Available online at www.sciencedirect.com

ScienceDirect

Procedia Manufacturing 39 (2019) 260–269

Procedia MANUFACTURING

www.elsevier.com/locate/procedia

25th International Conference on Production Research Manufacturing Innovation:
Cyber Physical Manufacturing
August 9-14, 2019 | Chicago, Illinois (USA)

Prognostic Health Management of Production Systems. New Proposed Approach and Experimental Evidences

Francesca Calabrese[a]*, Alberto Regattieri[a], Lucia Botti[b], Francesco Gabriele Galizia[c]

[a]Department of Industrial Engineering,University of Bologna, Bologna, 40136, Italy
[b]Interdepartment Research Center on Security and Safety, University of Modena and Reggio Emilia, 41121, Modena, Italy
[c]Department of Management and Engineering, University of Padova, 36100, Italy

Abstract

Prognostic Health Management (PHM) is a maintenance policy aimed at predicting the occurrence of a failure in components and consequently minimizing unexpected downtimes of complex systems. Recent developments in condition monitoring (CM) techniques and Artificial Intelligence (AI) tools enabled the collection of a huge amount of data in real-time and its transformation into meaningful information that will support the maintenance decision-making process. The emerging Cyber-Physical Systems (CPS) technologies connect distributed physical systems with their virtual representations in the cyber computational world. The PHM assumes a key role in the implementation of CPS in manufacturing contexts, since it allows to keep CPS and its machines in proper conditions. On the other hand, CPS-based PHM provide an efficient solution to maximize availability of machines and production systems. In this paper, evolving and unsupervised approaches for the implementation of PHM at a component level are described, which are able to process streaming data in real-time and with almost-zero prior knowledge about the monitored component. A case study from a real industrial context is presented. Different unsupervised and online anomaly detection methods are combined with evolving clustering models in order to detect anomalous behaviours in streaming vibration data and integrate the so-generated knowledge into supervised and adaptive models; then, the degradation model for each identified fault is built and the resulting RUL prediction model integrated into the online analysis. Supervised methods are applied to the same dataset, in batch mode, to validate the proposed procedure.

* Corresponding author. Tel.: 0512093420
E-mail address: francesca.calabrese9@unibo.it

10.1016/j.promfg.2020.01.333

Keywords: Prognostic Health Management (PHM); Cyber-Physical Systems (CPS); Predictive Maintenance; Fault Detection; Intelligent Diagnostics

1. Introduction

Maintenance of complex systems is responsible for keeping machinery in their correct functioning condition, so that unexpected downtimes are minimized. Recent researches are moving towards the adoption of predictive strategies, and in particular the Prognostics Health Management (PHM), that relies on Machine Learning (ML) and Artificial Intelligence (AI) algorithms to extract knowledge from raw signals collected from sensors installed on machinery and compute their Remaining Useful Life (RUL) [1],[2],[3]. The PHM approach typically requires to conduct several tests simulating the known fault conditions, to collect the training sets for model learning. There are two main issues that limit the application of this approach to industrial contexts: (1) the collection of data is a time-consuming activity and requires an *a priori* knowledge of all fault causes; (2) traditional learning models are trained *offline*, with batch data, and are not able to adapt to the actual condition of the equipment.

Within the concepts of Industry 4.0, manufacturing industries are becoming more digitalized and intelligent. One of the most enabling technology is the Internet of Things (IoT), whose basic concept is the pervasive presence around us of a variety of objects interacting and cooperating with each other to reach common goals [4]. According to the IoT technologies, all physical elements of Smart factories are connected and exchange information with each other, through sensors/actuators deployed in a wide area and modern Wireless communications. In this context, the huge amount of collected data needs to be stored, managed, shared, analyzed and computed to find lows and knowledge supporting self-managed and adaptive decision-making processes [5]. This situation leads to the necessity of interaction between the physical world and cyber world. Cyber-Physical Systems (CPS) communicate and cooperate with each other and humans in real-time, creating a virtual copy of the monitored physical processes and elements and making decentralized decisions [6]. CPS are able to realize the real-time, safe, reliable and dynamic collaboration with physical systems, which are provided with distributed computing systems that collect data, pass the data to the computing layer according to the demands of end devices which complete the given tasks [7].

The integration of IoT, CPS and PHM should provide a systematical view of the machine health prognosis, the machine-level maintenance scheduling and the system-level maintenance optimization [8]. In addition, the integration of CPS cloud computing and IoT edge computing allows to perform fault detection in real-time, with an algorithm that is able to self-adapt to the dynamic environment of industrial contexts, while accumulating training sets during the machine functioning and analyzing them through accurate ML algorithms performed into the cloud [9], [10].

The main goal of this paper is to provide a novel, *completely unsupervised and partially online PHM methodology*, which relies on streaming data collected from sensors and performs computation either at the edge or into the cloud, depending on the end device requirements. In particular, a light weight anomaly detection algorithm is adopted and implemented at the network edge, in order to detect deviating behaviors in real-time. Then, if a failure is recognized, the collected data at the edge are transmitted into the cloud, where a degradation model is built during an *offline* analysis. Finally, when the machine condition is restored, the degradation model updating and RUL prediction can be included into the computation carried out by edge devices, so that the machine operator can immediately recognize anomalous behaviors as well as the time left to the monitored equipment before reaching the FT (RUL).

