Streaming-based feature extraction and clustering for condition detection in dynamic environments: an industrial case

Francesca Calabrese[1], Alberto Regattieri[2], Francesco Pilati[3] and Marco Bortolini[4],

[1,2,4]*Department of Industrial Engineering (DIN), University of Bologna, Bologna, 40136, Italy*
francesca.calabrese9@unibo.it
alberto.regattieri@unibo.it
marco.bortolini3@unibo.it

[3]*Department of Industrial Engineering, University of Trento, 38123 Povo, Italy*
francesco.pilati@unitn.it

ABSTRACT

As relevant aspects of PHM, feature extraction and diagnostics represent the focus of many paper related to predictive maintenance. Feature extraction is a fundamental part in PHM, as the accuracy of diagnostic models strongly depends on the goodness of the features. In particular, it is performed when components or systems are provided with many sensors, in order to extract relevant and non-redundant information from the raw dataset. Diagnostic is a typical pattern recognition problem, in which the data are classified according to the operating or fault condition they refer. Thus, to obtain the Remaining Useful Life (RUL) of the component, usually an offline analysis, including feature extraction and diagnostics, is performed, whose results are then used for degradation modelling and finally RUL prediction. However, when systems do not operate in strictly controlled environments, as in case of industrial contexts, it is very challenging to obtain information about the operating condition at the moment of signal acquisition. This results in unlabeled datasets, which cannot be used by supervised learning algorithms, as ANNs and SVMs. In addition, operating conditions change over time and it is not always possible to know a priori all possible conditions. These considerations suggest to resort to streaming applications, in which models can directly learn from new incoming data. As the degradation rate may vary according to the operating condition, influencing the RUL prediction, one should always know in which condition the machinery is operating, or should recognize if a new condition is occurring. In addition, it is not possible to extract good features that distinguish different conditions, if a condition is not known. Therefore, features should be extracted in an unsupervised manner and incrementally, so that if a new condition occurs, eventually better features can be extracted. Furthermore, an incremental clustering should be conducted so to always recognize the condition under which the system is operating, if known, or to detect a new condition.
In this paper, a streaming-based procedure for feature extraction and clustering is proposed, which is validated on a real industrial case study. A batch and supervised feature extraction and diagnostics are also performed on the same dataset, to demonstrate that the two approaches have similar results, in terms of accuracy with respect to the known conditions. In addition, thanks to the incremental clustering, the proposed approach is also able to detect and automatically label new machinery operating conditions.

1. INTRODUCTION

Maintenance of equipment in manufacturing companies is assuming a more and more crucial role in the minimization of the life-cycle cost of their systems. Contrary to the past, the role of maintenance is not only to repair failed assets, but also to achieve the optimum availability, the optimum operating conditions, the maximum utilization of resources, the optimum equipment life, the minimum spares inventory and last but not least the ability to react quickly (Mobley, 2002). For more critical components, preventive strategies, like Condition-Based Maintenance (CBM) could be replaced by Predictive Maintenance (PM) strategies, that not only aim to isolate and identify a certain failure, as CBM, but also aim to know when a system will fail in advance, so to plan maintenance interventions and spare parts supplying with sufficient time (Nguyen & Medjaher, 2019). Thus, PM can be viewed as an evolution of CBM, that adds to the typical activity of CBM, i.e., data acquisition, feature extraction and diagnostic, also the prognostic task, whose aim is to predict the Remaining Useful Life (RUL) of monitored components (Jardine, Lin & Banjevic, 2006).

As one of the pillars of the fourth industrial revolution, PM also experiences the adoption of technologies from different domains, including Internet of Things (IoT), Edge Computing and Cloud Computing (Katona & Panfilov, 2018). The Industrial Internet of Things (IIoT) let each asset in the industrial plant be connected and communicate with other assets. Sensors installed on equipment can send the data to a cloud, where Machine Learning (ML)-based software are run in order to provide information about the health status of connected equipment. Basically, Cloud Computing, as a centralized service provider, makes possible to gather in a unique "place" data from all sensors spread in the industrial plant, enabling the sharing of information among the assets and the construction of an integrated knowledge of the entire plant. However, the high number of sensors increases the amount of the generated data, the transmission bandwidths and consumed power, the storage spaces and computation resources of cloud services (Qian, Lu, Pan, Tang, Liu & Wang, 2019). In addition, the data transmission from devices to the cloud put not trivial challenges from the privacy point a view. These reasons make attractive to perform data storage and processing directly at the edge of the network. Edge Computing allows to reduce the peaks in traffic flows, the bandwidth requirements of the centralized network and the transmission latency during data computing and storage, enabling real-time collection and analysis of the information and providing short upload time of massive data (Yu, Liang, He, Hatcher, Lu, Lin & Yang, 2018). However, because of the limited storage capacity of edge devices, several edge nodes will be used and coordinated for storing data, increasing the complexity of the data management. The integration of Cloud and Edge computing in a structure that locally processes high-priority tasks and delay-sensitive tasks while processes low-priority and delay-tolerant tasks in the cloud could represent the optimal infrastructure for PM applications (Calabrese, Regattieri, Botti & Galizia, 2019), (Angelopoulos, Michailidis, Nomikos, Trakadas, Hatziefremidis, Voliotis, & Zahariadis, 2020).

Prognostic Health Management (PHM) is a step-wise process for the realization of PM. Based on this approach, signals are first collected from critical components, in each possible operating and fault conditions; then, relevant and redundant features the best distinguish the health condition from the faulty ones and a monotonic Health Indicator (HI) that reflect the degradation process are computed for diagnostic and prognostic purposes, respectively; then, diagnostics is conducted to find the relationships between the feature space and the health conditions, so to be able to classify next observations; finally, prognostics is conducted based on the extracted HI, in order to predict the RUL of the component (Lei, Li, Guo, Li, Yan & Lin, 2018).

Once diagnostics and prognostics models had been trained on historical data, they can be applied to streaming data at the edge, taking the advantage of cloud services when necessary. In this way, it is possible to obtain a real-time feedback on the health condition of the monitored equipment and react as fast as possible. Examples of edge-cloud structures for streaming-based PM can be found in the works by Bowden Bowden, Marguglio, Morabito, Napione, Panicucci, Nikolakis, Makris, Coppo, Andolina, Macii, Becker & Jung (2019), Yaseen, Swathi & Kumar, (2018), Bose, Kar, Roy, Gopalakrishnan & Basu (2019), by Qian et al., (2019). A streaming-based analysis, can also help solve two main issues related to the application of PHM in industrial contexts. First, labeled data are not always available. While it is quite easy to collect data corresponding to healthy conditions, it is difficult to get data during faulty conditions, as failures can be very rare or, for safety reasons, it is not possible to simulate a failures. As a consequence, the labeled dataset may be unbalanced, reducing the performance of diagnostic models. Second, external factors strongly affect the machinery functioning. Among them, you can find not only environmental conditions, like external temperature, humidity level or surrounding equipment vibration levels, but also the machinery operating conditions. Indeed, machinery can work under different settings, because of the difference in the processed material or in the production parameters. Also these conditions, as the environmental ones, cannot be known a priori and can evolve in time, making pre-trained models obsolete. In other words, industrial plants are inherently dynamic, which limits the simply streaming application of a pre-trained PHM approach even to the same component if it works in different places.

To deal with these issues, unsupervised or semi-supervised and incremental learning can be introduced in a streaming analysis, in order to directly process unlabeled data and let models learn by themselves as a new observation is available. To this purpose, novelty detection algorithms assume a crucial role. Their aim, in a streaming analysis, is to detect a change in the machinery behavior and identify if the current condition is known or unknown. Based on the novelty detection algorithm results, then the activity to carry on for diagnostic and prognostic will be different.

In this paper, a methodology for streaming-based PHM application is presented. In particular, the attention has been focused on the operating condition recognition problem, which requires an incremental feature extraction and an incremental unsupervised diagnostic model. Here, the Incremental Principal Component Analysis introduced by Lippi & Ceccarelli (2019), the Anomaly Detection presented by Costa, Angelov & Guedes (2015) and the Incremental Clustering presented by Gu, Angelov & Príncipe (2018) have been slightly modified and integrated in a unique framework able to select the most relevant features on-line, to detect anomalous behaviors and to assign each observation to an existing or a new cluster (operating condition). The novelty of the methodology, with respect to other existing frameworks in literature, is that it can start "from scratch" and no previous batch and supervised analysis is needed. This methodology is suitable for components that experiment a very slow degradation that depends on the machinery setting.

