A new dynamic predictive maintenance framework using deep learning for failure prognostics

Khanh T.P. Nguyen*, Kamal Medjaher

LGP, ENIT, Toulouse INP, 47 Avenue d'Azereix, BP 1629 - 65016, Tarbes Cedex, France

ABSTRACT

In Prognostic Health and Management (PHM) literature, the predictive maintenance studies can be classified into two groups. The first group focuses on the prognostics step but does not consider the maintenance decisions. The second group addresses the maintenance optimization question based on the assumptions that the prognostics information or the degradation models of the system are already known. However, none of the two groups provides a complete framework (from data-driven prognostics to maintenance decisions) investigating the impact of the imperfect prognostics on maintenance decision. Therefore, this paper aims to fill this gap of literature. It presents a novel dynamic predive maintenance framework based on sensor measurements. In this framework, the prognostics step, based on the Long Short-Term Memory network, is oriented towards the requirements of operation planners. It provides the probabilities that the system can fail in different time horizons to decide the moment for preparing and performing maintenance activities. The proposed framework is validated on a real application case study. Its performance is highlighted when compared with two benchmark maintenance policies: classical periodic and ideal predicted maintenance. In addition, the impact of the imperfect prognostics information on maintenance decisions is discussed in this paper.

1. Introduction

Due to the increasing requirement of reliability, availability, maintainability and safety of systems, the traditional maintenance strategies are becoming less effective and obsolete. Beside, the revolution of Industry 4.0 provides more convenient supports for the wide development of the predictive maintenance (PdM) in practice. For example, the use of intelligent sensors provides a reliable solution for system monitoring in real time. Having this information, the manager can plan the maintenance activities more effectively to reduce machine downtimes and improve the production flow.

According to this rising practical requirement, the PdM has also received significant attention in literature over the last decade, see [1–3] for recent overviews. Generally, the predicted maintenance framework consists of two connected key parts: prediction of the system residual useful life time (RUL) and making decisions. Based on the prognostics approaches, the studies can be classified in two main groups: model based and data-driven based PdM framework.

The first PdM group relies on stochastically modeling the system degradation evolution in the discrete or continuous time. The discrete modeling can be based on the Markov process and its variants for which the transition probabilities of system states are assumed to be known

with the historical reliability data [4–7]. Beside, the continuous modeling is built on the assumption that there exits a stochastic process characterizing the degradation mechanism of the system [8–10]. Therefore, the performance of the PdM frameworks in this group depends on the prior knowledge quality of system degradation processes. From a theoretical view, it is very difficult to formalize or model a real deterioration mechanism of a complex system. From a practical view, even if a theoretical model is built, it can not be directly applied in industry where there exists various operation variables, *e.g.* loads over time can affect the validity of the proposed model. A simplification of the system real working conditions can lead to wrong maintenance decisions. To overcome these issues, the data-driven PdM framework has been developed.

In the second group, data-driven PdM framework uses sufficient data to predict the system RUL without knowing the physical nature of the degradation mechanism [11–14]. Its performance strictly depends on signal processing and feature engineering techniques. The traditional data-driven approaches [15–17] require manual processing and analysis of data by human experts and this might not be suitable for the case of big data where an automatic process is preferable [18]. Therefore, in recent studies, the deep learning (DL) methods become one of the most popular trends in data-driven diagnostic and

---

* Corresponding author.
E-mail addresses: thi-phuong-khanh.nguyen@enit.fr (K.T.P. Nguyen), kamal.medjaher@enit.fr (K. Medjaher).

prognostics that allows automatically extracting and constructing the useful information without the expertise knowledge of signal processing. For example, an effective multi-sensor health diagnostic method using a deep belief network classifier was presented in [19]. A combination between deep Boltzmann machines and a random forest was proposed in [20] to improve fault diagnostic performance for gearboxes by using acoustic and vibration signals. For recent studies, the article [21] developed an integrated hierarchical learning framework to perform both diagnostics and prognostics. In papers [22,23], the authors proposed to use the Long Short-Term Memory (LSTM) network, which is an architecture specialized in discovering the underlying time series patterns to predict the system RUL. In [24], a new deep neural network structure named Convolutional Bi-directional Long Short-Term Memory networks (CBLSTM) was designed to address raw sensory data for RUL prediction. In [25], a Restricted Boltzmann machine (RBM) was used as an unsupervised pre-training stage to learn abstract features for the LSTM input in a supervised RUL regression stage. The LSTM was also applied for the RUL prediction problem of proton exchange membrane fuel cell (PEMFC) [26,27]. In detail, the work proposed in [26] used the regular interval sampling and locally weighted scatterplot smoothing (LOESS) for data reconstruction. Then, the smoothing data is fed into a LSTM network to predict the RUL value. On the other hand, in [27], the authors developed a two-dimensional (2D) grid LSTM to optimize the prediction accuracy of the fuel cell performance degradation.

The above mentioned studies in the data-driven prognostics group only focus on the prognostics step and do not consider the maintenance decisions, which are covered separately. For example, the papers [28,29] addressed the post-prognostics issue but based on the assumption that the prognostics information of the system are already known. In [30], the authors developed a model that allows evaluating the failure probability of a furnace component and then, based on it, deciding the replacement time. However, by considering the prognostics aspect, their contribution belongs to the group of model-based approaches, i.e. the data are only used for model parameter estimation and not for model construction. Hence, it inherits some drawbacks of the model-based prognostics approaches, such as the requirement of expert knowledge to construct the model. Moreover, the developed model is application-specific and cannot be implemented in different physical systems. Finally, considering the maintenance decision formulation, the performance of the model presented in [30] strictly depends on the failure threshold definition that is not trivial in practice.

Therefore, it is necessary to propose a new framework that satisfies the following requirements: 1) the prognostics approach can be widely implemented for various systems, even complex ones; 2) the dynamic and flexible maintenance decision model should allow considering multiple options and evaluating rapidly their costs in order to make an instantaneous decision. This paper aims to address these requirements. The main contribution is to propose a new dynamic predictive maintenance framework from the point of view of operation planners. To our humble knowledge, this is the first paper that considers a complete process from data-driven prognostics to maintenance decision. It allows providing the system failure probabilities in different time windows and also making an instantaneous maintenance/inventory decision based on this prognostic information. In detail, we propose to use the LSTM network to estimate the probabilities that the system will fail in different time windows in the future. Although the LSTM network has been developed and improved in PHM literature, the previous studies [22–27] are based on a piece-wise linear (PWL) RUL target function, in which it is not trivial to define the RUL maximum value. Moreover, in these studies, the prognostics is treated as a regression problem. It only provides a predicted RUL value, whose the accuracy strictly depends on the prediction horizon, i.e. the period starting from the current time (prediction instant) to the real system failure time. Therefore, using the predicted RUL value at the first stage of the system lifetime can lead to a wrong decision. Contrary to these studies, the prognostics method proposed in this paper does not require the PWL assumption and allows providing the probabilities that the system will fall into different time intervals. As these time intervals are defined according to the requirements of the operation planner, the proposed method is expected to better adapt to practical demands. Moreover, its outputs do not depend on the period starting from the instant where the prediction is made to the real system failure time and then allows limiting the wrong decisions at the first-lifetime stage. Next, using these prognostics information, the proposed PdM framework allows providing the reasonable decisions to avoid the system failure, maximizing the system life time and reducing the inventory cost.

The remainder of paper is structured as follows. In Section 2, the Dynamic Predictive Maintenance (DPM) framework will be presented. The algorithm for evaluating the mean cost rate of the DPM framework and two benchmark policies will be developed in Section 3. In Section 4, the DPM framework will be verified on a real application case study: the turbofan engines. Its performance will be highlighted by comparing it with the classical periodic and the ideal predicted maintenance policies. The impact of the imperfect prognostics information on the decisions and, consequently, on the performance of the proposed DPM will be also investigated in the same section. Finally, the conclusion and further works will be discussed in Section 5.

2. New dynamic predictive maintenance framework

In practice, the prognostics information is usually required for a long horizon to plan different operation activities (e.g. maintenance, production or inventory, etc). Moreover, due to technological and logistic constraints, the maintenance actions cannot be performed at every time and everywhere. As an illustration, the maintenance activities for train or airplane engines cannot be realized during their journey. Hence, operation planners require the information whether the system fails in the determined time periods. For example, in the next week and the next month, what are the corresponding probabilities of the system failure? And then, how the maintenance decisions are made based on these prognostics information?