The remaining of the paper is organized as follows. In Section 2, main concepts for the implementation of PHM in industrial context are provided, focusing on the difference between supervised and unsupervised approaches, classification, regression, clustering and anomaly detection tasks, and *online* and *offline* computing. In Section 3, concepts and tools of industrial CPS, IoT and IoT-based CPS are presented, as key technologies enabling the implementation of timely maintenance. In Section 4, the proposed methodology is introduced. In particular, anomaly detection algorithm, data partitioning algorithm and degradation modelling are briefly described and integrated into an IoT-based CPS structure. In Section 5, the proposed methodology is applied and validated, based on current signals corresponding to different fault classes of a real industrial machine. Finally, experimental evidences and the issues emerged from the application are highlighted.

2. Prognostics Health Management

Prognostics Health Management (PHM) is a step-wise approach, made of data collection, signal processing, diagnostics and prognostics. Parameters like vibrations, acoustic emissions and others, are first collected from critical components by means of one or more sensors; then, relevant information able to reveal the health status of the equipment, i.e., *features*, are extracted from raw signals [11]. At this point, the fault is detected, isolated and identified as a pattern recognition problem, in which the relationships between the extracted features and the corresponding fault classes have to be established (diagnostics). Finally, prognostics is carried out, which includes the prediction of the degradation process for each identified fault according to historical information of the degradation trend, and with the prediction of the time length of the degradation curve from the current state, based on a pre-fixed FT.

Due to the large volume of data collected from machinery, several Machine Learning (ML) and Artificial Intelligence (AI) algorithms can be adopted for feature extraction, diagnostics and prognostics [12], [13], [14], [15]. All these algorithms can be classified based on three criteria: the nature of the available data, the task to accomplish, the time in which results are needed. Suppose to have a large data set $D \in \mathbb{R}^{m \times n}$, in which all the observations $\{x_1, x_2, \dots, x_m\}$ are associated with a certain number of exploratory attributes $\{a_1, a_2, \dots, a_n\}$ and a target variable that can a assume a finite number of numerical or categorical values.

Supervised, unsupervised and semi-supervised learning approaches can be distinguished, depending on whether the target value of each observation is considered at the moment of prediction. Specifically, in supervised learning, the observations are labelled with the corresponding fault class, and relationships between the exploratory attributes and target variable are found with respect to the target variable [16]. Unsupervised learning is not guided by the target variable, rather it aims to find hidden pattern in the data set, only based on the data structure [17]. Finally, in semi-supervised learning, as a combination of the two previously described approaches, both labelled and unlabeled observations are considered during the learning process.

Traditionally, learning algorithms can be adopted for either for classification or regression purposes [18]. In classification problems, labelled observations are used for building models (training step) that are able to predict the discrete target variable (i.e., the fault class) for future observations, whose only exploratory attributes are known [19]. Regression is used for predicting the value of the target variable when it can assume continuous values. Therefore, while classification is adopted during the diagnostics step, giving as output the fault class to which each new observation belongs, regression process is used to predict the values assumed by attributes in the future, to estimate when the FT will be reached and thus, the RUL of the monitored component [20]. In addition, anomaly detection and clustering algorithm are attracting more attention in the context of PHM. In particular, unsupervised and *online* anomaly detection algorithms are mostly adopted to immediately recognize whether a certain data point is similar to, or differs from, previous points. In a PHM program, these algorithms can be used for fault detection, in order to determine if a change in the system behavior is occurring [21]. Clustering is an unsupervised learning task in which a set of unlabelled data is grouped into clusters, so that objects belonging to the same clusters are similar between each other and dissimilar to objects belonging to other clusters. In a PHM program, clustering algorithms are used for the diagnostics step when no labelled data is available, in order to create groups of observations that correspond to the same fault class.

In regard to the time in which model results need to be available, *offline* or *online* approaches exist. *Offline* analysis considers the whole data set available when building the model, while *online* analysis is used for streaming data, that is continuously generated over time, is potentially unbounded and cannot be saved permanently to memory [22]. Therefore, while *offline* analysis is more focused on the accuracy and comprehensiveness of analysis, *online* approaches are oriented towards a real-time analysis. In particular, they are mostly used for fault detection, degradation models updating and RUL prediction, making models self-adaptive as new data are available and suitable for dynamic industrial environments [23], [24].

Due to the issues in the application of PHM to real industrial contexts, unsupervised and *online* approaches may be promising for real-time PHM implementation. In particular, the PHM program proposed in [25] is more suitable for an *online* application, as it considers the variable *time*. The main differences from the traditional approach are in the computation of a monotonic Health Indicator (HI), that directly describes the degradation process, instead of the features, and in the Health Stage (HS) division instead of the diagnostics, according to which the whole life of the component is divided into different stages, each corresponding to a certain severity of the fault condition.