The remaining of the paper is organized as follows. In section 2, the problem of operating condition recognition is defined, and both characteristics and requirements for tackling it in real industrial contexts are presented. In section 3, previous works related to streaming and semi-supervised analysis for fault/condition detection and diagnosis are reviewed and the mathematical formulation of the chosen algorithms is reported. In section 4, the proposed methodology is described, and the pseudo-code is also provided. In section 5, the proposed methodology is applied to a real industrial case, in order to verify its goodness in recognizing different operating conditions. Finally, in section 6, conclusions and directions of future research are provided.

2. RELATED WORKS AND PROBLEM SETTING

The ultimate goal of PHM is to predict the RUL of a component/system, in order to anticipate the occurrence of a failure, while maximizing its useful life. This goal is achieved by computing a Health Indicator (HI) from collected signals and building a degradation model that follows the degradation trend. Given the degradation model, at any point in time the RUL can be computed as the difference between the time in which the HI is expected to reach a prefixed FT and the current time. The essence of degradation modelling is to develop a "good" probability model that is able to describe the degradation phenomenon (Ye & Xie, 2015). In the data-driven approach, it can be done by resorting to statistical or stochastic models, whose parameters are estimated based on historical failures data, in order to minimize the Maximum Likelihood Estimation (MLE) (Ye, Wang, Member & Tsui, 2013), (Xia, Dong, Xiao, Du, Pan & Xi, 2018), (Sikorska, Hodkiewicz & Ma, 2011), (Si, Wang, Hu & Zhou, 2011). Adaptive degradation models have also attracted many researches in this filed, as they can adapt their parameters also based on actual data, instead of relying only on historical data (Zhai & Ye, 2017), (Datong, Yu & Xiyuan, 2011).

Machinery operating conditions (e.g., a different load, or a different set of temperatures) affect the degradation phenomenon. Operating conditions may change depending on the machinery user, on the material that has to be processed, on the environmental conditions of the industrial plant and on other factors. Thus, the degradation models should include a different degradation rate that varies according to the operating condition (Moghaddass & Zuo, 2014). For example, Bian, Gebraeel & Kharoufeh, (2015), assume the operating conditions to evolve as a continuous-time Markov chain and the resulting degradation model parameters are updated in real-time within the Bayesian framework. Although that approach is highly promising, it still requires the total number of settings to be known a priori. In addition, in many cases, a subset of the acquired signals could define the particular condition that is implemented. A change in one or more variable values included in the subset determines a change in the machinery setting. In these cases, if the operating condition is known, then the corresponding degradation model can be selected. So, basically, the problem of updating degradation models based on the operating condition, could be reduced to a condition operating recognition problem, which is similar to diagnostics or fault detection and identification problems, where fault conditions are replaced by multiple normal operating conditions.

As a pattern recognition problem, it can be faced by means of supervised ML models, which are trained on historical data related to different conditions to derive rules to classify new unlabeled observations (Liu, Yang, Zio & Chen, 2018), (Benkedjouh, Medjaher, Zerhouni, Rechak, 2013). Although these models work effectively in recognizing two or more conditions, supervised models require to know a priori all possible operating conditions. However, the same issues as in the diagnostic task, i.e., the lack of labeled data and dynamic nature of the industrial environments, also apply to the operating condition recognition problem. Thus, unsupervised, or semi-unsupervised and incremental learning may be adopted in order to process unlabeled data and create a new cluster each time a new operating condition occurs (Hu, Zhu, Cheng, He, Yan, Song, & Zhang, 2017).

As in the supervised learning, the unsupervised learning also depend on the extracted features. In general, there are two main issues related to the choice of the most suitable features. The first one is connected to the frequency of the data acquisition (sampling). The second one is connected to the number of the collected signals (dimensionality reduction). Indeed, signals are usually collected at high frequencies, in the order of kilohertz, so to get as much information as possible. However, signals collected at high frequencies present a very oscillating trend, which may compromise the classification/clustering task. For this reason, a sampling activity, usually referred as feature extraction, is often conducted (Lee, Wu, Zhao, Ghaffari, Liao & Siegel, 2014). In this step, statistics of the signals in the time or frequency domain are computed over a time window, so to greatly reduce the number of data samples. Alternatively, the signal is transformed in the time-frequency domain, and statistics, as well as energy information of the signals, are extracted in this domain. However, this activity results in an increased number of variables, which may be or may not be relevant for the classification/clustering problem. In addition, very often machinery are provided with a high number of sensors, from which several signals can be collected. For each added signal, the number of extracted statistics doubles down, resulting often in a very high-dimensional dataset. For this reason, a second step, named feature selection or feature extraction is performed in order to transform the high-dimensional dataset in a lower dimensional dataset, by automatically selecting or extracting only relevant and non-redundant features (Hu, Baraldi, Di, & Zio, 2017), (Wang, Li, Jiang & Cheng, 2017). In our problem, even if signals representing the operating condition are known, the different trends they assume in each condition is unknown. In particular, besides the amplitude of the signal, the number of signals that change also affects the operating condition. Thus, all signals should be monitored simultaneously, increasing the complexity of the algorithm and limiting their processing in streaming. A synthetic feature (computed for instance as the mean of all the signal values) may be not appropriate as the signals may assume very similar values among each other and the synthetic value may result to be the same for two different settings. To clarify this concept, an example of signal trends in different operating conditions is shown in Fig. 1. Here, the mean of the three signals is similar for setting 2 and 3.

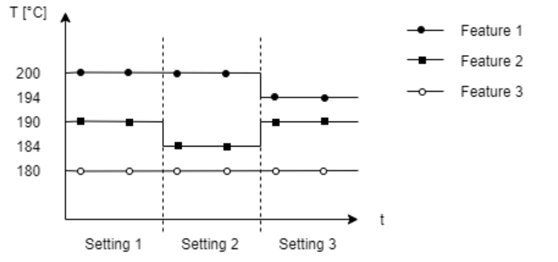

Figure 1. Example of a change in the signal trend corresponding to a change in the operating condition

Thus, an incremental and unsupervised feature extraction method, able to extract the best feature when required, is needed.

Although on-line, semi-supervised and incremental techniques in non-stationery or dynamic environments have attracted many researchers in last years, only a few works are related to predictive maintenance. These works are mainly based on the integration of novelty detection and diagnosis models in order to discover new scenarios, starting from data related to a healthy condition and some known fault conditions. In the work of Cariño, Delgado-Prieto, Iglesias, Sanchis, Zurita, Millan, Ortega Redondo, & Romero-Troncoso (2018), an ensemble-based classifier for novelty detection and an evolving classifier for diagnosis are separately performed to a different set of features, and then integrated in a unique methodology, in order to discover new patterns in case only data related to a healthy condition are available. In particular, statistical features in the time domain are extracted from raw signals and dimensionality reduction techniques are applied only for the novelty detection part. In this work, a measurement considered unknown can represent an outlier, a new fault or a new operating condition. Thus, the intervention of a user is required to verify which of the above cases the novelty refers to. In the work of Cariño, Delgado-Prieto, Zurita, Picot, Ortega, & Romero-Troncoso (2020), an hybrid approach is presented for multi-modal signal analysis, novelty detection and diagnosis, whose aim is to detect and incrementally include new discovered scenarios, based on time-frequency features. Both papers operate in a Semi-Supervised Learning context, which basically uses limited labeled data to transfer their class information to unlabeled data. Instead, Dyer, Capo & Polikar (2014) defined a new scenario, named Initially Labeled Streaming Environment (ISLE), which is characterized by an infinite verification latency, i.e., no labeled data are ever received after initialization. In this context, they developed a new algorithm, named COMPOSE, which learns drifting concepts from a streaming non-stationary environment that provides only unlabeled data after initialization. COMPOSE has been applied to fault diagnostics by Hu, Baraldi, Di Maio & Zio (2017), in which a semi-supervised feature selection step is also presented. Based on the resulting framework, both gradual and abrupt changes can be detected; then, a classifier is updated in order to include a new class and a feature set is selected for the new class. However, this framework still requires a feature set extracted off-line and a classifier to be trained on labeled data. Thus, it is not appropriate in cases in which no labeled data is available.