To answer the previous questions, this section aims to develop a new dynamic predictive maintenance framework that contains the total process from performing the prognostics based on multiple sensor measurements to making maintenance decisions, see Fig. 1. First, the prognostic method that provides the failure probability in different time windows will be developed in Section 2.1. Then, the decision rules taking into account spare-part-order option will be presented in Section 2.2.
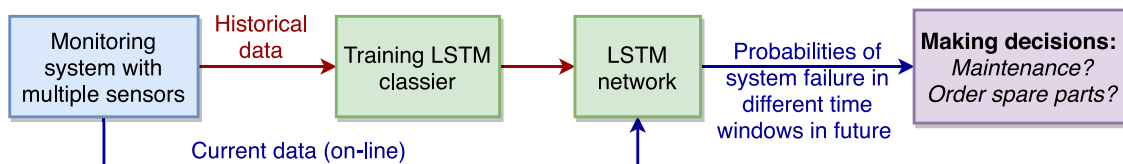

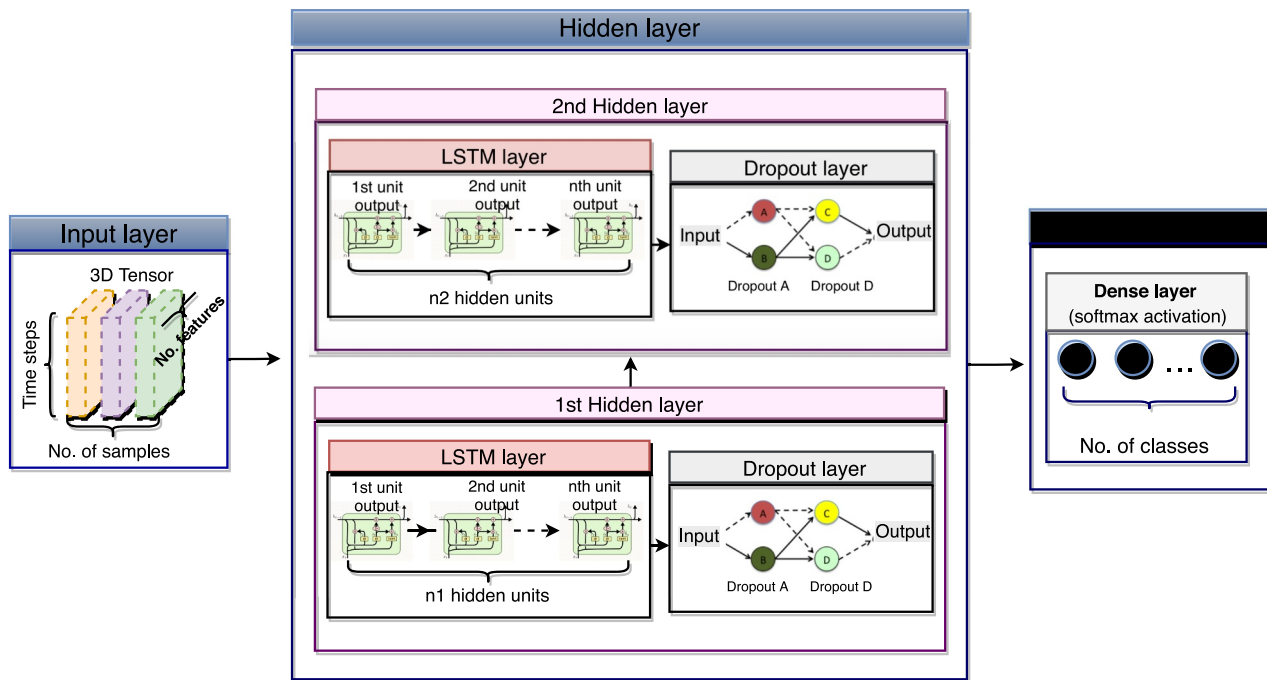**Fig. 1.** Dynamic predictive maintenance process.

Fig. 2. LSTM classifier architecture for obtaining prognostics information.

2.1. New prognostics method using LSTM classifier

For operation and maintenance schedules, the prognostics information about the possibility of the system failure in different time windows are required. These information can be evaluated using the RUL probability distribution. However, in literature, this distribution is obtained primarily based on the assumption about the degradation modeling [31–34]. In reality, especially in the case of complex systems with multiple sensor sources, it is not easy to derive the underlying degradation model. Even if theoretical models can be built, for some stochastic deterioration processes, the RUL distribution cannot be directly obtained with the analytical approaches. The RUL estimation in these cases has to be based on the simulation techniques that could be difficult to be used in real time applications [31,32]. To overcome this situation, we present in this section a new data-driven prognostic method that directly provides the probability of the system failure without prior knowledge of the failure mechanism. This method is based on the Long Short-Term Memory (LSTM) networks, one of Recurrent Neural Network (RNN) architectures, that has received increasing attention in recent prognostics studies [22–27,35]. One of the main advantages of the LSTM is the capacity of learning over long time sequences and retaining memory. Therefore, when applying LSTM for system prognostics, it allows looking back to the history of degradation processes and, consequently, well tracking the system states for RUL prediction. However, in literature, almost the existing works that use LSTM for prognostics focused on the construction of regression models to predict the RUL value. In this case, the result accuracy strictly depends on the prediction horizon and can lead to a wrong decision when using the predicted RUL value at the first stage of the system lifetime. Moreover, these studies are based on a piece-wise linear (PWL) RUL target function, in which it is not trivial to define the RUL maximum value. To remedy to these drawbacks, the methodology proposed in this paper does not require the PWL assumption and allows providing the probabilities that the system will fall into different time intervals. This can be translated as the probability that the system RUL belongs to different classes: every class corresponds to every time window.

More clearly, let us assume N components of the same type are monitored during their operation by m sensors for each component. Then, the monitoring data acquired for each component i, $i = 1, 2, ..., N$, during its life time T_n can be expressed in a matrix form: $\mathbf{X_i} = [\mathbf{x^1}, \mathbf{x^2}, ..., \mathbf{x^t}, \mathbf{x^{T_n}}]$, $\mathbf{X_i} \in \mathbb{R}^{m \times T_n}$, where $\mathbf{x^t} = [x_1^t, x_2^t, ..., x_m^t]$ is a vector of sensor measurements at time t. During the training stage, the proposed LSTM network takes the sensor measurement sequences $\mathbf{X_i}$, $i = 1, 2, ..., N$, to learn to which time window the true RUL belongs. Next, during the test stage, at time t, the constructed LSTM classifier will take the vector of sensor measurements $\mathbf{x^t}$ as an input data and outputs the probability that the RUL belongs to determined time windows.

2.1.1. Data preprocessing

Before training the LSTM network, it is necessary to reprocess the heterogeneous data from multiple sensor sources as follows:

- **Normalization:** The input data are obtained from multiple sensor sources with different ranges of values. In order to use these heterogeneous data for training the LSTM classifier, it is necessary to normalize every feature value by its mean and variance. After normalization, all features have the same range from zero to one.
- **Data labeling:** In order to perform the classification, it is necessary to label the data. The proposed methodology allows defining data labels according to the time windows in which operation planners require the failure prognostics information for scheduling maintenance and production activities. For example, if the operation planner requires the information whether the system fails in two different time windows w_0 and w_1 ($w_0 < w_1$), then the data will be labeled by three classes. The first class, noted Deg0, represents the case where the system residual life time will be greater than the time window w_1, i.e. $RUL > w_1$. The second class, noted Deg1,

characterizes the case where the system residual life time is estimated in the period $[w_0, w_1]$, *i.e.* $w_0 \leq RUL < w_1$. Finally, the third class, noted Deg2, concerns the case where the system residual life will not exceed the time window w_0, *i.e.* $RUL \leq w_0$. Considering n_c classes, the classification output is a 1D array of n_c elements. If the true RUL belongs to a given class, its corresponding element will be set to one while the remaining elements of the output array are set to zero. Note that in this paper, we will consider three classes. However the number of classes or time windows can be easily extended when necessary.

- **Formalization:** The LSTM input layer requires 3D tensor (see Fig. 2) for training the models and making predictions. Indeed, for time series data, it is necessary to format the input data as a 3D array with three dimensions: sample (n_s), time step (n_t), and feature (n_f), see [36]. To identify n_f, the different features can be extracted from a sensor output by numerous methods developed in literature, see a brief review in [37]. On the other hand, the sensor signals can be directly used to feed the LSTM input. For the case study presented in this paper, one sensor output is considered as one feature. Next, the shape of the time step axis (n_t) corresponds to the length of time sequence, which can be looked back by LSTM network when fitting models and predicting the output. Finally, considering n_s, a sample is a 2D array (n_t, n_f) that presents a time sequence of features. For instance, by monitoring the system during 100 hours with 21 sensors, and if the sensor outputs are directly used as inputs to the LSTM, then the number of features is 21 because each sensor output is considered as a feature. Next, if the data are recorded once per hour and the sequence length of LSTM is chosen to be 30, the number of samples is: $n_s = 100 - 30 + 1 = 71$. This means that the data can be translated to a 3D tensor of shape ($n_s = 71$, $n_t = 30$, $n_f = 21$). Note that the first sample is a 2D array having 30 rows (from the first time step until the 30-th time step) and 21 columns (corresponding to 21 sensor measurements).