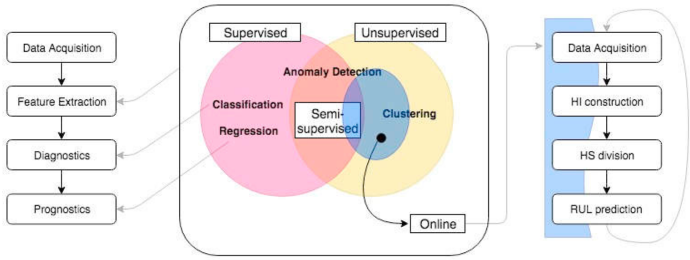

Fig. 1. Machine Learning for PHM: a supervised and offline scheme (left) and an unsupervised and online methodology (right).

3. IoT-based Industrial Cyber-Physical Systems

Industrial Cyber-Physical Systems (I-CPS) integrate the physical world, made of sensors, actuators and equipment deployed in the industrial plant, with the cyber world, composed by the computation, networking and control systems, that collect and analyze data from both the physical and digital worlds to enable the operation, interconnection and provide industrial systems with intelligence. In I-CPS, machines devices exchange information among each other through the Machine-to-Machine (M2M) communication model, realized with “Wired” or “Wireless” networks [26].

IoT technologies in industrial contexts make possible the interconnection between intelligent industrial devices and control and management platforms. In I-IoT systems, the data source is represented by the sensors deployed in a wide area that provide different types of data. The requirements generated by end devices are carried out into the cloud through the core network. Thanks to its significant storage and computational capabilities, the cloud is responsible for data processing. Often, IoT gateways are also included as intermediaries between the sensors/end devices and the cloud servers, for making data pre-processing and reducing redundancy and unnecessary overhead [27].

The I-IoT corresponds to the integration of communication layers of I-CPS [28], whose cyber space components, i.e., control, networking and computing systems, are provided by IoT technologies. In particular, the control system provides self-awareness and self-diagnosis capabilities to machines, enabling the detection and classification of failures for efficient management, effective utilization and timely maintenance. The networking system enables the communication among isolated industrial devices. The computing system provides the computational platform for time collection, storage, processing and analysis of the data [29].

Traditionally, the I-IoT requirements, i.e., transmission, storage and computing [27], have been satisfied by cloud-based structures, made of centralized servers that collect data from the connected devices and send back to them the results after data processing. Although its high storage and processing capacity, high reliability, scalability and interoperability, cloud-based computing has several drawbacks from the data transmission point of view. First, the significant volume of data to transfer to the cloud may overload the network. Second, delays can occur in data transmission due to the far location of the end devices with respect to the cloud, that is especially critical for time-sensitive requirements and real-time applications [29].

Hence, edge computing is attracting great attention, since it moves data computing and storage from the cloud to the network edge located nearby the users. In this way, the peak in traffic flows, the bandwidth requirements of the centralized network and the transmission latency during data computing and storage is strongly reduced. Specifically, (1) due to the short transmission time of edge computing, the latency of the network is reduced, enabling real-time collection and analysis of the information; (2) due to its location, the edge-based storage provides short upload time of massive data; (3) Even though limited, edge devices have enough capacity to satisfy real-time IoT requirements. Finally, edge nodes mitigate the power consumption of the IoT devices through the computation tasks offloading [27].

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 [27]. Moreover, they are not able to perform complex data analytics. Hence, an IoT-based I-CPS can be realized through the integration of cloud and edge computing, that locally processes high priority and delay sensitive tasks while processes low priority and delay tolerant tasks in the cloud [29].

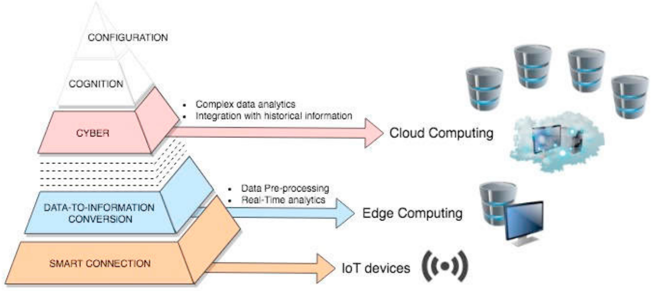

Fig. 2. A CPS architecture based on IoT technology

In [30], a five-level architecture for I-CPS implementation is proposed. First, in the smart connection layer, accurate and reliable data are acquired from machines from sensors. A second layer is dedicated to the data-to-information conversion, in which algorithms transform data into meaningful information in order to bring self-awareness to machines. The third layer is the cyber layer, that acts as a central information hub, which receives information from all connected devices and integrates them with historical information, providing machines with self-comparison ability. Then, in the cognition level, the priority of tasks to optimize maintenance process is decided. Finally, a supervisory control from the cyber to physical devices makes machine self-configurable and self-adaptive. As shown in Fig. 2, in an IoT-based CPS, smart connection can be realized through the IoT devices, the data-to-information conversion at the edge for data pre-processing and real-time analytics, the cyber layer could be represented by only the cloud computing, in which complex data analytics and the integration with historical information can occur.

