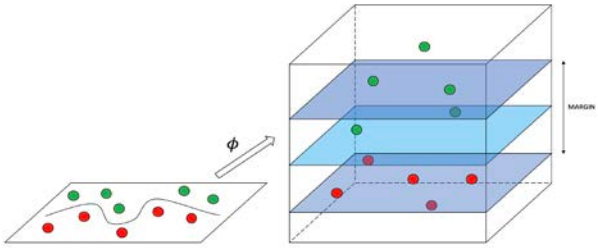

Figure 4 Separating plane and boundaries planes for a linearly separable dataset

In this section, signal processing and diagnostic, as suggested by the PHM paradigm, are applied to a rotating machine. First, a brief description of the data collection step is provided. Then, the collected signals are processed in the time-domain and in the time-frequency-domain in order to extract relevant features to give as input for the intelligent methods introduced in the previous section. They are finally compared by evaluating the accuracy value for both training and testing. The accuracy is computed as the ratio between the values correctly predicted for the target variable and the total of predictions.

Data acquisition. Vibration data are collected by using a triaxial accelerometer mounted on a mechatronic actuator of a rotating equipment. Acceleration measurements are taken into three directions, each connected to a channel of a digitizer. Samples are registered every 3,9063E^{-5} seconds, and the sampling frequency is equal to 25600 Hz. The vibration data are taken for the component under health condition and damaged condition of the actuator. For each condition, three runs are conducted, lasting 117 seconds each. For the analysis, two different run modes characterized by different vibration signatures are chosen: in particular, mode 2 shows more anomalous peaks than mode 1. This situation can strongly influence diagnostic step. Acceleration values are computed as follows:

$$a = \sqrt{a_x^2 + a_y^2 + a_z^2} \tag{9}$$

where a_x, a_y, a_z are the acceleration measured along the three axes of the accelerometer. Each run is divided into bins containing 25,600 input data. Therefore, 351 signal segments are obtained for each machine condition and for each mode.

Time – domain analysis. For each signal segment, 10 time-domain features are extracted: root mean square (RMS), variance, skewness, kurtosis, peak, crest factor, root amplitude, clearance, shape factor and impulse factor. Then, their values are normalized in the range of 0/1, except for skewness, which is normalized in the range of -1/-1, by dividing each feature for the corresponding maximum value. Normalized feature dataset, together with the target vector indicating the condition to which each observation belongs, is given as input of the four classification models. The training set is built by using the first run of each mode. Hence, it contains 234 samples, 117 for each condition. Testing is performed on the other two sets, separately. Decision tree is trained by using the Gini-index as splitting rule. k-NN algorithm is trained by using the Euclidean distance and different values of k.

Best accuracy values are provided by k equal to 2 for both modes. A FFNN using the backpropagation algorithm is trained by making the number of hidden layers vary from 1 to 2, the number of neurons in the first hidden layer from 1 to 10, and the number of neurons in the second hidden layer from 1 to 5. The minimum gradient of $10E^{-10}$ and the maximum number of iterations equal to 500 are set as termination criteria. Best accuracy values are given by a neural network composed by 1 only hidden layer with 6 nodes. Finally, Support Vector Machine is trained by adopting a linear kernel function.

As shown in Table 1, which summarizes accuracy values in % for the training set and for the two testing sets, results of different models are similar for mode 1. For mode 2, the decision tree and the k-NN algorithm seem to give higher accuracy values than other models. However, looking at testing results of both modes, the decision tree and ANN give lower accuracy values. This means that they are less generalizable than other two models. In addition, for mode 2, k-NN clearly over-fit the training set. At the contrary, the SVM is the model with the best trade-off between the accuracy and the generalization, in both domains.

Table 1 Model accuracy in the time-domain

Model	Mode 1			Mode 2		
	Training	Testing 2	Testing 3	Training	Testing 2	Testing 3
DT	95,71	72, 65	94,06	95,71	75,93	73,15
K-NN	95,71	93,36	97,90	90,00	74,53	72,34
ANN	95,71	77,89	86,55	81,43	60,31	56,87
SVM	97,90	98,43	97,60	82,50	83,74	85,30

Time – frequency – domain analysis. Each signal segment is transformed into the time-frequency-domain by using the Morlet wavelet transform, which has a band-pass characteristic and it is a good detector of high-frequencies. The following features are directly extracted from coefficients of the transform:

$$f_1 = \min(C_i(t)) \qquad f_2 = \max(C_i(t))$$
$$f_3 = \sqrt{E[(C_i(t) - C_m)^2]} \qquad f_4 = f_2 - f_1 \tag{10}$$

where, $C_i(t)$ are the coefficients generated by the Morlet transform, and C_m is their average value. As in the time-domain analysis, the features dataset, together with the target vector indicating the condition to which each observation belongs, is given as input of the four classification models vector. Diagnostic is conducted in the same manner as in the time-domain analysis. The differences are in the parameters of k-NN and ANN. In particular, for k-NN algorithm, the best value k is equal to 5 for both modes; for ANNs, best training results are given by a structure with 1 hidden layer containing 4 nodes for mode 1 and 5 nodes for mode 2.