2.1.2. LSTM Classifier architecture

A LSTM network is a recurrent neural network that allows addressing the vanishing gradient problem caused by the repeated use of recurrent weight matrix. It includes the LSTM cell blocks that contain different components called the input gate, the forget gate and the output gate. The details of LSTM cell block and its relevant mathematical functions were explained in previous studies [22,23,35]. In this paper, the LSTM network is implemented for classification, and can be called as LSTM classifier. Hence, the terms LSTM network and LSTM classifier can be merged hereafter. This section aims to describe the architecture and configuration of the proposed classifier for the system failure prognostics.

The LSTM classifier proposed in this paper is constructed by using the python deep learning library, Keras. Fig. 2 shows its architecture that has three types of layers: the input, the hidden and the output layers.

The input layer is a prototype bringing the data into the network for further processing. It requires 3D tensor with the following shape: number of samples, number of time steps and number of features.

Next, the hidden layer is the principal part of the network. It seeks to construct the relation between the input and the output. It can contain one single or multiple layers. In this paper, two LSTM layers are sequentially stacked into the hidden layer. For LSTM configuration, the number of memory units for every layer have to be provided. It is necessary to specify that the LSTM units return all of the outputs from the unrolled LSTM units through time. Then, it allows learning sequences of observations and it is well adapted to time series problem. However, it can easily over-fit training data. Therefore, the "Dropout" regularization method is used for every LSTM layer to improve the model

performance. In Keras library, the "Dropout" is defined by the probability that a unit can be excluded from the network. In detail, for each training case in a mini-batch, by dropping out units, a thinned network is sampled and its weights are evaluated [38]. Therefore, each hidden unit in a network trained with dropout can learn how to work with randomly chosen other units to create useful features. This should make each unit more robust and drive the network towards a generalization to prevent the over-fitting.

Finally, the output layer contains a feed-forward neural network that is a regular fully connected layer. This layer is used as a prototype between the network and the output. It allows transforming the 3D tensor at the hidden layer output to 1D array at the classifier output. In this paper, the classifier output is defined as a vector of 3 elements characterizing the probability that an observation belongs to 3 classes: Deg0, Deg1, and Deg2. Then, there are 3 units in the output layer and the "softmax" activation function is proposed to be used. The output layer provides the probability distribution over the three classes (Deg0, Deg1, and Deg2).

For training the LSTM classifier, it is necessary to define the objective function as "caterogical_crossentropy" that is specially used to solve the multiple mutually-exclusive class problem. This function returns the cross-entropy $H(p, q)$ between a predicted probability distribution ($p(x)$) and a true probability distribution ($q(x)$). It is given by:

$$H(p, q) = - \sum_x q(x)\log(p(x)) \tag{1}$$

Next, for the optimization algorithm, we propose to use ADAM [39], which is an extension to stochastic gradient descent. This algorithm is widely used for deep learning applications thanks to its efficient computation, the little memory requirement, and the suitability for problems of large data and/or parameters.

For evaluating the performance of the model, the metric function is defined as "categorical_accuracy". Similar to the objective function, it provides the mean accuracy rate across all predictions for multi-class classification problems. However, its results are not used when training the model.

### 2.1.3. Probability confusion matrix

This section aims to present a metric to evaluate the effectiveness of the proposed LSTM classifier. In machine learning field, the confusion matrix (M) is a popular way to evaluate the performance of the classifier algorithm. As its rows represent the true labels (TL) and its columns characterize the predicted labels (PL), the diagonal elements show the numbers of correct labels for every class. In detail, the element (M_{ij}) of this matrix represents the numbers of observations (x) whose the predicted labels are j while the true labels are i. It is given by:

$$M_{ij} = \text{Count}((PL_x = j) \cap (TL_x = i)); \tag{2}$$

In this paper, we do not only predict to which class the observation belongs, but provide the probability that it belongs to every class. Then, it is necessary to redefine the confusion matrix as the probability confusion matrix to evaluate the performance of the proposed algorithm. The element $\hat{M}_{ij}$ of this matrix is the mean probability that the predicted value of an observation x is j while the true value is i. It is given by:

$$\hat{M}_{ij} = \frac{\sum_x \mathbb{P}((PL_x = j) \cap (TL_x = i))}{\text{Count}(TL_x = i)} \tag{3}$$

where $\mathbb{P}((PL_x = j) \cap (TL_x = i))$ is the probability that the predicted label of an observation x is j while its true label is i.
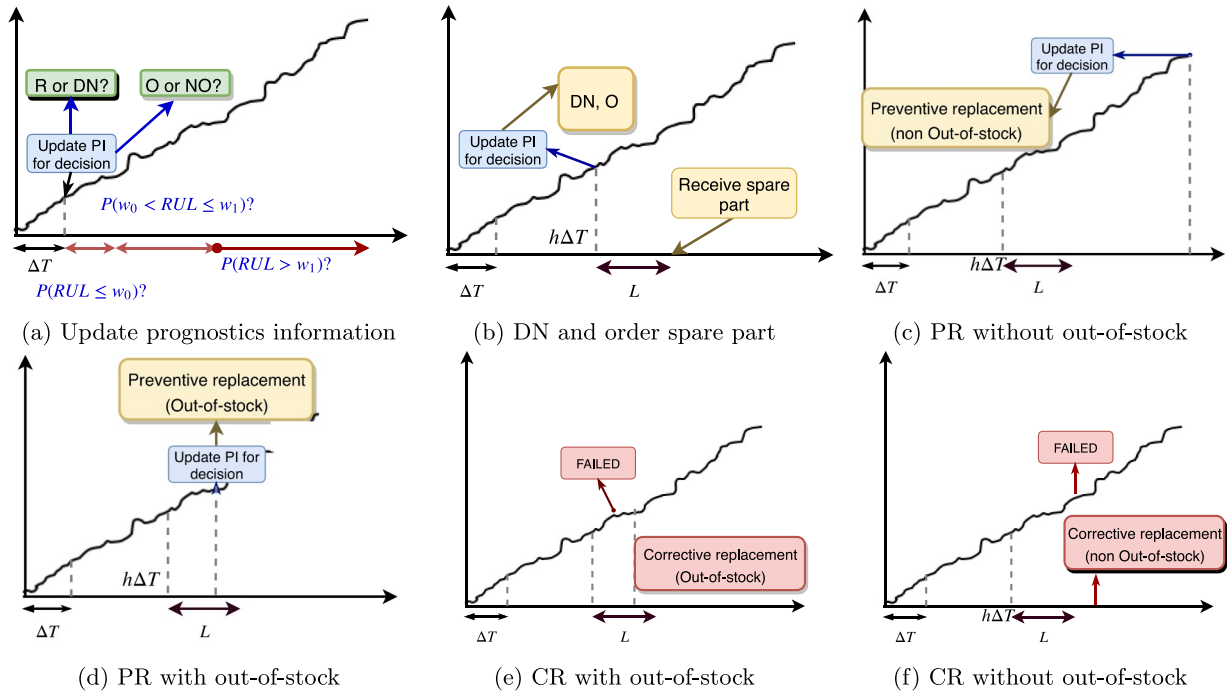
Fig. 3. Illustration of Dynamic Predictive Maintenance Policy. PR: Preventive replacement, CR: Corrective replacement, DN: Do Nothing, O: Order, NO: Non Order, PI: Prognostics Information, ΔT: Decision period or prognostics information updating period, L: Order lead time.

2.2. Decision rules based on prognostic information

In predictive maintenance, the system can be continuously monitored over time with multiple sensors but its inspection intervals are usually greater than Δ_{min}, which is the minimum achievable between two successive inspections [40]. As an illustration, the inspection interval of a train or an aircraft is normally higher than their journey time and the maintenance decision is only performed at the inspection time. The interval between two successive inspections is assumed to be constant, and noted as ΔT. The maintenance activities in this paper are assumed to be perfect. They can be realized by corrective or preventive replacements of the failed component by its spare part. On the other hand, to minimize the holding inventory cost, the spare part for a such replacement is only ordered if necessary. The lead time, L, for an order is assumed to be greater than the decision period ($L > \Delta T$). The corrective or preventive replacement can be performed when the spare part is unavailable, but it is subject to add the extra out-of-stock cost. Then, the maintenance or spare-part-order decisions must be carefully considered based on the prognostics information.