4. The proposed methodology

In this section, a completely unsupervised and partially *online* methodology for the implementation of PHM is described, which includes feature extraction, anomaly detection, data partitioning, degradation process modelling and RUL prediction [31]. In an IoT-based CPS, predictive maintenance can be performed by means of the proposed methodology. Indeed, light-weight algorithms for feature extraction, anomaly detection and data partitioning can be implemented at the edge, while more computationally expensive algorithms, like those for degradation modelling, can be performed in the cloud, only when needed. In this way, network overloading can be avoided and important information about the health status of monitored machinery can be obtained in real-time. Note that the data is collected in streaming through one only sensor, no prior knowledge is available regarding the current health status of the monitored component, and at the beginning, no historical data is used for model training. These assumptions make the methodology suitable for the implementation "from scratch", i.e., from the first available data sample. The main goal of the methodology is to recognize as soon as possible if the behavior of the monitored component is moving towards a different condition, so to determine the FT, build the degradation model and predict the RUL.

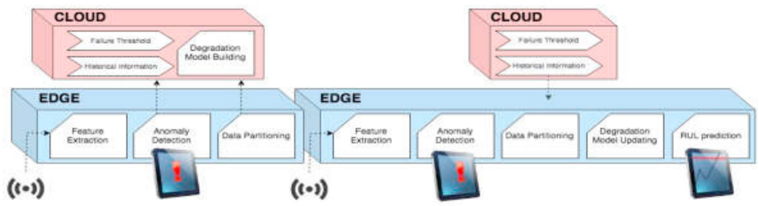

Fig. 3. The proposed methodology

As shown in Fig. 3, the proposed methodology is made of two main blocks, one *online* and the other *offline*, performed at the edge and into the cloud, respectively. In the first implementation, the *online* step only includes data collection, feature extraction, anomaly detection and data partitioning. When an anomaly is detected, i.e., a fault condition is occurring, an alarm signal is generated and the extracted features as well as the formed data partitions are sent to the cloud. Here, the degradation modelling and FT identification are conducted based on the results of data partitioning. Finally, when the proper condition is restored, the degradation model updating and RUL prediction are also included into the *online* block and performed at the edge. The integration of edge and cloud computing allows to conduct fault detection and RUL prediction almost in real-time, as they require, and to conduct the degradation modelling into the cloud, by using accurate and computationally expensive ML algorithms.

Before implementing the procedure, it is necessary to identify the time window over which features have to be extracted. For example, for rotating machinery it could correspond to the cycle time. Suppose l to be the number of data samples read during the fixed time window. Since the features corresponding to the current time window have to be extracted before next l data samples are collected, time features are the most suitable for real-time extraction. For the *online* block, the anomaly detection algorithm presented in [32] and the data partitioning algorithm presented in [33] have been slightly modified and integrated into a unique algorithm that, each time an anomaly is detected it also decides which cloud (clusters with arbitrary shapes) it belongs to. When the feature vector x_1 at the first iteration $K = 1$ is obtained, then the parameters listed in Table 1 and Table 2 are initialized as indicated in the third column. In addition, a binary variable indicating the status (not anomalous – anomalous) is initialized as not anomalous and the number of points belonging to the same status ks is set to 1. Then, for each iteration K: (1) the feature vector x_K is extracted; (2) the parameters listed in Table 1 are updated as indicated in the last column; (3) the condition (C1) is checked for anomaly assessment. If it is not satisfied, x_K is not anomalous; thus, ks is incremented by 1, x_K is assigned to the current cloud, whose parameters are updated as indicated in the last column of Table 2, and the algorithm passes to the next iteration. Otherwise, x_K is anomalous. Hence, the density at the cloud centers is computed by Eq. 1 and the condition (C2) is checked to decide whether it has to be assigned to the current cloud or should create a new cloud. If it is satisfied, a new cluster is created, whose parameters are initialized as indicated in the third column of Table 2.

Table 1. Parameters involved in the anomaly detection

Parameter	Description	Initialization	Recursive calculation
μ_K	Mean	x_1	$\mu_K = (K - 1/K)\mu_{K-1} + (1/K)x_K$
Σ_K	Scalar Product	$\|x_1\|^2$	$\Sigma_K = (K - 1/K)X_{K-1} + (1/K)\|x_K\|^2$
$\bar{D}_K$	Global Density	$D(x_1)$	$\bar{D}_K = (K - 1/K)\bar{D}_{K-1} + (1/K)D_K$
μ_D	Local Density	x_1	$\mu_D = \left((ks - 1/ks)\mu_D + (ks)D(x_k)\right)(1 - \Delta_D) + D(x_k)\Delta_D$