In conclusion, in case of a set of signals directly correlated to the operating condition (we call them characteristic signals from now onward) is available and collected at high frequency, and in case of a streaming analysis performed in an edge-cloud infrastructure, the characteristics and requirements of the condition recognition problem can be summed up as follows:

1. As the characteristic signals slightly oscillate around a certain mean value that correspond to the set point of that signal, a high frequency is not needed. Thus, the mean of each signal, directly in the time domain, can be computed over a certain time window, in order to reduce the amount of data to process and store.
2. A set of relevant features has to be extracted from the sampled characteristic signals. As it operates in an unsupervised and dynamic environment, the algorithm for feature extraction has to be incremental. In particular, if the condition is known, the set of relevant features is known as well. Thus, the algorithm can directly extract them. When the condition is unknown, the set of relevant features is unknown too. Thus, the algorithm should evaluate which features are relevant.
3. Based on the extracted features, a novelty detection method has needed in order to detect if a change in the original characteristic signals has occurred. When the behavior of the system changes, the algorithm has to recognize whether the condition is known or unknown, and label the observations accordingly. In particular, in the first case, the current observation is assigned to an existing cluster (that is, to one of the existing operating condition). In the second case, a new cluster should be created, and all similar observations are assigned to the same unknown cluster.
4. Finally, a validation is needed. It may be performed into the cloud each time a new cluster is created. In particular, when a new condition is detected, the corresponding sampled characteristic signals can be temporary stored into the edge and then sent to the cloud. Here, a batch analysis can be conducted in order to extract relevant

features and initialize the corresponding cluster, so that the corresponding operating condition from that moment will be considered known.

Here, an incremental Principal Component Analysis (PCA) has been chosen for feature extraction. PCs are extracted each time a new data is available and PCs retaining the 90% of variance are selected each time a new condition is detected. Novelty detection is performed at each observation and a completely unsupervised and incremental clustering algorithm is adopted for assigning data points to existing or new conditions.

3. THEORETICAL BACKGROUND

In next subsections, the theoretical backgrounds of the adopted algorithms for incremental feature extraction, anomaly detection and incremental clustering is provided.

3.1. Incremental PCA

Lippi & Ceccarelli (2019) presented an exact incremental implementation of PCA. As the authors state, exacts means that it provides the same results, i.e., the same PCs as in the batch version. In addition, it also contains an online data normalization, which is fundamental when variables assume very different values. Basically, the difference between the batch PCA and its incremental version formulated in that paper lies in the covariance matrix computation, which is recursive. The steps of the algorithm are the following:

1. The sample mean $\bar{x}_{n(j)}$and the standard deviation $\bar{\sigma}_{n(j)}$ are computed for each variable j $(j = 1, \dots, m)$ over the first n available observations, in order to compute the standardized matrix Z_n as follows

$$Z_n = \begin{bmatrix} x_1 - \bar{x}_n \\ \dots \\ x_n - \bar{x}_n \end{bmatrix} \Sigma_n^{-1} \qquad (1)$$

   Where $\Sigma_n \equiv diag(\sigma_n)$ is an $m \times m$ matrix

2. The covariance matrix Q_n of the data matrix X_n is computed as follows

$$Q_n = \frac{1}{n-1} Z_n^T Z_n \qquad (2)$$

3. The standard diagonalization of Q_n is made by means of the eigenvector matrix C_n as follows

$$Q_n = C_n^{-1} \begin{bmatrix} \lambda_1 & & \\ & \dots & \\ & & \lambda_m \end{bmatrix} C_n \qquad (3)$$

   Where the eigenvalues λ_i are put in descending order and express the variance associated with the ith Principal Component (PC), that is the ith eigenvector of C_n.

4. Finally, the time evolution of PC values until the time stamp n is computed as

$$PC_n = Z_n C_n \qquad (4)$$

5. At the step $n + 1$, the mean and the standard deviation are updated and the standardized matrix Z_{n+1} is computed as follows

$$Z_{n+1} = \begin{bmatrix} Z_n \Sigma_n + \Delta \\ y \end{bmatrix} \Sigma_{n+1}^{-1} \qquad (5)$$

   Where $y = x_{n+1} - \bar{x}_{n+1}$, Δ is a $n \times m$ matrix made of repeating n times the vector $\delta = \bar{x}_n - \bar{x}_{n+1}$

6. The covariance matrix nQ_{n+1} is computed ad follows

$$nQ_{n+1} = Z_{n+1}^T Z_{n+1} \qquad (6)$$

   Which only depends on the covariance matrix computed at the point n and the new feature vector $\boldsymbol{x}_{n+1}$.

7. Finally, the updated Q_s are used to compute the nth values for the evolving PCs by means of Eq. (4).

3.2. Anomaly Detection

Angelov, Ramezani, & Zhou (2008) introduced for the first time the concept of Recursive Density Estimation (RDE) in the context of detection and object tracking in video streams. Aiming to decide whether a pixel belongs to the background or the foreground in real-time, the introduced approach basically substitutes the traditional Gaussian kernel adopted for modelling the pixel probability density function (pdf) with the Cauchy function, which allows the recursive estimation of pixel pdf as a new image frame occurs. In Costa et al., (2015), concepts of RDE theory are adopted in the context of online fault detection, in order to discover anomalous behaviors. Basically, the parameters involved are the global density, the mean value and the scalar product, which can be recursively computed by Eq. (7), Eq. (8) and Eq. (9), respectively.

$$D(\boldsymbol{x}_k) = \frac{1}{1 + \|\boldsymbol{x}_k - \boldsymbol{\mu}_k\|^2 + \Sigma_k - \|\boldsymbol{\mu}_k\|^2} \qquad (7)$$

$$\boldsymbol{\mu}_k = \frac{k-1}{k} \boldsymbol{\mu}_k + \frac{1}{k} \boldsymbol{x}_k \qquad (8)$$

$$\Sigma_k = \frac{k-1}{k} \Sigma_k + \frac{1}{k} \|\boldsymbol{x}_k\|^2 \qquad (9)$$

where $\boldsymbol{x}_k \in \mathbb{R}^n$ is the feature vector at the time stamp k.
At the first iteration ($k = 1$), the parameters are initialized as $D(\boldsymbol{x}_1) = 1$, $\boldsymbol{\mu}_1 = \boldsymbol{x}_1$, $\Sigma_1 = \|\boldsymbol{x}_1\|^2$. The, for each $k > 1$, the parameters are updated and the condition in Eq. (10) is checked to decide whether the current point represents an anomaly or not

$$\begin{gathered} IF\ D(\boldsymbol{x}_k) < \mu_D \\ for\ k = t_1, \dots, k-1, k \\ THEN\ \boldsymbol{x}_k\ is\ an\ anomaly \end{gathered} \qquad (10)$$

Where

$$\mu_D = \left(\frac{ks-1}{ks}\mu_D + \frac{1}{ks}D(\boldsymbol{x}_k)\right)(1 - \Delta_D) + D(\boldsymbol{x}_k)\Delta_D \quad (11)$$

is the mean value of the local density computed recursively, $\Delta_D = |D(\boldsymbol{x}_k) - D(\boldsymbol{x}_{k-1})|$ is the absolute value of difference between the global density computed at two consecutive time stamps, and ks is the number of data samples from the last status change. Indeed, if the condition is satisfied, then the status of the system switches from normal to anomalous. and ks is set to 0. The condition expressed in Eq. (10) basically means that if the global density is lower than the mean density for a certain number of time stamps (or seconds), then the status becomes anomalous. Indeed, when a new data point arrives, if it is close to the previous point, then $\boldsymbol{\mu}_k$ is close to the $\boldsymbol{x}_k$, $D(\boldsymbol{x}_k)$ stays close to 1 and μ_D stays close to the actual mean of the data points, as $(1 - \Delta_D)$ is very close to 1, which gives a more importance to the first term of Eq. (11). However, when the new point is far from the previous ones, $D(\boldsymbol{x}_k)$ slightly decreases, while μ_D becomes closer to $D(\boldsymbol{x}_k)$, as the term Δ_D gives more importance to the second term of Eq. (11). As new points are closer to the previous point, then $D(\boldsymbol{x}_k)$ continue to decrease until the condition in the Eq. (10) is satisfied. Note that, when the status changes from normal to anomalous, ks is set to 0, leading μ_D to notably decrease. When the mean density is lower than the global density for a certain number of points, or seconds, then the status returns to be normal. Thus, the condition expressed by Eq. (12) applies:

$$\begin{gathered} IF\ D(\boldsymbol{x}_k) > \mu_D \\ for\ k = t_2, \dots, k-1, k \\ THEN\ \boldsymbol{x}_k\ is\ normal \end{gathered} \quad (12)$$

Note that, both t_1 in Eq. (10) and t_2 in Eq. (12) are set by the user and may represent either the data samples or seconds.