In detail, at the beginning of the inspection period (h-th for example), if the system still works, its failure probabilities in different future time windows are updated based on the monitoring data. As the maintenance is only performed at the inspection time, it is necessary to know the probability that the system will fail in the next period, $P(RUL \leq \Delta T)$. Hence, the first time window (w_0) is equal to the inspection interval ($w_0 = \Delta T$). On the other hand, to make a spare-part order at the right moment, the second time window must be relevant to the lead time. It is defined as follows:

$$w_1 = \left[\frac{L}{\Delta T}\right]^+ \cdot \Delta T; \quad \text{where } [x]^+ \text{ is the upper integer of a}$$

$$\text{real number } x. \tag{4}$$

Based on the prognostics information ($P(RUL \leq w_0)$, $P(w_0 < RUL \leq w_1)$, and $P(RUL > w_1)$), the operation planner will decide:

1. to order (O) or no-order (NO) the spare part (if the order is not done yet). The decision is based on the comparison between the cost of

two options (O) and (NO). The option having the lower cost will be chosen. A cost of NO-option is defined by:

$$NO = C_{os} \cdot P(w_1 < RUL \leq w_1 + \Delta T) \tag{5}$$

The NO-option cost is not a real charged cost. It is only a damage estimation of the wrong decision when we do not order at the precise moment. In fact, if the spare part is ordered at that moment, we can avoid the out-of-stock cost when a such replacement will be realized in the time windows (w_1, $w_1 + \Delta T$). Hence, in the case of no-order, the NO-option cost is equal to the product of $P(w_1 < RUL \leq w_1 + \Delta T)$ and the out-of-stock cost (C_{os}). Contrarily, the O-option cost is linked to the estimated holding inventory cost in future and can be evaluated by:

$$O = C_i \int_L^\infty (x - L) \cdot f_{RUL}(x) dx; \tag{6}$$

where C_i is the inventory cost per unit time of a spare part, L is the lead time and $f_{RUL}(x)$ is the RUL probability density function (pdf). As we discussed in the previous section, in reality it is not easy to obtain the RUL pdf, especially in the case of a complex system. On the other hand, for the discrete decision process, when the decision is only made at the beginning of the inspection period (ΔT), the O-option can be evaluated by the following equation:

$$O = C_i \sum_{i=[w_1/\Delta T]^+}^{\infty} P((i-1)\Delta T < RUL \leq i\Delta T) \cdot (i\Delta T - w_1) \tag{7}$$

Note that the probability $P((i-1)\Delta T < RUL \leq i\Delta T)$ can be obtained by the LSTM prognostic method proposed in this paper (see Section 2.1) when considering the multi-class classification problem with numerous time windows $[(i-1)\Delta T, i\Delta T]$. From another standpoint, operation planners usually prefer simple mathematical formulas that allows making instantaneous decision with an acceptable error. Therefore, we propose to simplify the Eq. 7 by the following equation:

$$O = P(RUL > w_1) \cdot C_i \left[\frac{\max(\bar{T}_F - h\Delta T - w_1, \quad \Delta T)}{\Delta T}\right]^+ \Delta T \tag{8}$$
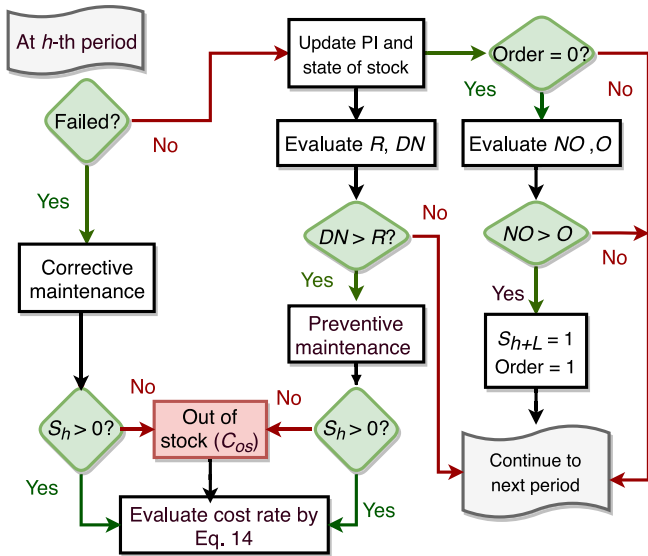
Fig. 4. Algorithm for evaluating the DPM cost rate for a life cycle.

where h is the current period, and $\bar{T}_F$ is the mean time to failure of the system. In practice, $\bar{T}_F$ can be empirically obtained based on the historical reliability data of the system. Note that Eq. 8 is only an approximation of the holding inventory cost to make the decision rule simpler to be applied in reality. It is used for the real case study presented in this paper. Consequently, it is shown that the cost difference when using this expression is acceptable compared with the ideal case.

2. to replace (R) or do-nothing (DN). The decision is based on the comparison between the cost rate (cost part unit time) of two options (R) and (DN): the option having the lower cost rate will be chosen. Their cost rates are given by:

$$R = \frac{C_p + C_{os} \cdot \delta(S_h = 0)}{h\Delta T};$$
(9)

$$DN = \frac{P(RUL \le w_0) \cdot (C_c + C_i \cdot \delta(S_h = 1)\Delta T + C_{os} \cdot \delta(S_{h+1} = 0))}{(h+1)\Delta T}$$
(10)

In detail, given the current period h, if one decides to replace the degraded component, she/he has to pay the preventive maintenance cost (C_p) and the out-of-stock cost (C_{os}) linked to a spare part unavailability ($S_h = 0$). Then, the sum of these costs will be divided by the actual life time of the system ($h\Delta T$) to evaluate the cost rate. Contrarily, if the component is not preventively replaced, there exists the risk that it will fail in the next period ($P(RUL \le w_0)$, with $w_0 = \Delta T$). In this case, the corrective maintenance will be performed with the cost C_c, which also includes the downtime cost, at the beginning of $h + 1$ period. Hence, its life time is equal to $(h + 1)\Delta T$. In addition, the holding inventory cost and the out-of-stock cost can be taken into account.

Fig. 3 illustrates the possible progress of the presented dynamic predictive maintenance policy. At the beginning of the inspection period, the probabilities that the system will fail in three time windows ($P(RUL \le w_0)$, $P(w_0 < RUL \le w_1)$, and $P(RUL > w_1)$) are updated. Based on these information, the appropriate options (R or DN, O or NO) will be chosen (see Sub-Fig. 3.(a)). Then, if at the moment $h\Delta T$ we decide to do nothing and order the spare part, the system continues to work in the next period and the spare part is received after the lead time

L (see Sub-Fig. 3.(b)). For the next inspection period, as the order is done at $h\Delta T$, only the R and DN options are evaluated. If the R option is chosen after the moment $h\Delta T + L$, then the preventive replacement (PR) will be realized without the out-of-stock issue (see Sub-Fig. 3.(c)). Contrarily, the out-of-stock cost (C_{os}) have to be added in the maintenance cost (see Sub-Fig. 3.(c)). In the case where the system is failed before a preventive maintenance, a corrective replacement (CR) will be performed at the beginning of the next period. If the spare part is unavailable, the out-of-stock problem will occur (see Sub-Fig. 3.(d)) and vice-versa (see Sub-Fig. 3.(e)).

3. Performance evaluation of the proposed predictive maintenance framework

In order to highlight the performance of the proposed dynamic predictive maintenance framework (DPM), this section aims to present the algorithm for evaluating the DPM average cost rate. This result will be compared with the cost rates of the two following policies:

- **Periodic maintenance policy (PeM)** that is based on the historical reliability data of the system. In detail, using the historical reliability data, the mean time to failure $\bar{T}_F$ of the system is evaluated. Then, the periodic preventive maintenance with the cost C_p will be performed at the moment T_R:

$$T_R = \left[\frac{\bar{T}_F}{\Delta T}\right]^+ \cdot \Delta T$$
(11)

As the maintenance activities have been planned in advance, the spare parts will be available at this moment. Contrarily, the spare part is unavailable for a corrective maintenance if the system is failed before the moment T_R. In this case, the corrective maintenance cost C_c and the out-of-stock cost C_{os} will be added. Hence, the average cost rate of the Policy 2 is given by:

$$\bar{CR}_{PeM} = \frac{1}{n}\sum_{i=1}^{n}\frac{Cp}{T_R}\cdot\delta(T_{F_i} > T_R) + \frac{(C_{os} + C_c)}{[T_{F_i}/\Delta T]^+ \cdot \Delta T}\delta(T_{F_i} < T_R)$$
(12)

where n is the number of system life cycles, T_{F_i} is the failure time of the i-th life cycle and $\delta(x)$ is a direct function: $\delta(x) = 1$ when x is true.