where

$$\mathbf{D(x_k) = 1/(1 + \|x_K - \mu_k\|^2 + \Sigma_k - \|\mu_k\|^2)} \tag{1}$$

$$\mathbf{\Delta_D = |D(x_k) - D(x_{k-1})|} \tag{2}$$

Table 2. Parameters involved in the Autonomous Data Partitioning (ADP) algorithm

Parameter	Description	Initialization	Recursive calculation
$S_{K,n}$	Number of members of cluster n	1	$S_{K,n} = S_{K-1,n} + 1$
c_K	Cloud center	x_K	$c_{K,n^*} = (S_{K-1,n^*}/S_{K,n^*})c_{K-1,n^*} + (1/S_{K,n^*})x_K$

$$\textbf{IF } \bar{D}_K < \mu_D \textbf{ for the past m iterations} \tag{C1}$$

$$\textbf{IF } D_K(x_K) > \max_{(i=1,2,\ldots,NC)}\left(D_K(c_{K-1,i})\right) \textbf{ OR } D_K(x_K) < \min_{(i=1,2,\ldots,NC)}\left(D_K(c_{K-1,i})\right) \tag{C2}$$

$$\textbf{IF } \left(d(x_K, c_{K-1,n^*}) < \frac{\gamma_K}{2}\right) \tag{C3}$$

Otherwise, the nearest cloud center to the current point is found, c_{K-1,n^*}, and the condition (C3) is checked for deciding whether the point has to be assigned to the nearest cloud or create a new data cloud. If it is satisfied, x_K will form a new cloud, whose parameters are initialized as indicated in the third column of Table 2. Otherwise, it is assigned to the nearest cloud, whose parameters are updated.

When the anomaly is detected, it should be checked if a fault condition is actually occurred. To this purpose, a machine worker could personally verify what has happened. When a fault condition is actually occurred, the extracted features together with the data clouds are sent to the cloud for further analysis and stored as the first available training set. In this methodology, the extracted features represent the input of the anomaly detection algorithm. However, since the HI for degradation modelling has to be as monotonic as possible and a dynamic smoothing could slow down the procedure, the μ_K is considered as HI and therefore transmitted and stored into the cloud.

Based on the HI values, a State-Space Model (SSM) is built, which describes the stochastic process underlying the degradation progression by using two equations. The state equation reflects the evolution of a failure, which is usually unmeasurable (latent degradation condition). The observation equation reflects the relationship between the latent degradation condition and the HI [34]; Given an SSM, the main task is to make an inference on the unobserved health state and predict the future state based on CM data. Dynamic Bayesian methods provide a unified framework for state estimation of stochastic processes [35]. Note that the FT is set as the HI value corresponding to the feature vector considered anomalous.

After that the proper condition is restored, the *online* block is implemented again. In addition, the degradation model built during the *offline* analysis is updated based on new available data. In model updating, three parameters have to be set by the user: the number of iterations, b, between two subsequent model updating, the number of HI values, f, to forecast at each model updating, the number of previous points, p, to consider for model updating. Each time the model is updated, the RUL is computed as the difference between the current time and the time at which the HI value is predicted to reach the FT. Note that b, f and p largely affect the performance of the model during the updating. In particular, if b is high, the HI could reach the FT before the next updating; however, low values of b make the algorithm slower. Similarly, high values of f allow to predict large time before when the FT will be reached, but lower the accuracy of forecasting; low values of f could not allow to predict with sufficient time the RUL (as in the third scenario). Finally, if p is high, the model updating depends on numerous previous points, making last points less influent; in contrast, a low p makes the algorithm more sensitive to the last changes in the data trend.

5. The case study

In this section, the proposed methodology is applied to a rotating component operating in a real industrial context. To demonstrate the potential of the methodology, five different conditions have been considered, obtained by conducting different tests on the equipment. Since only one of these conditions corresponds to the correct functioning, four different scenarios have been analyzed, each including the nominal condition and one of the fault conditions. The aim is to predict, for each scenario, when the fault condition will occur based on the data collected during the simulations.

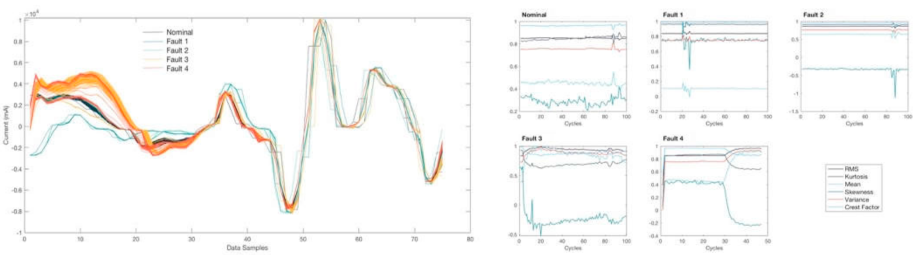

Fig. 4. (a) Raw signals of the five conditions; (b) Feature extracted for each condition.

The parameter collected from the monitored component is the current (mA) absorbed by the main actuator of a subsystem of a packaging machine during each machine cycle. It has been collected with a sampling frequency of 500

Hz. For each condition, 100 cycles are available, except for one condition, for which the test lasted only 47 cycles. Each cycle is made of 75 data samples, thus it lasts 0,15 seconds. Therefore, 15 seconds of monitoring have been conducted for each condition, except for the Fault 4, for which the monitoring is of 7,05 seconds. In Fig. 4 (a), all cycles of each condition are shown. The first step was to find the features that best describe the different conditions.