3.3. Clustering

Based on the concepts of RDE, a clustering algorithm has also been introduced in Gu et al., (2018), in both offline and online version. Here, the online version will be briefly described. At the first iteration ($k = 1$) the local parameters of each cluster are initialized as follows

$$C_k = 1;\ \boldsymbol{\mu}_k^1 = \boldsymbol{x}_1;\ S_k^1 = 1 \quad (13)$$

Where, C_K is the cluster at the time stamp $k = 1$, $\boldsymbol{\mu}_1^1$ is the focal point of cluster 1 at the time stamp $k = 1$, and S_1^1 is the number of data points belonging to the cluster 1 at the time stamp $k = 1$. In addition, the global parameters expressed by Eq. (8), Eq. (9) are also initialized. Then, for each $k > 1$,

1. The mean value $\boldsymbol{\mu}_k$ and the scalar product Σ_k are updated by means of Eq. (8) and (9), respectively.
2. The condition expressed by Eq. (14) is checked to decide whether the current point has to be assigned to an existing cluster or should be a new focal point itself

$$IF\ D(\boldsymbol{x}_k) > \max_{i=1,\dots,C_k} D_k(\boldsymbol{\mu}_k^i)\ OR\ D(\boldsymbol{x}_k) < \min_{i=1,\dots,C_k} D_k(\boldsymbol{\mu}_k^i) \quad (14)$$

$$THEN\ \boldsymbol{x}_k\ becomes\ a\ new\ focal\ point$$

   The condition expressed in Eq. (14) means that if the global density is greater than, or lower than, the densities computed at each of the existing focal points (the density of each cluster), then the current point creates a new cluster.
3. If Eq. (14) is satisfied, then a new cluster is created, whose parameters are initialized by means of Eq. (15)

$$C_k = C_{k+1};\ \boldsymbol{\mu}_k^{C_k} = \boldsymbol{x}_k;\ S_k^{C_k} = 1 \quad (15)$$

4. Otherwise, the distance between the current feature vector $\boldsymbol{x}_k$ and the focal point of each existing cluster is computed, in order to decide whether the current point can be assigned to the closest cluster, by means of Eq. (16)

$$\begin{gathered} IF\ \|\boldsymbol{x}_k - \boldsymbol{\mu}_k^n\| < \sqrt{\Sigma_k - \|\boldsymbol{\mu}_k\|^2} \\ THEN\ \boldsymbol{x}_k\ is\ assigned\ to\ \boldsymbol{\mu}_k^n \end{gathered} \quad (16)$$

   Condition expressed by Eq. (16) means that if the distance between the current point and the nearest focal point $\boldsymbol{\mu}_k^n$ is lower than the distance among all points arrived until the current stamp, then the current point is assigned to the nearest cluster.
5. If the condition expressed by Eq. (16), then the parameters of the closest cluster C_k^n are updated by means of Eq. (17)

$$\boldsymbol{\mu}_k^n = \frac{S_k^n - 1}{S_k^n}\boldsymbol{\mu}_k^n + \frac{1}{S_k^n}\boldsymbol{x}_k;\ S_k^n = S_k^n + 1 \quad (17)$$

4. THE PROPOSED METHODOLOGY

In this section, a new methodology to perform streaming feature extraction and condition recognition is presented. It gathers in a unique algorithm the three models described in the previous section, which have been modified in order to be applied to the problem stated in Section 2. The goal of the proposed methodology is twofold: first, to recognize, in real-time, the occurrence of a change in the operating condition of the monitored machinery; second, to group similar observations representing the same operating condition, both known and unknown. The main assumptions are the following:

1. If a data point (i.e, a feature vector) is considered anomalous and creates a new cluster, then the system has entered an unknown operating condition.

2. If the point is anomalous and is assigned to an existing cluster, that is different from the current cluster, then the system has entered a known operating condition.
3. If an anomaly is detected, but the data point is assigned to the current cluster, then the operating condition has not changed, and the anomaly can be ignored (for instance, because it corresponds to a measurement error).

To achieve the above mentioned goals, the methodology starts from data collection and performs, as data becomes available, feature extraction, anomaly detection and clustering, as described in the previous section. Feature extraction is made of two steps: sampling and dimensionality reduction. The sampling step is performed after a certain number of observations is available, or a certain time has passed, in order to extract one or few more values (features) from a signal segment of a certain length. The dimensionality reduction is performed after the sampling step. Thus, each time the set of features extracted from all characteristic signals is available, the IPCA is triggered, so to extract a subset of relevant features, based on the actual sampled signals. Finally, the anomaly detection and the clustering algorithms are applied to the last extracted subset of features, in order to evaluate whether it is anomalous and group it accordingly. There are several decisions to take before applying the above described methodology.

First, the length of the signal segments, t_s, from which the features have to be extracted for the sampling purpose. It depends on how the signal evolves over time (for example, for rotating components, it may be equal to, or a multiple of, the rotation frequency) and on how quickly the feedback is needed.

Second, which features to extract from the signal segments. It depends on the kind of component and collected signals. For vibrations collected from rolling bearings, one can think to transform the signal into the time-frequency domain and extract the energy of the signal (Calabrese, Gamberi, Margelli, Pilati & Regattieri, 2019). For other kinds of signals, such as temperatures or pressures, features may be extracted directly in the time domain.

Third, the number of iterations n needed for the initialization of the covariance matrix $\boldsymbol{X}_n$ and the first PCs.

Finally, the values of t_1 and t_2, which are used by the anomaly detection algorithm for determining whether the set of extracted features is anomalous or not (Eq. (10) and Eq. (12)). These values depend on how rapidly the changes in the machinery behavior are expected to occur and how rapidly the system is expected to react. The higher the above two needs, the lower t_1 and t_2 should be. However, the lower t_1 and t_2 . the more sensitive the methodology is to measurement errors or point anomalies.

Once the above input parameters have been set, then the following step-by-step methodology can be applied, whose pseudo-code can be read in Fig. 2.