- **Ideal predicted maintenance policy (IPM)** that is based on the hypothesis of the perfect predicted failure time. In this case, at the inspection time, we assume that the residual life time is correctly determined. Then, the decisions based on this perfect information will minimize the cost rate value, which is given by:

$$\bar{CR}_{IPM} = \frac{1}{n}\sum_{i=1}^{n}\frac{Cp}{[(T_{F_i} - 1)/\Delta T]^- \cdot \Delta T}$$
(13)

where $[x]^-$ is the lower integer of x. Thanks to the perfect prognostic information, the preventive maintenance will be performed with an available spare part at the moment $[(T_{F_i} - 1)/\Delta T]^- \cdot \Delta T$ which is the inspection time before the failure. Hence, the cost rate of i-th life cycle equals to the ratio between the preventive maintenance cost C_p and the life cycle duration.

The algorithm for evaluating the cost rate of **the proposed DPM policy** is presented in Fig. 4. At the beginning of the i-th life cycle, the boolean variable *Order* and the state of the stock are set to zero. Without loss of generality, considering the h-th inspection period for example, if the system is failed, the corrective replacement is performed. Otherwise, if the system still works but the R option is chosen, then the preventive replacement is realized. The proposed DPM cost rate is evaluated by Eq. 14.

$$CR_i = \begin{cases} \dfrac{C_c + C_{os} \cdot \delta(S_h = 0)}{h \cdot \Delta T}; \text{when the system is failed and the corrective replacement is performed.} \\[2em] \dfrac{C_p + C_{os} \cdot \delta(S_h = 0)}{h \cdot \Delta T}; \text{when the system still works but the preventive replacement is performed.} \end{cases} \tag{14}$$

Considering Eq. 14, if the system is failed, the corrective maintenance will be performed with an available spare part when $S_h > 0$. Contrarily, the out-of-stock cost (C_{os}) will be added if $S_h = 0$. After the corrective maintenance, the i-th life cycle is ended and a new life cycle $i + 1$ will be started.

However, when the system still works, the inventory state (characterized by S_h and a boolean variable $Order$) and the prognostics information PI (obtained by the LSTM classification, see sub-Section 2.1) will be updated. Based on these information, two decision branches will be exploited.

On the first branch, the corresponding cost rates of the options R and DN are evaluated by Eq. 10. If $DN > R$, a preventive maintenance will be performed with the available spare part when $S_h > 0$ while the C_{os} will be added when $S_h = 0$. Then, the i-th life cycle is ended and the corresponding cost rate is calculated by Eq. 14. Otherwise, if $DN > R$, the system continues to work until the next inspection time.

On the second branch, if $Order = 0$, the corresponding cost rates of the options NO and O are evaluated by Eq. 5 and Eq. 8. Then, if $NO > O$, the boolean variable $Order$ and the state of stock after a lead time will be set to one and we move to the next period. In the cases where $Order = 1$ or the NO option cost is less than the O-option cost, we directly consider the next period.

After evaluating the cost rate of n life cycles, the average cost rate of the DPM policy is given by:

$$\bar{CR}_{DPM} = \frac{1}{n} \sum_i^n CR_i \tag{15}$$

4. Real application case study

In this section, the proposed DPM is verified on the benchmarking data set: Turbofan Engine Degradation Simulation provided by NASA Ames Prognostics Data Repository. This data set is widely used in PHM field, see [41] for a review of the prognostic studies using it. It is generated by C-MAPSS tool that simulates various degradation scenarios of the fleet of engines of the same type. At the beginning of each scenarios, the engine is normally operating. It is degraded until a failure in the training set. In the test set, the degradation process ends some time prior to system failure. Both of training and test sets consist of 26 columns that describe the characteristics of the engine units. The first and second column respectively represent the ID and the degradation

Table 1
Characteristics of four test sets.

Test set	Trajectory number	Condition number	Fault mode number	Min	Max	Mean	Median
FD001	100	1	1	31	303	130.96	133.5
FD002	259	6	1	21	367	131.24	132
FD003	100	1	2	38	475	165.96	148
FD004	249	6	2	19	486	166.19	153

Table 2
Configuration parameters of the LSTM classifier.

Epochs	Dropout	1st LSTM units	2nd LSTM units
50	0.2	100	50

time steps for every engine. The next three columns characterize the operation modes of the engines while the final 21 columns correspond to the outputs of 21 sensors.

The C-MAPSS data set includes 4 subsets: FD001, FD002, FD003 and FD004, which correspond to 4 different cases combining different operating conditions and fault modes. The subsets FD001 and FD003 are subject to a single operating condition while the subsets FD002 and FD004 present six operating conditions. Moreover, there exists only one fault mode in the subsets FD001 and FD002 while the subsets FD003 and FD004 are more complex due to two failure modes. Table 1 summarizes the characteristics of the corresponding test sets. In this table, the statistical indicators (Min, Max, Mean, Median) of the recorded data length of different turbofan engines are evaluated for every test set. It can be seen that the recorded data length are not uniform across each set. For example, in the test set FD001, there are 100 trajectories. Among them, the minimum length and the maximum length are respectively 31 and 303. The mean and median values of the recorded data lengths for this set are then respectively 130.96 and 133.5.

4.1. Discussion of prognostics accuracy

To construct the LSTM classifier presented in Section 2.1, the python deep learning library Keras was used. The configuration parameters are summarized in Table 2. For the formalization of the LSTM data input, it is necessary to define the sequence length (n_t). The value of this parameter should be smaller than that one of the recorded data length, *i.e.* the length of the recorded trajectories. In addition, the greater value of n_t is, the higher capacity of looking back in the history data of the LSTM will be. However, this will lead to an increase of the training time. Therefore, for every test set in the case study, it is preferable to have the values of n_t enough large to benefit from the history information, but must be smaller than the minimum length of the recorded trajectories. Note that the minimum recorded lengths are 31 for FD001, 21 for FD002, 38 for FD003 and 19 for FD004, see Table 1. Then, the corresponding values n_t are respectively chosen as follows: 30 for FD001, 20 for FD002, 30 for FD003, and 10 for FD004.

As mentioned in the previous section, the prognostic method proposed in this paper provides the probability that the system will fail in the different time windows in the future instead of a precise RUL value. Therefore, we cannot use the common evaluation criteria such as PHM08 score, MAE, MAPS, MSE to evaluate the performance of the proposed method and compare it with the published studies in literature [42–44]. Instead, we propose to use the confusion probability matrix presented in Section 2.1.3 to evaluate the accuracy of the prognostics information. It also provides the benchmark results for further studies for whom wants to focus on the improvement of the prognostic algorithms.

As an illustration, let assume that the managers are interested on the probability that the systems will fail in three different time windows: ($RUL \leq w_0$), ($w_0 < RUL \leq w_1$), and ($RUL > w_1$), where $w_0 = 10$ and $w_1 = 20$. Fig. 5 presents the mean probability confusion matrix on the test sets with three respective classes: Deg2 ($RUL \leq w_0$), Deg1 ($w_0 < RUL \leq w_1$), and Deg0 ($RUL > w_1$). Note that a high RUL value corresponds to a low degradation. The following results can then be distinguished:

- when the system belongs to the state Deg2, the mean confusion probabilities that the system belongs to the Deg0 are very low for all the test sets. The worst case is 2.9% for the FD004 test set when
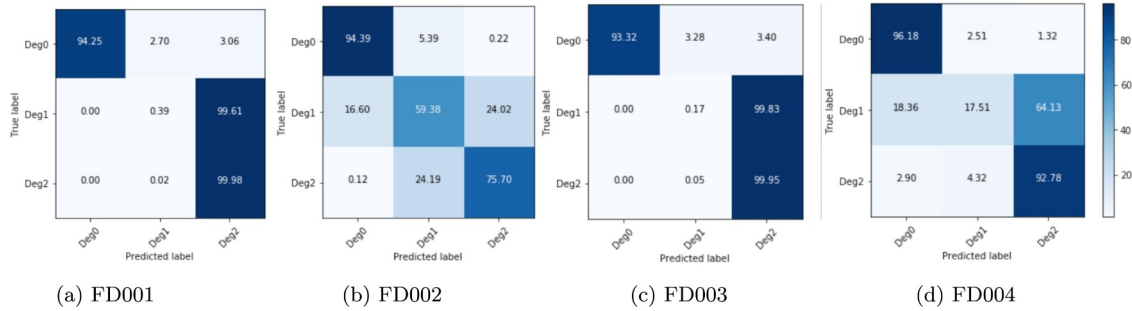
Fig. 5. Probability confusion matrix (%) on test sets with $w_0 = 10$, $w_1 = 20$. (Deg2: $RUL \leq w_0$, Deg1: $w_0 < RUL \leq w_1$, Deg0: $RUL > w_1$). Note: a dark color corresponds to a high probability value.
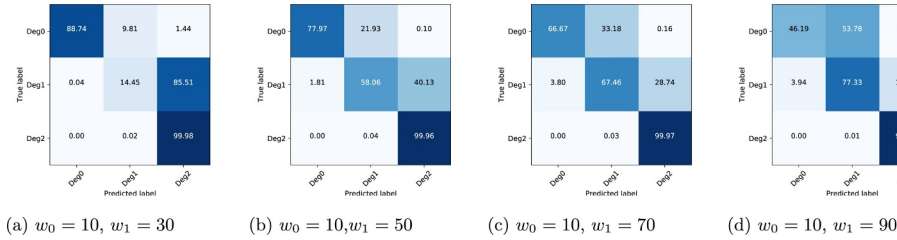
Fig. 6. Probability confusion matrix (%) on test set FD001 with different time window w_1. (Deg0: $RUL \leq w_0$, Deg1: $w_0 < RUL \leq w_1$, Deg2: $RUL > w_1$). Note: a dark color corresponds to a high probability value. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 3
Dynamic predictive maintenance scenarios, $C_p = 100$, $C_c = 500$, $C_i = 0.1$, $C_{os} = 10$ (Deg2: $RUL \leq 10$, Deg1: $10 < RUL \leq 20$, Deg0: $RUL > 20$).