To this purpose, different supervised classification models have been trained with different sets of time features, in order to identify the set of features giving the best accuracy of classification. First, the most adopted features in the time-domain have been extracted, that are: Root Mean Square (RMS), Kurtosis, Mean, Skewness, Variance and Crest Factor, which have been normalized in the range [0,1], by dividing the features of each cycle by its maximum value assumed in all the cycles. Fig. 4 (b). shows the extracted features at each cycle, after the normalization, for each condition. Then, the typical classification algorithms (Decision tree, Support Vector Machines and Artificial Neural Networks) have been trained, and results, in terms of accuracy, are shown in Table 3. Both SVM and ANN trained with only three features (kurtosis, mean and skewness) provide best accuracy performance. Therefore, this set of features is selected as input for the *online* methodology.

Table 3 Accuracy of four classification models with two different sets of selected features

Selected features	Decision Tree	Linear SVM	Quadratic SVM	ANN
All features	94,2	98,4	98,4	98,0
Kurtosis, mean, skewness	90,2	96,6	**98,7**	**99,1**

Since the correct prediction of the RUL mostly depends on the ability to detect anomalous behaviors, only anomaly detection is first conducted, whose results are shown in Fig. 5 (a). In all cases, during the nominal behavior an erroneous anomaly value is detected. Conditions *Fault 1* and *Fault 2* have a similar behavior. Indeed, in both cases, the anomaly is detected with a latency of 0,016 seconds. Condition *Fault 3* is detected after 0,024 seconds. Finally, condition *Fault 4* is detected after 0,078 seconds.

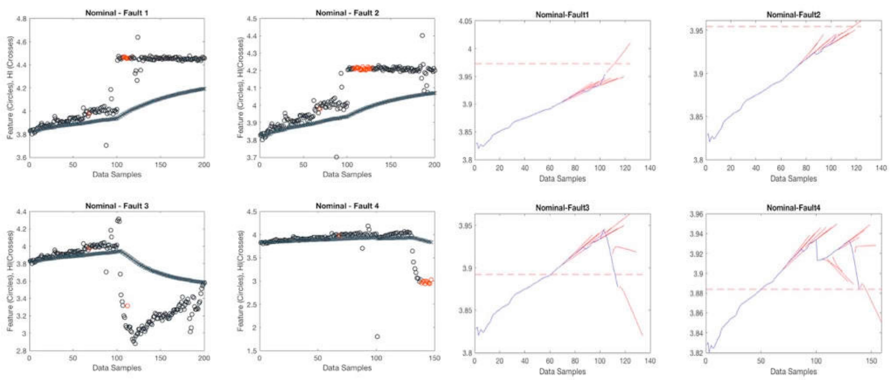

Fig. 5. (a) Anomaly Detection; (b) RUL prediction.

For each of the fault behaviors, an SSM has been built. A noniterative subspace approach is adopted for the parameter initialization and the prediction error minimization approach for the parameter estimation [36]. The HI value at which the first anomaly was detected has been adopted as the time to start the model updating, the HI values forecasting and the RUL prediction. Different values of b, f and p have been used for model updating. The best trade-off between the time necessary to update the model and the time in which the RUL is predicted was provided by: $b = 5; f = 10; p = 10$. The left side of Table 4 includes the performance of the degradation models for each condition (accuracy and the Mean Squared Error), while the right side includes the performance of the *online* RUL prediction: the iteration in which the FT is achieved, K (FT); the iteration in which the model prediction breaks due to the estimation of the RUL, K (RUL); the actual RUL, computed as difference between K (FT) and K (RUL) multiplied for the time cycle; the last column represents the RUL predicted by the model. As shown in Fig. 5 (b), even if the model is the same, three different situations occur. In the first two situations, the RUL is predicted before the FT is reached. However, the predicted RUL is greater than the actual one. In the third condition, the model does not predict any RUL, since the HI value reaches the FT before the next prediction. Finally, in the last situation, the model predicts the RUL at the same time instant as the HI value is equal to the FT, even if it predicts that the RUL has a positive value. Hence, while the anomaly detection has good performance in all scenarios, the degradation process performance depends on the particular scenario.