1. Given the time window equal to t_s, the mean value for each signal $x_j(t)$, where $(j = 1 \dots, p)$ is the number of the characteristic signals, is computed over t_s.

```
Read $\boldsymbol{t}_{s_i}$ data samples
Estract the mean value $\overline{\boldsymbol{x}}_{ij}$ over $\boldsymbol{t}_s$ data samples for each variable $j$
If i< $n$
  Compute $Z_n, Q_n, P_n$ by Eq. (1), Eq. (3), Eq. (4)
Else
  Compute $Z_i, Q_i, P_i$ by Eq. (5), Eq. (6), Eq. (4)
  If $S_k^* = 1$ (*current cluster)
    Select $P_{ij}$ $(j = 1, \dots, m)$ PCs such that $\lambda_1 + \dots + \lambda_m > 90\%$
    $\boldsymbol{x}_k = P_{ij}$
  Else
    $\boldsymbol{x}_k = \{P_{i-1}; P_i\}$
  End
If $k$=1
  $status = 0,\ ks = 1,\ D_{x_k} = 1, \boldsymbol{\mu}_k = \boldsymbol{x}_k, \Sigma_k = \|\boldsymbol{x}_k\|,\ \boldsymbol{\mu}_D = D_{x_k}$
  $C_k = 1, \boldsymbol{\mu}_k^1 = \boldsymbol{x}_k,\ S_k^1 = 1$
Else
  If $status = 0$
    Update $\boldsymbol{\mu}_k$ by Eq. (8)
    Update $\Sigma_k$ by Eq. (9)
    Update $D_{x_k}$ by Eq. (7)
    Compute $\Delta_D$
    Update $\boldsymbol{\mu}_D$ by Eq. (11)
    If condition (10) is satisfied
      $status = 1$
      $ks = 0$
      Compute the distance between $\boldsymbol{x}_k$ and $\boldsymbol{\mu}_k^i$ $(i = 1, \dots, C_k)$
      Find the nearest cluster $\boldsymbol{\mu}_k^n$
      If condition (16) is satisfied
        Update $C_k,\ \boldsymbol{\mu}_k^1, S_k^1$ by Eq. (17)
      Else
        Create a new cluster and initialize its parameters by Eq. (15)
      End
    Else
      Assign the point to the current cluster and update its parameters by Eq. (17)
    End
  Else
    If condition (12) is satisfied
      $status = 1$
      $ks = 0$
      Compute the distance between $\boldsymbol{x}_k$ and $\boldsymbol{\mu}_k^i$ $(i = 1, \dots, C_k)$
      Find the nearest cluster $\boldsymbol{\mu}_k^n$
      If condition (16) is satisfied
        Update $C_k,\ \boldsymbol{\mu}_k^1, S_k^1$ by Eq. (17)
      Else
        Create a new cluster and initialize its parameters by Eq. (15)
      End
    Else
      Assign the point to the current cluster and update its parameters by Eq. (17)
    End
  End
  Increase $k, ks$
End
```

Figure 2. The Pseudo-Code of the Proposed Methodology

2. After n iterations, i.e., after n mean values are available, the matrix $\boldsymbol{X}_n$ is obtained, where the generic element $\overline{\boldsymbol{x}}_{ij}$ represents the mean extracted from the original signal j $(j = 1, \dots, p)$ after t_{s_i} time windows $(i = 1, \dots, n)$.
3. The batch PCA is then applied to $\boldsymbol{X}_n$ in order to obtain the initial covariance matrix and the first m PCs that explain the 90% of the variance are selected. The extracted feature vector $\{\boldsymbol{x}_k\}, (k = n, n + 1, \dots)$ represents the input vector for the anomaly detection and clustering algorithms.
4. When $k = n$, the anomaly detection and clustering algorithms are triggered. Thus, the mean value, the scalar product and the global density are initialized as described in Eq. (8), Eq. (9) and Eq. (7), respectively. In addition, the assumption to be in a normal condition is made

($status = 0$). The extracted feature vector $\{\boldsymbol{x}_k\}$ $k = n$ creates the first cluster, whose parameters are initialized based on Eq. (15).

5. For each mean vector $\{\overline{\boldsymbol{x}}_i\}, (i > n)$, the incremental PCA is applied to the new matrix $\boldsymbol{X}_{i+1}$, which contains the old matrix $\boldsymbol{X}_n$ and the new mean vector $\{\overline{\boldsymbol{x}}_i\}$.
   a. If a new cluster has been created at the previous iteration, then the m PCs explaining the 90% of the variance are selected. That is, the operating condition has changed and the feature vector that best represents the new situation may change accordingly. Thus, the number of relevant PCs has to be re-evaluated.
   b. Otherwise, the same number of PCs extracted at the previous step is selected.

   In both cases, the feature vector $\{\boldsymbol{x}_k\}, k > n$, is obtained.
6. For each $\{\boldsymbol{x}_k\}$, the parameters of the anomaly detection algorithm, i.e., the mean value, the scalar product, the global density and the mean density, are updated based on Eq. (8), Eq. (9), Eq. (7) and Eq. (11), respectively.
7. The condition expressed by Eq. (10) is checked to establish whether the current point is an anomaly or not.
   a. If the global density computed at the current time stamp is lower than the mean density for a certain number of past observations (or for a certain time), that is, Eq. (10) is satisfied, than the current feature vector is an anomaly, the distance between it and the centers of all the N existing clusters $D_k(\boldsymbol{\mu}_k^l)$, $(l = 1, \ldots, N)$, is computed and the nearest cluster is identified.
      i. If the condition expressed in Eq. (16) is satisfied, than the point is assigned to the nearest cluster, whose parameters are updated by means of Eq. (17). This means that the anomaly corresponds to a change of the operating condition that is already known.
      ii. Otherwise, a new cluster is created, whose parameters are initialized based on Eq. (15). This means that the anomaly corresponds to a change of the operating condition that is unknown.
   b. If condition (10) is not satisfied, that is, if the global density computed at the current point is greater than the mean density for some observations in t_1, then the current feature vector is not an anomaly and the point is directly assigned to the current cluster, whose parameters are updated by means of Eq. (17).
8. Once a point is detected anomalous, then the system enters in an anomalous status $status = 1$
   a. If condition expressed by Eq. (12) is satisfied, then the system returns to a normal status, which could correspond to an existing or new condition. Thus, the nearest cluster is found and
      i. If the condition expressed by Eq. (16) is satisfied, then the point is assigned to the current cluster, whose parameters are updated by means of Eq. (17). This means that the system has entered a known operating condition.
      ii. Otherwise, a new cluster is created, whose parameters are initialized by means of Eq. (15). This means that the system has achieved a new stability condition, but that is different from the known ones.
   b. If the condition expressed in Eq. (12) is not satisfied, then the point is assigned to the current cluster, whose parameters are updated by means of Eq. (17). This means that the system continues to be in an anomalous status and the algorithm is still not able to evaluate whether the operating condition is changing or not.

5. CASE STUDY

In this section, the proposed procedure is applied to a sub-system of an automatic machinery, which has been operating in a real industrial context for many years. The system is made of two electric motors, a heated extruder and a volumetric pump. The inner screw is subject to a very slow degradation, that becomes relevant after two or three years of functioning. The group is provided with 30 sensors, which measure the temperature and the percentage of usage of thermo-resistors in different zones, the power and the velocity of the motors, as well as the input and output pressures of a volumetric pump, placed at the end of the extruder. Based on the expertise of technicians, it has been found that the percentage of usage of one of the electric motors is a good Health Indicator (HI). As it is shown in Figure 3, its trend is characterized by sudden jumps, which occur when the machinery setting changes. In addition, the degradation rate varies according to the specific setting.

The operating condition is determined by a set of temperature values, whose actual signals are collected during the machinery functioning. In order to establish the operating condition in which the machinery is working, the proposed procedure introduced in the previous paragraph will be applied. Results will be compared with the corresponding batch algorithms, i.e., the PCA and most adopted classification algorithms in the field of diagnostics, that are the Support Vector Machines (SVMs) and the K-Nearest Neighbor (K-NN).

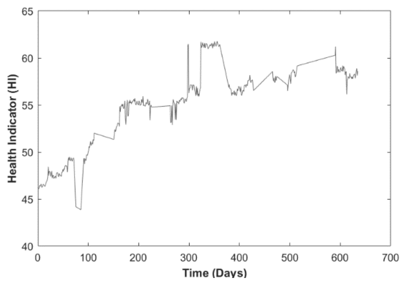

Figure 3. The Health Indicator of the monitored component

5.1. Dataset Description

Signals are collected in two different sources. In the first one, signals are acquired at a frequency of 1Hz. Each day, almost 8 hours of data are recorded; then, files are directly downloaded by the PLC of machinery. The second one is a low-frequency data source: each day, after 30 minutes of the machine functioning, four different statistics over a batch of 30 seconds for each collected signal are computed. The actual characteristic signals are collected in the high-frequency data source.
The available dataset includes data related to two units. As summarized in the Table 1, the first unit was monitored for almost 11 months, while the second unit was monitored for almost 21 months. For each unit, two different settings were implemented. The information related to the setting change are summarized in Table 2. While for the first unit there is only a change from setting 1 to setting 2, in the second unit, there are two changes: from setting 3 to setting 4 and from setting 4 to setting 3.
It worth to note that the data were not recorded each day. This situation is very common in real applications, especially when the implementation of a predictive maintenance program is in its initial phase. In addition, transients nor failures were recorded, and no maintenance intervention was realized during the considered period.

Table 1. Dataset Description.