Results of both training and testing are summarized in Table 2. For mode 1, models provide similar accuracy values, and no model seems to over-fit data. For mode 2, the same consideration can be done, except for ANN, which still over-fit the data.

Table 2 Model accuracy in the time-frequency-domain

Model	Mode 1			Mode 2		
	Training	Testing 2	Testing 3	Training	Testing 2	Testing 3
DT	95,71	97,60	98,04	94,29	90,17	91,79
K-NN	97,14	97,65	97,65	91,43	91,86	91,79
ANN	95,71	96,87	98,04	95,71	90,66	90,93
SVM	98,30	97,65	97,65	88,20	93,51	92,88

In general, results obtained in the time-frequency-domain are better than those in the time-domain, especially from the generalization point of view. In addition, accuracy values of mode 2 are always lower than those of mode 1. The reason could be the presence of anomalous peaks in the vibration signature of mode 2. In particular, the z-axis seems to give some problems, which are not related to the health condition of the component. Hence, a further analysis is conducted considering the value of the acceleration without the component along the z-axis. Results are shown in Table 3. Results provided by the decision tree are slightly better in the time-domain, but much worse in the time-frequency-domain. For k-NN the situation is the opposite. Instead, results obtained with ANN are similar in the time-frequency-domain, but much better in the time-domain, especially from the generalization point of view. SVM provides similar or slightly better results.

Table 3 Model accuracy of mode 2 after signal preprocessing

Model	Time-domain			Time-frequency-domain		
	Training	Testing 2	Testing 3	Training	Testing 2	Testing 3
DT	98,57	79,99	74,76	92,86	50,78	51,64
K-NN	91,43	62,65	63,43	94,29	91,79	92,18
ANN	82,86	90,23	87,81	95,71	89,84	90,54
SVM	82,10	86,71	84,99	89,90	96,32	92,57

The application of diagnostic and signal processing results in quite good accuracy values, which vary according to both the analysis domain and the specific model. All models work better when signals are transformed into the time-frequency-domain. In this particular case, wavelet transform is used as to reduce the noise, and thus effects of anomalous values in the signals. The second experiment on mode 2 highlights that the manual cleaning of signals in this domain does not improve the accuracy. However, it would be preferable for some models, in particular ANNs, in the time-domain analysis, in which the distribution of the acceleration values influence by far the classification results.

5. Conclusions

PHM offers a systematic approach to detect and analyse faults and to predict the RUL of components. This paper is focused on feature extraction and diagnostic, aimed to identify indicators of the health status of machinery and to build models which are able to classify the condition of machinery after learning from such indicators. Different signal processing techniques and different AI models are applied in a supervised manner to a rotating component, for which vibration signals under both good and failed

conditions are available. Results show that such models perform well and confirms many of the results that can be seen in literature. However, one issue emerged: which and how many features are needed to be extracted? In addition, the data regarding the health conditions are not always available. This means that classification models cannot be used. A particular class of AI algorithms is used to deal with these problems: they are referred as to deep learning, whose aim is the automatic extraction of discriminative features, by using an unsupervised learning. Deep learning has been recently adopted in the fault diagnosis field and it is a very promising tool, that is worth to deepen.

References

Alsina, E. F. *et al.* (2016) 'Artificial neural network optimisation for monthly average daily global solar radiation prediction', *Energy Conversion and Management*, 120, pp. 320–329. doi: https://doi.org/10.1016/j.enconman.2016.04.101.

Alsina, E. F. *et al.* (2018) 'On the use of machine learning methods to predict component reliability from data-driven industrial case studies', *The International Journal of Advanced Manufacturing Technology*, 94(5), pp. 2419–2433. doi: 10.1007/s00170-017-1039-x.

Alsina, E. F., Cabri, G. and Regattieri, A. (2016) 'A Neural Network Approach to Find The Cumulative Failure Distribution: Modeling and Experimental Evidence', *Quality and Reliability Engineering International*, 32(2), pp. 567–579. doi: 10.1002/qre.1773.