Life time	True RUL	Deg0 (%)	Deg1 (%)	Deg2 (%)	Order	Stock	Maintenance
Engine ID82 ($T_f = 214$)							
170	44	99.99	0.01	0	0	0	0
180	34	99.91	0.09	0	0	0	0
190	24	80.55	19.46	0.09	1	0	0
200	14	0.02	76.96	23.02	1	0	0
210	4	0	0	100	1	1	1
Engine ID81 ($T_f = 200$)							
180	60	99.78	0.21	0.01	0	0	0
190	50	99.61	0.37	0.02	0	0	0
200	40	78.86	21.02	0.11	1	0	0
210	30	34.63	64.15	0.22	1	0	0
220	20	0.37	93.62	6.01	1	1	0
230	10	0.36	76.22	23.42	1	1	1
Engine ID83 ($T_f = 293$)							
250	43	99.95	0.05	0	0	0	0
260	33	99.9	0.09	0.01	0	0	0
270	23	4.06	95.76	0.18	1	0	0
280	13	0.05	43.05	56.9	1	0	1

considering 6 operating conditions and 2 failure modes. In the cases of the FD001 and FD003 test sets, when only one operating condition is investigated, these probabilities are negligible and are approximately equal to zero.

- when the system is in the state Deg0, the mean confusion probability of the state Deg2 is also non-significant for all the test sets. The maximum value is 3.4% in the case of FD003 test set where two failure modes and one operating condition is considered.
- when the system belongs to the state Deg1, its predicted state is normally the state Deg2, especially in the case of FD001 and FD003 test sets. However, this phenomenon can be explained by the fact that two time windows w_0 and w_1 are so close together. Hence, the characteristics of the systems that belong to the state Deg1 in the test sets are similar to the ones of the state Deg2 in the training sets. Next, we will consider the change of the confusion probability matrix in different lengths of the time window w_1.

Using the test set FD001, Fig. 6 shows the impact of time window w_1 on the probability confusion matrix. It can be seen that the confusion probabilities between the state Deg0 and the state Deg2 are negligible

for all 4 values of w_1. On the other hand, the confusion probability that the predicted state is Deg2 when the true state is Deg1 is decreasing in w_1. Contrarily, the confusion probabilities between the states Deg0 and Deg1 are increasing in w_1. The confusion probabilities between the predicted classes can be reduced by further studies on the optimization of the LSTM architecture or on the configuration parameters. However, as mentioned in the previous section, this paper does not focus specifically on the development of the prognostics method. We are interested on the use of the prognostics information for the predicted maintenance. Therefore, instead of the improvement of the prognostics, we will consider in the next section the impact of these confusions on the performance of the proposed predicted maintenance policy.

4.2. Dynamic predictive maintenance framework

Next, the proposed DPM framework will be verified on the first turbofan engine data set, FD001. In order to evaluate and compare the performance of three maintenance policies (DPM, PeM and IPM) presented in Section 3, the total information of the engines states during all their life cycles (from the beginning to the failure) are necessary.

(a) Current time $t = 180$

(b) Current time $t = 190$

(c) Current time $t = 200$
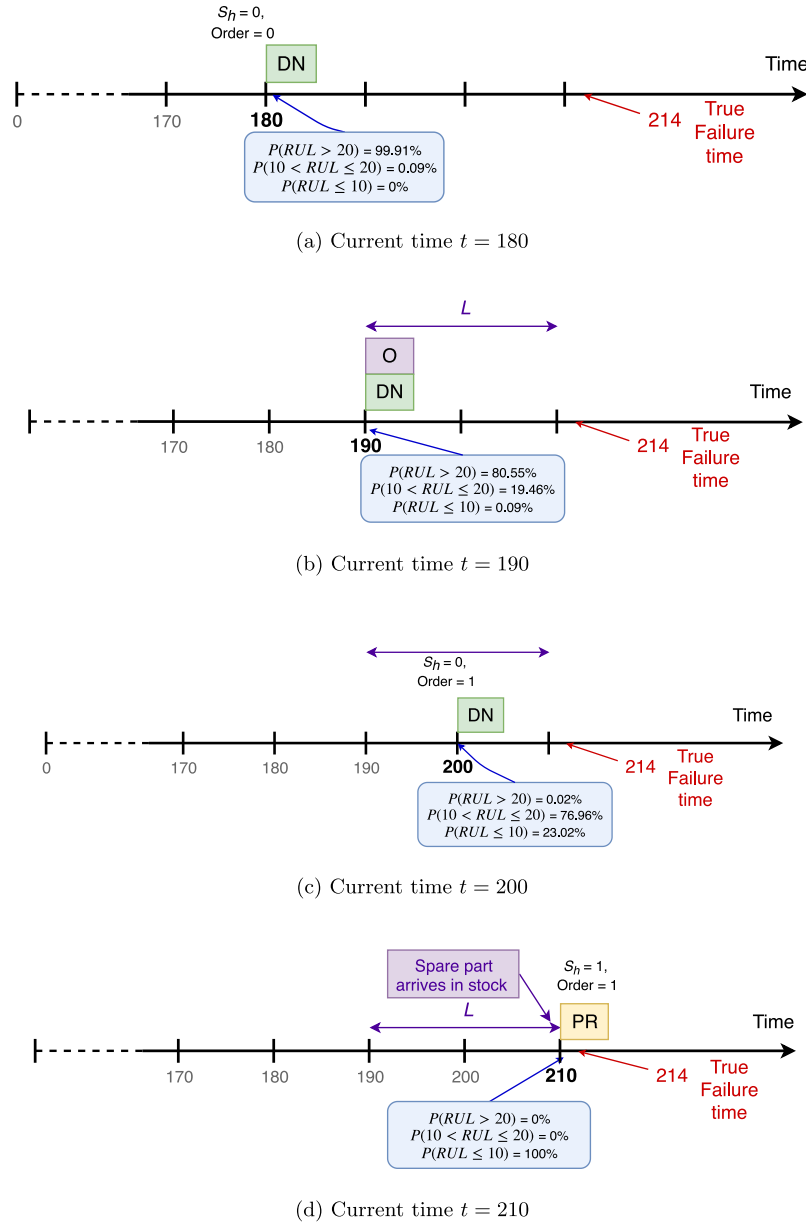

(d) Current time $t = 210$

Fig. 7. Illustration of the maintenance/inventory decisions at each time step for the engine ID82.

Table 4
Illustration of the flexibility of maintenance decisions.