Unit	Period of collection	Settings
1	From 2017-10-20 to 2018-09-17	2
2	From 2017-01-12 to 2018-09-06	2

Table 2. Setting changes.

Unit	Period	Setting
1	From 2017-10-20 to 2017-11-03	1
	From 2017-12-04 to 2018-09-17	2
2	From 2017-01-12 to 2017-06-12	3
	From 2017-06-12 to 2017-06-26	4
	From 2017-06-26 to 2018-09-06	3

5.2. The proposed methodology

For feature extraction, 11 variables (temperatures) have been considered. As shown in the Figure 4, where both set value and actual value are depicted, the actual values of the setting points oscillate around the set value.
First, a the sampling step has been conducted before extracting relevant features. The mean value of each variable is computed over a batch 1800 samples, that correspond to a 30 minutes of data. This value has been arbitrarily chosen, based on the following factors: the accuracy of prediction, the latency of the algorithm, the memory storage of the possible edge device and its computational capacity. In this case, the batch of 1800 samples represents a good compromise among all these factors. However, it demonstrated to effectively smooth the signal with no loss of information.

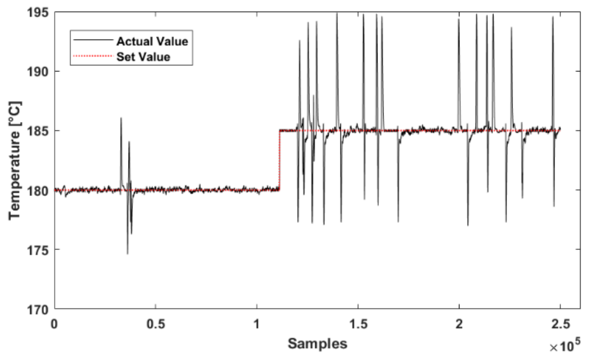

Figure 4. Setting: set value vs. actual value

Therefore, the Incremental PCA is applied to the 11 sampled signals. Figure 5 shows the results of the Incremental PCA, compared with those obtained from a batch PCA, for unit 1 and unit 2, respectively.

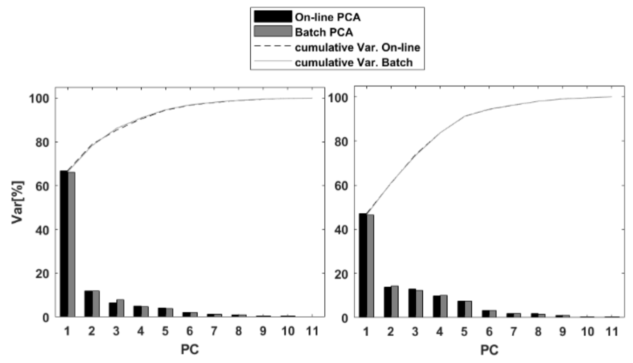

Figure 5. Performance of Incremental PCA vs. batch PCA for unit 1 (left) and unit 2 (right)

As shown in Table 3, the first four PCs extracted from the data related to the first unit are able to retain the 90% of the variance during all the analysis. The same variance is retained by the first five PCs for the second unit. The trend of these PCs are shown in Figure 6 (unit 1) and Figure 7 (unit 2).

Therefore, instead of considering the eleven variables, only four (unit 1) and five (unit 2) PCs are necessary for condition recognition. Indeed, based on the extracted PCs, the proposed algorithm is able to correctly recognize the change of the operating condition, as shown in Figure 8 and Figure 9, where the black dots correspond to the first setting, the grey dots to the second setting and the red crosses represent the moment in which a changing behavior is detected. As summarized in Table 4, in both cases, the algorithm recognizes the switch from setting 1 to setting 2 and from setting 3 and setting 4 after 9 data samples. In addition, for unit 2, the algorithm also recognizes the switch from setting 4 to setting 3, as data points are assigned to the existing cluster corresponding to the first operating condition.

Note that, in both cases, there are several data points considered anomalous. However, none of them creates a new cluster. These points correspond to true anomalies in the data, like measurement errors or anomalous peaks, which are evident in the raw signals.

Table 3. Dataset Description.

Unit	PC	Variance	Cumulative Variance
1	1	66,45%	66,45%
	2	11,88%	78,33%
	3	7,81%	86,14%
	4	4,77%	90,91%
2	1	46,61%	46,61%
	2	14,49%	61,1%
	3	12,46%	73,56
	4	10,16%	83,72%
	5	7,55%	91,27%

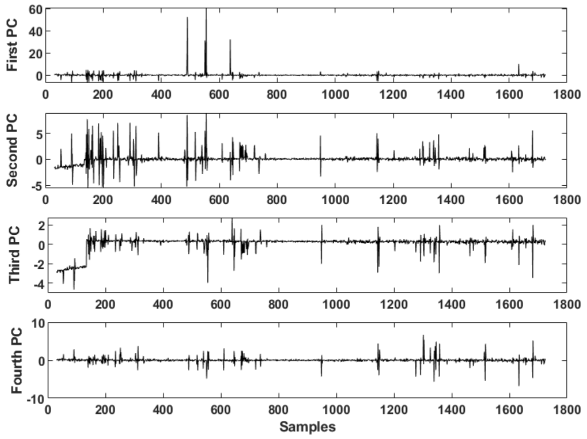

Figure 6. Relevant PCs (unit 1)

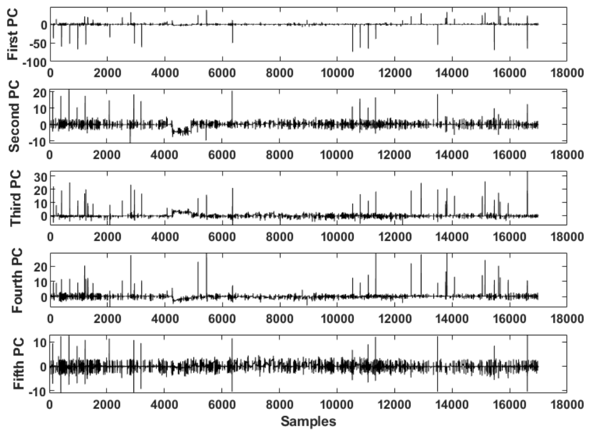

Figure 7. Relevant PCs (unit 2)

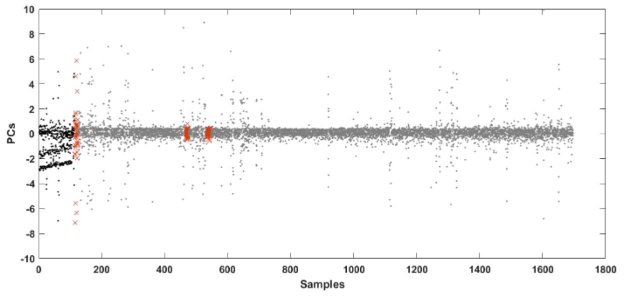

Figure 8. Online Anomaly Detection and Clustering (unit 1)

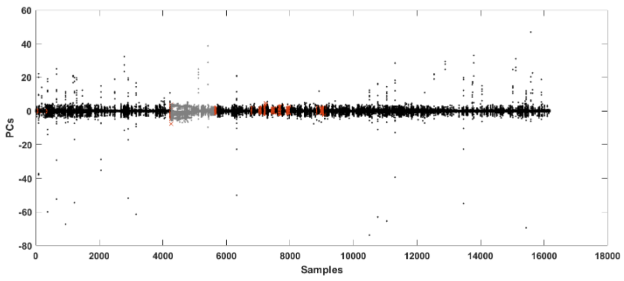

Figure 9. Online Anomaly Detection and Clustering (unit 2)

Table 4. Clustering Results.

Unit	Setting change (Detected)	Setting change (Real)
1	114	105
2	4257	4248
	5674	4904

5.3. Supervised approach

To validate the proposed methodology, a supervised analysis has been conducted. In particular, the batch PCA has been applied to the matrix $\boldsymbol{X}$, where the generic element x_{ij} corresponds to the mean value computed over 1800 data points at time i for the characteristic signal j $(j = 1, \dots, 11)$. The PCs retaining the 90% of the variance have been extracted. As expected, for unit 1, 4 PCs have been extracted, while for unit 2, the number of the extracted PCs is equal to 5. Then, two classification models, Support Vector Machine (SVM) and K-Nearest Neighbor (K-NN) have been applied to both the original matrix ***X*** and the matrix of extracted PCs. Results for both the units are shown in Table 5. Both models, in both cases, without and with PCA, provide the 100% of accuracy in classifying the different operating conditions.