Table 4. Performance of the State-Space Model and RUL prediction

Condition	Accuracy (%)	MSE (%)	K (FT)	K (RUL)	Actual RUL (sec)	Detected RUL (sec)
Nominal – Fault 1	95,92	2,242e^{-6}	108	104	0,6	1,2
Nominal – Fault 2	96,03	1,957e^{-6}	108	104	0,6	2,1
Nominal – Fault 3	94,66	3,217e^{-6}	112	114	-	-
Nominal – Fault 4	92,39	6,037e^{-6}	139	139	0	0,9

6. Conclusions

In this paper, two main aspects of today's manufacturing industries are presented. The first one is related to the maintenance of equipment, and in particular to PHM for predictive maintenance. The other is related to the recent advances in IoT and CPS as enabling technologies for PHM implementation. First, concepts, tools and methods of PHM have been described, particularly focusing on the different kinds of ML and AI approaches supporting its implementation. Then, concepts, components and architectures of IoT and CPS in industrial contexts have been briefly described, particularly focusing on how these technologies can be used for prognostics purposes. In the fourth section, a *complete unsupervised and partially online methodology* for PHM in the context of IoT-based CPS has been proposed, in which real-time anomaly detection and RUL prediction are carried out at the network edge, while the degradation modelling into the cloud, where historical information could also be integrated into the analysis. Finally, the proposed methodology has been applied to a rotating component installed in a real industrial environment, for which current signals corresponding to five different conditions were available. The methodology is able to recognize the changing behavior in short time. Indeed, the fault state is recognized after few milliseconds and the number of false alarms is equal to 1. In the first and second scenario, RUL prediction has good performance as well, since the methodology breaks before the occurrence of the fault condition and the difference between the actual RUL and the predicted RUL (error) is low. Instead, the other two scenarios highlight some issues, mostly related to the latency of the anomaly detection algorithm and the monotonicity of the HI. Therefore, a trade-off between latency and false positives (dependent on the condition C1) should be established and another HI should be considered, or dynamic smoothing techniques included in the methodology. In addition, only one sensor is assumed to be installed on the component and each failure is considered separately However, when many sensors are available, *online* dimensionality reduction techniques and dynamic feature selection should be included, which each time selects the best features set. Moreover, the selection of degradation model corresponding to the incipient fault becomes an issue. Further works will be dedicated to solve these issues.

References

[1] Shin I, Lee J, Lee JY, Jung K, Kwon D, Youn BD, et al. A Framework for Prognostics and Health Management Applications toward Smart Manufacturing Systems. Int J Precis Eng Manuf Technol 2018;5:535–54. doi:10.1007/s40684-018-0055-0.
[2] Jardine AKS, Lin D, Banjevic D. A review on machinery diagnostics and prognostics implementing condition-based maintenance. Mech Syst Signal Process 2006;20:1483–510. doi:10.1016/j.ymssp.2005.09.012.
[3] Alsina EF, Cabri G, Regattieri A. A Neural Network Approach to Find The Cumulative Failure Distribution: Modeling and Experimental Evidence. Qual Reliab Eng Int 2016;32:567–79. doi:10.1002/qre.1773.
[4] Rúbio EM, Dionísio RP, Torres PMB. Industrial IoT Devices and Cyber-Physical Production Systems: Review and Use Case. Innov. Eng. Entrep., 2019, p. 292–8. doi:10.1007/978-3-319-91334-6_40.
[5] Qi Q, Tao FEI, Member S. Digital Twin and Big Data Towards Smart Manufacturing and Industry 4.0: 360 Degree Comparison. IEEE Access 2018;6:3585–93. doi:10.1109/ACCESS.2018.2793265.