ID	T_f	O time	No. Stock period	R time	OOS
Case A: $C_p = 100$, $C_c = 500$, $C_i = 1$, $C_{os} = 10$					
81	240	200	1	230	0
83	293	270	0	280	1
84	267	250	0	260	1
Case B: $C_p = 100$, $C_c = 200$, $C_i = 0.1$, $C_{os} = 20$					
81	240	210	1	240	0
83	293	270	0	290	0
84	267	210	3	260	0

Hence, in this section, the FD001 training set will be divided into two parts. The first part, including 80 first trajectories, is used for training the LSTM while the second part, consisting of 20 remaining trajectories, is used for evaluating the performance of the DPM framework.
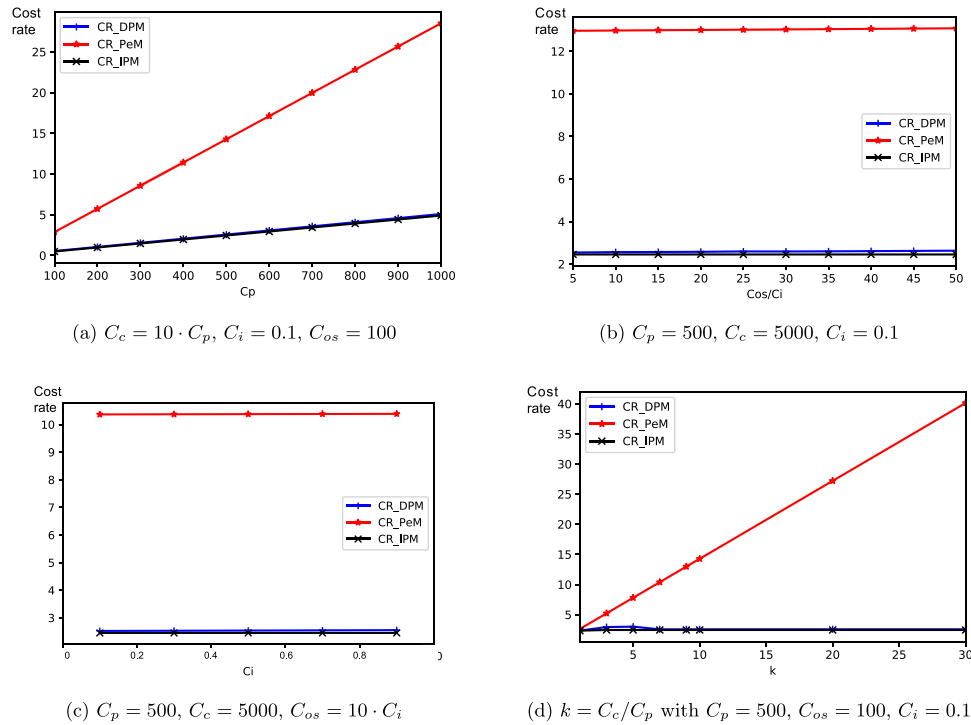
Given $w_0 = \Delta T = 10$, $w_1 = L = 20$, at every inspection time $h \cdot \Delta T$, where $h \in [1, 2, 3, ...]$, and based on the sensor measurements, the

LSTM classifier provides the prognostics information of the engine for making maintenance decisions.

4.2.1. Optimal decisions of DPM

Table 3 shows some last inspection periods of the life cycle of three engines (ID 81, 82, 83) to illustrate how the DPM works. The two first columns present the lifetime and the true RUL of each engine, while the three next columns show respectively the probabilities that an engine belongs to the three classes: Deg0 ($RUL > 20$), Deg1 ($10 < RUL \leq 20$) and Deg2 ($RUL < 10$). The remaining columns (Order, Stock, and Maintenance) present the corresponding boolean variables which characterize the *Order*, the stock and the maintenance states, respectively.

Considering the engine ID82, one can see that at $t = 180$, the values in the three columns Order, Stock and Maintenance equal to zeros. That means before $t = 180$, the optimal decision is to do nothing (DN) and no order (NO). Fig. 7 illustrates the maintenance/inventory decisions for the engine ID82. From figure 7(a), one can see that, at $t = 180$, *i.e.* at the beginning of the h-th decision period ($h = 18$), the engine still works, its

(a) $C_c = 10 \cdot C_p$, $C_i = 0.1$, $C_{os} = 100$

(b) $C_p = 500$, $C_c = 5000$, $C_i = 0.1$

(c) $C_p = 500$, $C_c = 5000$, $C_{os} = 10 \cdot C_i$

(d) $k = C_c/C_p$ with $C_p = 500$, $C_{os} = 100$, $C_i = 0.1$

Fig. 8. Comparison of the mean cost rates (CR) of the maintenance polices: DPM (dynamic predictive maintenance), PeM (periodic maintenance) and IPM (ideal predictive).

sensor measurements are fed into the LSTM classifier to update the prognostics information, *e.g.* $P(RUL > 20)$, $P(10 < RUL \le 20)$, $P(RUL > 20)$. As the probability $P(RUL > 20)$ is high ($P(RUL > 20) = 99.91\%$), the proposed DPM provides the optimal option is to do nothing. Next, at the 19-th decision period (sub-Fig. 7(b)), the LSTM classifier gives $P(RUL > 20) = 0.02\%$, $P(10 < RUL \le 20) = 76.96\%$, and $P(RUL > 20) = 23.02\%$. Based on these information, the DPM policy indicates that the optimal decision is to order spare part and do nothing. The ordered spare part will be delivered after the lead time $L = 20$. Similarly, at the 20-th decision period, (sub-Fig. 7(c)), the optimal decision is to do nothing. Finally, at the 21-th decision period (sub-Fig. 7(d)), the ordered spare part was delivered. The LSTM classifier indicates that the probability of the engine failure within this period is 100%. Therefore, the optimal decision is to preventively replace the engine.

For the engine ID81, when its true RUL is higher than 50, the probability that it belongs to the state Deg0 is high ($P(Deg0) = 99.61\%$). Then, the DPM indicates that the optimal option is to do nothing (Table 3). When the true RUL of this engine equals to 40, the spare part order is done because of a non-negligible probability of the warning class ($P(Deg1) = 21.02\%$). This means that the boolean variable *Order* is set to one from this moment. After a lead time of 2 periods, the spare part is available for maintenance. However, it is kept in stock until the next period. When the true RUL is equal to 10, the preventive maintenance is realized.

For the engine ID83, because of a small confusion of the prognostic information at the life time 280, that is $P(Deg2)$ is higher than $P(Deg1)$ (while the true RUL is 13), the maintenance action is performed in order to avoid the failure. At this moment, the spare part is not available, which means that the out-of-stock cost must be charged (Table 3).

4.2.2. Flexibility of DPM decisions

One of the advantages of the proposed DPM framework is that the decisions are quickly and simply made based on the prognostics information. It allows evaluating the cost of options at the decision time and therefore, brings more flexibility for decisions in order to well adapt to the cost change. For an illustration, Table 4 shows the decision flexibility according to the change of costs. The two first columns present the ID and the failure time of engines. The third column (O time) shows the order time while the fifth column (R time) describes the replacement time. The fourth column (No.Stock period) and the sixth column (OOS) respectively indicate the number of periods for holding a spare part and the out-of-stock state. By comparing the results of Case A and Case B, we find that:

- The replacement times of the engine ID81 and ID83 are postponed when the corrective maintenance cost (C_c) is decreasing.
- The order time of the engine TD84 is accelerated when the ratio between the inventory cost (C_i) and the out-of-stock cost (C_{os}) is increasing. In this case, the out-of-stock state is restricted.

4.3. Performance of the DPM framework

In this section, the performance of the proposed DPM is highlighted when compared with the classical periodic maintenance policy (PeM). We will also investigate whether an imperfect prognostics has a significant negative impact or not on the DPM cost rate through a comparison with the ideal case (perfect prognostics), called the ideal predictive maintenance (IPM).

Fig. 8 presents the mean cost rates (CR) when deploying three maintenance polices (DPM, PeM and IPM) for 20 engines (ID from 81 to 100) given in the FD001 training data set with different combinations of costs. For all cases, the performance of the DPM is highlighted: its cost rate is significantly lower than the one of the PeM and is also very close to the ideal case IPM with perfect prognostics. Note that the perfect prognostics is only an ideal hypothesis that can not be attained in reality. These results show that even if the proposed prognostic method (LSTM classification) is not perfect, the predictive maintenance framework works well. It allows significantly reducing the operation cost rates and almost reaching the ideal values.