Table 5. Classification Results.

Unit 1			
Without PCA		With PCA	
Model	Accuracy	Model	Accuracy
SVM	100%	SVM	100%
K-NN	100%	K-NN	100%
Unit 2			
Without PCA		With PCA	
Model	Accuracy	Model	Accuracy
SVM	100%	SVM	100%
K-NN	100%	K-NN	100%

5.4. Results and discussions

The proposed methodology, that includes feature extraction, anomaly detection and clustering, is able to recognize a change in the operating condition. In addition, it is able to group data that are related to the same operating condition into the same cluster. No prior information about belonging class is provided and no data distribution is assumed.
An important step in the methodology is represented by the incremental feature extraction. As shown in Table 5, the supervised approach provides a 100% of prediction accuracy in both cases, without and with PCA. This result justifies the implementation of the incremental PCA into the streaming methodology, as it allows to reduce the quantity of data to store (4 or 5 variables against the 11 original variable), while keeping the accuracy of prediction at the same value. This is also possible as incremental PCA is performed every 30 minutes, and thus the time for computation is no strict.
Note that, while the application of batch PCA to the batch and supervised analysis does not improve the accuracy of the classification (always equal to the 100%), the integration of the incremental PCA into the streaming methodology strongly affects the results. As shown in Table 6, for the Unit 1, the algorithm recognizes the occurrence of two different settings. However, its latency is much higher (30 data points). Instead, for the Unit 2, results are much worse. The change from setting 3 (Cluster 1) to setting 4 (Cluster 2) is detected with a higher latency, as well as the change from setting 4 to setting 3. In addition, another cluster (Cluster 3) is created. This would imply that another setting change has occurred, which is not true.

Table 6. Clustering Results without IPCA.

Unit	Setting change (Detected)	Setting change (Real)
1	144 – Cluster 2	105
2	4287 – Cluster 2	4248
	5702 – Cluster 1	4904
	7751 – Cluster 3	-
	8000 – Cluster 1	-
	8017 – Cluster 3	-

Thus, the proposed methodology provides promising results in terms of novelty detection and clustering. As shown in Table 4, the latency of the algorithm in recognizing the change of setting is equal to 9 data points in two cases out of three. In the last case, which corresponds to the recognition of an existing condition, i.e., the assignment to an existing cluster, the latency is much higher. This is because of two main reasons: the first one is that there are few data samples belonging to the setting 2 of unit 2. Thus, the algorithm has not enough time to let the parameters be stable. Second, a lot of anomalous peaks in the original signals occurred for unknown reasons. Thus, both the global density and the mean density trends are so fluctuating to make condition expressed in Eq. (10) never satisfied.

6. CONCLUSIONS

In this paper, a new methodology for condition recognition is presented. The rationale is that, very often, the degradation of a certain component/system may be affected by the operating condition in which the machinery is functioning. Thus, if the operating condition is known, the corresponding degradation model can be applied in order to compute the RUL. However, operating conditions are not known a priori: they may change because of machine users, the processed material, the production parameters as well as environmental conditions. In addition, the development of technologies as IoT, Cloud Computing and Edge Computing make possible the implementation of Predictive Maintenance directly at the edge of the network, where some tasks can be carried out in order to get a real-time feedback of the health status of machinery. Thus, the proposed methodology aims to perform a streaming and unsupervised analysis, that can be performed at the edge, and includes an incremental PCA for feature extraction, and a novelty detection and clustering algorithm based on RDE concept.
First, the problem of condition recognition is stated. It requires to identify the signals that reflect the implemented operating condition, to extract the relevant features for each condition, to recognize when a change of the operating condition occurs and finally to group the data related to the same operating condition into the same cluster. Second, related works are investigated and the mathematical

background of the adopted algorithms for incremental feature extraction, anomaly detection and clustering, is provided. Then, the proposed methodology is described, which is completely unsupervised and can be applied "from scratch". The only assumption is related to the first data samples, which are assumed to belong to a known operating condition. The incremental feature learning allows to select the relevant features according to actual data; the anomaly detection algorithm recognizes if a change in the behavior is occurring; the clustering algorithm decides whether the anomaly corresponds to an existing cluster, to a new cluster or just to a measurement error. A case of an automatic machinery operating in a real industrial context, whose monitored subsystem is subject to very slow degradation that depends on the implemented machinery setting, is presented. The case study shows that the proposed methodology is able to recognize when the operating condition has already occurred or is new, and to assign the data sample accordingly.

The advantages of the proposed methodology can be summarized as follows: first, when the computed HI is subject to sudden jumps depending on the operating conditions, and no information is available on which condition is implemented, the presented methodology allows to associate the trend of the HI in a specific time to a specific operating condition and thus, if known, to update the corresponding degradation model for the RUL prediction. In addition, if the condition is unknown, the methodology allows to automatically group similar observations in a unique cluster, which correspond to a new operating condition. This activity can be seen as an automatic labelling procedure. Finally, the distinction of known conditions from the new ones allows to keep in memory only the extracted features, when the condition is known, and store all the collected signals, when the condition is unknown. In this way, the storage capacity of the edge device, the bandwidth for data transmission, as well as the policies for the data transmission to the cloud, can be notably reduced.

Further research will be dedicate to (1) the integration of a classification model that, for each cluster, creates a class, so that it can be applied in the streaming procedure for assigning the data to existing operating condition, (2) the integration in the streaming methodology of RUL prediction based on degradation models trained on data related to the known operating conditions.

REFERENCES

Angelopoulos, A., Michailidis, E. T., Nomikos, N., Trakadas, P., Hatziefremidis, A., Voliotis, S., & Zahariadis, T. (2020). Tackling faults in the industry 4.0 era—a survey of machine-learning solutions and key aspects. *Sensors (Switzerland)*, *20*(1), 1–34. https://doi.org/10.3390/s20010109

Angelov, P., Ramezani, R., & Zhou, X. (2008). Autonomous novelty detection and object tracking in video streams using evolving clustering and Takagi-Sugeno type neuro-fuzzy system. *Proceedings of the International Joint Conference on Neural Networks*, 1456–1463. https://doi.org/10.1109/IJCNN.2008.4633989

Benkedjouh, T., Medjaher, K., Zerhouni, N., & Rechak, S. (2013). Remaining useful life estimation based on nonlinear feature reduction and support vector regression. *Engineering Applications of Artificial Intelligence*, *26*(7), 1751–1760. https://doi.org/10.1016/j.engappai.2013.02.006

Bian, L., Gebraeel, N., & Kharoufeh, J. P. (2015). Degradation modeling for real-time estimation of residual lifetimes in dynamic environments. *IIE Transactions*, *8830*(December 2017). https://doi.org/10.1080/0740817X.2014.955153

Bose, S. K., Kar, B., Roy, M., Gopalakrishnan, P. K., & Basu, A. (2019). AdepoS: Anomaly detection based power saving for predictive maintenance using edge computing. *Proceedings of the Asia and South Pacific Design Automation Conference, ASP-DAC*, 597–602. https://doi.org/10.1145/3287624.3287716

Bowden, D., Marguglio, A., Morabito, L., Napione, C., Panicucci, S., Nikolakis, N., Makris, S., Coppo, G., Andolina, S., Macii, A., Becker, P., & Jung, S. (2019). A cloud-to-edge architecture for predictive analytics. *CEUR Workshop Proceedings*, *2322*.

Calabrese, F, Gamberi, M., Margelli, S., & Pilati, F. (2019). Data-driven prognostics: from an offline and supervised analysis to an innovative, online and unsupervised methodology. *Proceedings of the Summer School Francesco Turco*, *1*, 74–80.