[6] Lu Y. Industry 4.0: A survey on technologies, applications and open research issues. J Ind Inf Integr 2017;6:1–10. doi:10.1016/j.jii.2017.04.005.
[7] Liu Y, Peng Y, Wang B, Yao S, Liu Z, Concept A. Review on Cyber-physical Systems. IEEE/CAA J Autom Sin 2017;4:27–40. doi:10.1109/JAS.2017.7510349.
[8] Xia T, Xi L. Manufacturing paradigm-oriented PHM methodologies for cyber-physical systems. J Intell Manuf 2017:1–14. doi:10.1007/s10845-017-1342-2.
[9] Lee J, Bagheri B, Kao HA. A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems. Manuf Lett 2015;3:18–23. doi:10.1016/j.mfglet.2014.12.001.
[10] Park D, Kim S, An Y, Jung J. LiReD : A Light-Weight Real-Time Fault Detection Neural Networks. Sensors (Basel) 2018;18:2018. doi:10.3390/s18072110.
[11] Saufi MSRM, Ahmad ZA, Lim MH, Leong MS. A review on signal processing techniques for bearing diagnostics. Int J Mech Eng Technol 2017;8:327–37.
[12] Pérez-Ortiz M, Jiménez-Fernández S, Gutiérrez PA, Alexandre E, Hervás-Martínez C, Salcedo-Sanz S. A review of classification problems and algorithms in renewable energy applications. Energies 2016;9:1–27. doi:10.3390/en9080607.
[13] Martin-del-campo S, Sandin F. Online feature learning for condition monitoring of rotating machinery. Eng Appl Artif Intell 2017;64:187–96. doi:10.1016/j.engappai.2017.06.012.
[14] Calabrese F, Regattieri A, Bortolini M, Gamberi M, Francesco P. PHM-based maintenance in complex systems: reference framework, competitive approaches, experimental evidences and future challenges. Mech Syst Signal Process 2019.
[15] Lolli F, Gamberini R, Regattieri A, Balugani E, Gatos T, Gucci S. Single-hidden layer neural networks for forecasting intermittent demand. Int J Prod Econ 2017;183:116–28. doi:https://doi.org/10.1016/j.ijpe.2016.10.021.
[16] Liu R, Yang B, Zio E, Chen X. Artificial intelligence for fault diagnosis of rotating machinery: A review. Mech Syst Signal Process 2018;108:33–47. doi:10.1016/j.ymssp.2018.02.016.
[17] Datta, A, Mavroidis, C, Hosek M. A Role of Unsupervised Clustering for Intelligent Fault Diagnosis. ASME Int Mech Eng Congr Expo Proc 2007;9:687–95. doi:10.1115/IMECE2007-43492.
[18] Lolli F, Balugani E, Ishizaka A, Gamberini R, Rimini B, Regattieri A, et al. Machine learning for multi-criteria inventory classification applied to intermittent demand. Prod Plan Control 2018;7287. doi:10.1080/09537287.2018.1525506.
[19] Calabrese F, Casto A, Regattieri A, Piana F. Components monitoring and intelligent diagnosis tools for Prognostic Health Management approach. Proc. Summer Sch. Fr. Turco, 2018, p. 142–8.
[20] Shahraki AF, Yadav OP, Liao H. A review on degradation modelling and its engineering applications. Int J Performability Eng 2017;13:299–314. doi:10.1555/10.23940/ijpe.xx.xx.xx.xxx.
[21] Ahmad S, Lavin A, Purdy S, Agha Z. Unsupervised real-time anomaly detection for streaming data. Neurocomputing 2017;262:134–47. doi:10.1016/j.neucom.2017.04.070.
[22] Park CH. Anomaly Pattern Detection on Data Streams. 2018 IEEE Int Conf Big Data Smart Comput 2018:689–92. doi:10.1109/BigComp.2018.00127.
[23] Hyde R, Angelov P, MacKenzie AR. Fully online clustering of evolving data streams into arbitrarily shaped clusters. Inf Sci (Ny) 2017;382–383:96–114. doi:10.1016/j.ins.2016.12.004.
[24] Casalino G, Castellano G, Mencar C, Aldo B. Incremental adaptive semi-supervised fuzzy clustering for data stream classification. IEEE Conf. Evol. Adapt. Intell. Syst., IEEE; 2018. doi:10.1109/EAIS.2018.8397172.
[25] Lei Y, Li N, Guo L, Li N, Yan T, Lin J. Machinery health prognostics: A systematic review from data acquisition to RUL prediction. Mech Syst Signal Process 2018;104:799–834. doi:10.1016/j.ymssp.2017.11.016.
[26] Lee GY, Kim M, Quan YJ, Kim MS, Kim TJY, Yoon HS, et al. Machine health management in smart factory: A review. J Mech Sci Technol 2018;32:987–1009. doi:10.1007/s12206-018-0201-1.
[27] Yu WEI, Liang FAN, He X, Hatcher WG, Lu C, Lin JIE, et al. A Survey on the Edge Computing for the Internet of Things. IEEE Access 2018;6:6900–19. doi:10.1109/ACCESS.2017.2778504.
[28] Fei X, Shah N, Verba N, Chao K. CPS Data Streams Analytics based on Machine Learning for Cloud and Fog Computing : A Survey. Futur Gener Comput Syst 2019;90:435–50. doi:10.1016/j.future.2018.06.042.
[29] Xu H, Yu WEI, Griffith D, Golmie N. A Survey on Industrial Internet of Things : A Cyber-Physical Systems Perspective. IEEE Access 2018;6:78238–59. doi:10.1109/ACCESS.2018.2884906.
[30] Bagheri B, Yang S, Kao HA, Lee J. Cyber-physical Systems Architecture for Self-Aware Machines in Industry 4.0 Environment. IFAC-PapersOnLine 2015;48:1622–7. doi:10.1016/j.ifacol.2015.06.318.
[31] Calabrese F, Regattieri A, Botti L, Mora C, Galizia FG. Prognostic Health Management: an unsupervised and partially online methodology for fault detection and RUL prediction. Mech Syst Signal Process 2019.
[32] Costa BSJ, Angelov PP, Guedes LA. Fully unsupervised fault detection and identification based on recursive density estimation and self-evolving cloud-based classifier. Neurocomputing 2015;150:289–303. doi:10.1016/j.neucom.2014.05.086.
[33] Gu X, Angelov PP, Príncipe JC. A method for autonomous data partitioning. Inf Sci (Ny) 2018;460–461:65–82. doi:10.1016/j.ins.2018.05.030.
[34] Sun J, Zuo H, Wang W, Pecht MG. Application of a state space modeling technique to system prognostics based on a health index for condition-based maintenance. Mech Syst Signal Process 2012;28:585–96. doi:10.1016/j.ymssp.2011.09.029.
[35] Bechhoefer E, Clark S, He D. A State-Space Model for Vibration Based Prognostics. Annu. Conf. Progn. Heal. Manag. Soc., 2010.
[36] Ljung L. System Identification: Theory for the User. Second. Prentice Hall; 1999.