Alsyouf, I. (2007) 'The role of maintenance in improving companies' productivity and profitability', *International Journal of Production Economics*, 105(1), pp. 70–78. doi: 10.1016/j.ijpe.2004.06.057.

Jardine, A. K. S., Lin, D. and Banjevic, D. (2006) 'A review on machinery diagnostic and prognostic implementing condition-based maintenance', *Mechanical Systems and Signal Processing*, 20(7), pp. 1483–1510. doi: 10.1016/j.ymssp.2005.09.012.

Javed, K., Gouriveau, R. and Zerhouni, N. (2015) 'A New Multivariate Approach for Prognostic Based on Extreme Learning Machine and Fuzzy Clustering', *IEEE Transactions on Cybernetics*, 45(12), pp. 2626–2639. doi: 10.1109/TCYB.2014.2378056.

Javed, K., Gouriveau, R. and Zerhouni, N. (2017) 'State of the art and taxonomy of prognostic approaches, trends of prognostic applications and open issues towards maturity at different technology readiness levels', *Mechanical Systems and Signal Processing*, 94, pp. 214–236. doi: 10.1016/j.ymssp.2017.01.050.

Khan, S. and Yairi, T. (2018) 'A review on the application of deep learning in system health management', *Mechanical Systems and Signal Processing*. Elsevier Ltd, 107, pp. 241–265. doi: 10.1016/j.ymssp.2017.11.024.

Larose, D. T. (2005) *Discovering Knowledge in Data: An Introduction to Data Mining*. Edited by J. W. & Sons. doi: 10.1002/0471687545.

Lee, J. *et al.* (2014) 'Prognostic and health management design for rotary machinery systems - Reviews, methodology and applications', *Mechanical Systems and Signal Processing*. Elsevier, 42(1–2), pp. 314–334. doi: 10.1016/j.ymssp.2013.06.004.

Lei, Y. (2016) *Intelligent Fault Diagnosis and Remaining Useful Life Prediction of Rotating Machinery*. Elsevier Science. Available at: https://books.google.it/books?id=fTJQDAAAQBAJ.

Liu, R. *et al.* (2018) 'Artificial intelligence for fault diagnosis of rotating machinery: A review', *Mechanical Systems and Signal Processing*. Elsevier Ltd, 108, pp. 33–47. doi: 10.1016/j.ymssp.2018.02.016.

Lolli, F. *et al.* (2017) 'Single-hidden layer neural networks for forecasting intermittent demand', *International Journal of Production Economics*, 183, pp. 116–128. doi: https://doi.org/10.1016/j.ijpe.2016.10.021.

Peng, Z. K. and Chu, F. L. (2004) 'Application of the wavelet transform in machine condition monitoring and fault diagnostic: A review with bibliography', *Mechanical Systems and Signal Processing*, 18(2), pp. 199–221. doi: 10.1016/S0888-3270(03)00075-X.

Regattieri, A., Manzini, R. and Battini, D. (2010) 'Estimating reliability characteristics in the presence of censored data: A case study in a light commercial vehicle manufacturing system', *Reliability Engineering & System Safety*, 95(10), pp. 1093–1102. doi: https://doi.org/10.1016/j.ress.2010.05.001.

Sugumaran, V., Muralidharan, V. and Ramachandran, K. I. (2007) 'Feature selection using Decision Tree and classification through Proximal Support Vector Machine for fault diagnostic of roller bearing', *Mechanical Systems and Signal Processing*, 21(2), pp. 930–942. doi: 10.1016/j.ymssp.2006.05.004.

Tinga, T. and Loendersloot, R. (2014) 'Aligning PHM, SHM and CBM by understanding the physical system failure behaviour', *European Conference of the Prognostic and Health Management Society*, pp. 1–10. Available at: http://doc.utwente.nl/92058/1/PHME_1263_final.pdf.

Vercellis, C. (2009) *Business intelligence: data mining and optimization for decision making*. Edited by John Wiley & Sons.

Vishwakarma, M. *et al.* (2017) 'Vibration Analysis & Condition Monitoring for Rotating Machines: A Review', *Materials Today: Proceedings*, 4(2), pp. 2659–2664. doi: 10.1016/j.matpr.2017.02.140.

Widodo, A. and Yang, B. S. (2007) 'Support vector machine in machine condition monitoring and fault diagnosis', *Mechanical Systems and Signal Processing*, 21(6), pp. 2560–2574. doi: 10.1016/j.ymssp.2006.12.007.