Calabrese, Francesca, Regattieri, A., Botti, L., & Galizia, F. G. (2019). Prognostic Health Management of Production Systems. New Proposed Approach and Experimental Evidences. *Procedia Manufacturing*, *39*, 260–269. https://doi.org/10.1016/j.promfg.2020.01.333

Cariño, J. A., Delgado-Prieto, M., Iglesias, J. A., Sanchis, A., Zurita, D., Millan, M., Ortega Redondo, J. A., & Romero-Troncoso, R. (2018). Fault Detection and Identification Methodology under an Incremental Learning Framework Applied to Industrial Machinery. *IEEE Access*, *6*, 49755–49766. https://doi.org/10.1109/ACCESS.2018.2868430

Cariño, J. A., Delgado-Prieto, M., Zurita, D., Picot, A., Ortega, J. A., & Romero-Troncoso, R. J. (2020). Incremental novelty detection and fault identification scheme applied to a kinematic chain under non-stationary operation. *ISA Transactions*, *97*, 76–85. https://doi.org/10.1016/j.isatra.2019.07.025

Costa, B. S. J., Angelov, P. P., & Guedes, L. A. (2015a). Fully unsupervised fault detection and identification based on recursive density estimation and self-evolving cloud-based classifier. *Neurocomputing*, *150*(Part A), 289–303. https://doi.org/10.1016/j.neucom.2014.05.086

Costa, B. S. J., Angelov, P. P., & Guedes, L. A. (2015b). Fully unsupervised fault detection and identification based on recursive density estimation and self-evolving cloud-based classifier. *Neurocomputing*, *150*(Part A), 289–303. https://doi.org/10.1016/j.neucom.2014.05.086

Datong, L., Yu, P., & Xiyuan, P. (2011). *Online Adaptive Status Prediction Strategy for Data - Driven Fault Prognostics of Complex Systems*.

Dyer, K. B., Capo, R., & Polikar, R. (2014). Compose: A semisupervised learning framework for initially labeled nonstationary streaming data. *IEEE Transactions on Neural Networks and Learning Systems*, *25*(1), 12–26. https://doi.org/10.1109/TNNLS.2013.2277712

Gu, X., Angelov, P. P., & Príncipe, J. C. (2018a). A method for autonomous data partitioning. *Information Sciences*, *460–461*, 65–82. https://doi.org/10.1016/j.ins.2018.05.030

Gu, X., Angelov, P. P., & Príncipe, J. C. (2018b). A method for autonomous data partitioning. *Information Sciences*, *460–461*, 65–82. https://doi.org/10.1016/j.ins.2018.05.030

Hu, R., Zhu, X., Cheng, D., He, W., Yan, Y., Song, J., & Zhang, S. (2017). Graph self-representation method for unsupervised feature selection. *Neurocomputing*, *220*, 130–137. https://doi.org/10.1016/j.neucom.2016.05.081

Hu, Y., Baraldi, P., Di, F., & Zio, E. (2017). A Systematic Semi-Supervised Self-adaptable Fault Diagnostics approach in an evolving environment. *Mechanical Systems and Signal Processing*, *88*(October 2016), 413–427. https://doi.org/10.1016/j.ymssp.2016.11.004

Hu, Y., Baraldi, P., Di Maio, F., & Zio, E. (2017). A Systematic Semi-Supervised Self-adaptable Fault Diagnostics approach in an evolving environment. *Mechanical Systems and Signal Processing*, *88*, 413–427. https://doi.org/10.1016/j.ymssp.2016.11.004

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

Katona, A., & Panfilov, P. (2018). Building predictive maintenance framework for smart environment application systems. *Annals of DAAAM and Proceedings of the International DAAAM Symposium*, *29*(1), 0460–0470. https://doi.org/10.2507/29th.daaam.proceedings.068

Lee, J., Wu, F., Zhao, W., Ghaffari, M., Liao, L., & Siegel, D. (2014). Prognostics and health management design for rotary machinery systems - Reviews, methodology and applications. *Mechanical Systems and Signal Processing*, *42*(1–2), 314–334. https://doi.org/10.1016/j.ymssp.2013.06.004

Lei, Y., Li, N., Guo, L., Li, N., Yan, T., & Lin, J. (2018). Machinery health prognostics: A systematic review from data acquisition to RUL prediction. In *Mechanical Systems and Signal Processing* (Vol. 104, pp. 799–834). Academic Press. https://doi.org/10.1016/j.ymssp.2017.11.016

Lippi, V., & Ceccarelli, G. (2019). Incremental principal component analysis: Exact implementation and continuity corrections. *ICINCO 2019 - Proceedings of the 16th International Conference on Informatics in Control, Automation and Robotics*, *1*, 473–480. https://doi.org/10.5220/0007743604730480

Liu, R., Yang, B., Zio, E., & Chen, X. (2018). Artificial intelligence for fault diagnosis of rotating machinery: A review. In *Mechanical Systems and Signal Processing* (Vol. 108, pp. 33–47). Academic Press. https://doi.org/10.1016/j.ymssp.2018.02.016

Mobley, R. K. (2002). Role of Maintenance Organization. In *An Introduction to Predictive Maintenance* (pp. 43–59). https://doi.org/10.1016/b978-075067531-4/50003-8

Moghaddass, R., & Zuo, M. J. (2014). An integrated framework for online diagnostic and prognostic health monitoring using a multistate deterioration process. *Reliability Engineering and System Safety*, *124*, 92–104. https://doi.org/10.1016/j.ress.2013.11.006

Nguyen, K. T. P., & Medjaher, K. (2019). A new dynamic predictive maintenance framework using deep learning for failure prognostics. *Reliability Engineering and System Safety*, *188*, 251–262. https://doi.org/10.1016/j.ress.2019.03.018

Qian, G., Lu, S., Pan, D., Tang, H., Liu, Y., & Wang, Q. (2019). Edge Computing: A Promising Framework for Real-Time Fault Diagnosis and Dynamic Control of Rotating Machines Using Multi-Sensor Data. *IEEE Sensors Journal*, *19*(11), 4211–4220. https://doi.org/10.1109/JSEN.2019.2899396

Si, X. S., Wang, W., Hu, C. H., & Zhou, D. H. (2011). Remaining useful life estimation - A review on the statistical data driven approaches. In *European Journal of Operational Research* (Vol. 213, Issue 1, pp. 1–14). Elsevier B.V. https://doi.org/10.1016/j.ejor.2010.11.018

Sikorska, J. Z., Hodkiewicz, M., & Ma, L. (2011). Prognostic modelling options for remaining useful life estimation by industry. *Mechanical Systems and Signal Processing*, *25*(5), 1803–1836. https://doi.org/10.1016/j.ymssp.2010.11.018

Wang, J., Li, S., Jiang, X., & Cheng, C. (2017). An automatic feature extraction method and its application in fault diagnosis. *Journal of Vibroengineering*, *19*(4), 2521–2533. https://doi.org/10.21595/jve.2017.17906

Xia, T., Dong, Y., Xiao, L., Du, S., Pan, E., & Xi, L. (2018). Recent advances in prognostics and health management for advanced manufacturing paradigms.

In *Reliability Engineering and System Safety* (Vol. 178, pp. 255–268). Elsevier Ltd. https://doi.org/10.1016/j.ress.2018.06.021

Yaseen, M., Swathi, D., & Kumar, T. A. (2018). IoT based condition monitoring of generators and predictive maintenance. *Proceedings of the 2nd International Conference on Communication and Electronics Systems, ICCES 2017*, *2018-Janua*(Icces), 725–729. https://doi.org/10.1109/CESYS.2017.8321176

Ye, Z., Wang, Y., Member, S., & Tsui, K. (2013). *Degradation Data Analysis Using Wiener Processes With Measurement Errors*. *62*(4), 772–780.

Ye, Z., & Xie, M. (2015). *Stochastic modelling and analysis of degradation*. *September 2014*. https://doi.org/10.1002/asmb.2063

Yu, W. E. I., Liang, F. A. N., He, X., Hatcher, W. G., Lu, C., Lin, J. I. E., & Yang, X. (2018). A Survey on the Edge Computing for the Internet of Things. *IEEE Access*, *6*, 6900–6919. https://doi.org/10.1109/ACCESS.2017.2778504

Zhai, Q., & Ye, Z. S. (2017). RUL Prediction of Deteriorating Products Using an Adaptive Wiener Process Model. *IEEE Transactions on Industrial Informatics*, *13*(6), 2911–2921. https://doi.org/10.1109/TII.2017.2684821