In detail, considering the sub-Fig. 8.(a), when the corrective maintenance cost (C_c) and the preventive maintenance cost (C_p) increase 10
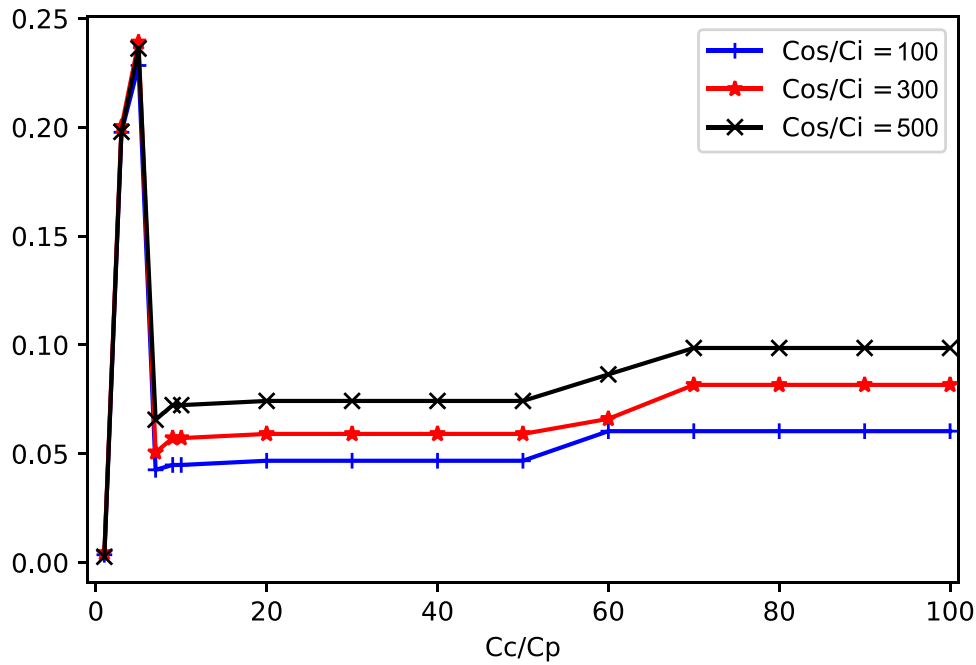
Fig. 9. Relative difference between the DPM and IPM mean cost rate, with $C_i = 0.1$, $C_p = 500$.

times, the mean cost rate of the PeM policy significantly rises from 2.87 to 28.51 while the one of the DPM slightly increases from 0.55 to 5.09. Moreover, the growth of the DPM is close to the progress of the IPM cost rate which increases from 0.49 to 4.91. However, although the DPM policy allows significantly reducing the cost rate when compared to the PeM policy, its cost rate increases proportionally to the preventive maintenance cost. Now, when the predictive maintenance cost C_p is constant, $C_p = 500$, and the corrective maintenance cost C_c increases from 500 to 15,000 (sub-Fig. 8.(d)), the benefit of the proposed DPM policy becomes interesting. In this case, the mean cost rate of the DPM policy is almost constant and close to the ideal cost rate while the mean cost rate of the PeM policy rapidly increases from 2.61 to 40.1. These results show that thanks to prognostics information, the DPM policy provides the correct decisions and helps avoiding corrective maintenance interventions and, therefore, its cost rate is not sensitive to the change of C_c. Note that when $C_c = C_p$, *i.e.* $k = C_c/C_p = 1$, it is not necessary to implement a preventive maintenance since the mean cost rate of all the three policies will converge to the one of the corrective maintenance. Next, sub-Figs. 8.(b) and (c) show that the cost rates of all three maintenance policies are not sensitive to the change of the out-of-stock cost (C_{os}) and to the inventory cost (C_i). This can be explained by the fact that the values of C_i and C_{os} are insignificant when compared to the maintenance costs. However, thanks to prognostic information, the DPM policy provided the reasonable maintenance/inventory decisions that help reducing the cost rate about 6 times when compared to the one of the classical periodic policy.

In the following, we will consider the impact of the imperfect prognostics information on the DPM. To do that, the focus will be put on the difference between the performance of the proposed DPM and the one of the ideal case (IPM) by evaluating the relative difference (*RD*) defined as:

$$RD = \frac{CR_{DPM} - CR_{IPM}}{CR_{IPM}}; \qquad RD \in [0, \infty]$$

Fig. 9 presents the relative difference between the DPM and the IPM mean cost rates in the ratio k, where $k = C_c/C_p$ with different values of C_{os}/Ci (given $C_p = 500$ and $C_i = 0.1$). One can note that when k increases from 1 to 5, the *RD* rapidly increases from 0 to 24% of the IPM mean cost rate. This growth of *RD* can be explained by wrong decisions due to

an imperfect prognostics information. For example, at h-th period, because the probability $P(RUL \leq w_0)$ and the ratio k between C_c and C_p are not significant, a do-nothing *DN* option can be chosen. Unfortunately, the system is failed in the next period, then the corrective maintenance cost is incurred in this case. When k is equal to 7, the ratio between C_c and C_p is enough large. Consequently, the DPM will automatically modify the decisions to avoid the corrective actions. Therefore, the *RD* goes down and becomes less than 7% for all three cases of C_{os}/C_i. After that, the DPM policy can be considered as stable. Next, when C_c continues to increase, a preventive maintenance will be accelerated to avoid a corrective maintenance. Consequently, it can lead to the out-of-stock situation. Therefore, the *RD* can slightly increase when the ratio between C_{os} and C_i is greater than 50.

5. Conclusion

In this paper, a new dynamic predictive maintenance (DPM) framework was presented. It is a complete process from performing the prognostics based on heterogeneous sensor data to making maintenance decisions. Its advantages are recognized on both of two aspects: prognostics estimation and post-prognostics decision.

Considering the prognostics aspect, the proposed methodology does not rely on any specific degradation model or a particular target RUL function with benefits of LSTM classifier, and therefore shows promising abilities for industry applications. Instead of predicting a RUL value, it allows providing the probabilities that the system will fall into different time intervals. As these time intervals are defined according to the requirements of the operation planner, the proposed methodology is expected to better response to practical demands. Moreover, its outputs do not depend on the period starting from the instant to make the prediction to the real system failure time, which allows limiting the wrong decisions at the first-lifetime stage.

Regarding the post-prognostic aspect, the proposed methodology also includes a decision model that allows rapidly evaluating the costs of maintenance and inventory options in order to choose the optimal activities at the beginning of the decision period. The efficiency and the performance of the proposed model were proved and highlighted when compared with the classical periodic policy and the ideal predicted prognostics results obtained from turbofan engines data provided by the

NASA Ames Prognostics Center of Excellence. For different cost combinations, the mean cost rate of the proposed DPM is significantly lower than the one of the PeM policy and close to the ideal case (IPM) with perfect prognostics information.

One of the limitation of the proposed DPM methodology is that the model considers only the perfect maintenance. Further work will investigate different levels of imperfect maintenances. It also can be extended by integrating the production planning and the complete inventory management. Finally, the impact of different quality levels of prognostic information on the performance of the predictive maintenance should be carefully investigated.

References

[1] Snchez-Silva Mauricio, Frangopol Dan M., Padgett Jamie, Soliman Mohamed. Maintenance and operation of infrastructure systems: review. J Struct Eng 2016;142(9).
[2] Wang K, Wang Y. How AI affects the future predictive maintenance: a primer of deep learning. 2018978-981-10-5767-0. p. 1–9.
[3] Gouriveau R, Medjaher K, Zerhouni N. From prognostics and health systems management to predictive maintenance 1: monitoring and prognostics. 2016. https://doi.org/10.1002/9781119371052.
[4] Papakonstantinou K, Shinozuka M. Planning structural inspection and maintenance policies via dynamic programming and markov processes. part ii: pomdp implementation. Reliab Eng Syst Saf 2014;130:214–24.
[5] Papakonstantinou K, Shinozuka M. Planning structural inspection and maintenance policies via dynamic programming and markov processes. part i: theory. Reliab Eng Syst Saf 2014;130:202–13.
[6] Nguyen TPK, Castanier B, Yeung TG. Maintaining a system subject to uncertain technological evolution. Reliab Eng Syst Saf 2014;128:56–65.
[7] Nguyen KTP, Yeung T, Castanier B. Acquisition of new technology information for maintenance and replacement policies. Int J Prod Res 2017;55(8):2212–31.
[8] You M, Liu F, Wang W, Meng G. Statistically planned and individually improved predictive maintenance management for continuously monitored degrading systems. IEEE Trans Reliab 2010;59(4):744–53.
[9] Fan H, Hu C, Chen M, Zhou D. Cooperative predictive maintenance of repairable systems with dependent failure modes and resource constraint. IEEE Trans Reliab 2011;60(1):144–57.
[10] Huynh KT, Barros A, Brenguer C. Multi-level decision-making for the predictive maintenance of k-out-of-n:f deteriorating systems. IEEE Trans Reliab 2015;64(1):94–117.
[11] Si X-S, Wang W, Hu C-H, Zhou D-H. Remaining useful life estimation a review on the statistical data driven approaches. Eur J Oper Res 2011;213(1):1–14.
[12] Loutas TH, Roulias D, Georgoulas G. Remaining useful life estimation in rolling bearings utilizing data-driven probabilistic e-support vectors regression. IEEE Trans Reliab 2013;62(4):821–32.
[13] Benkedjouh T, Medjaher K, Zerhouni N, Rechak S. Remaining useful life estimation based on nonlinear feature reduction and support vector regression. Eng Appl Artif Intell 2013;26(7):1751–60.
[14] Mosallam A, Medjaher K, Zerhouni N. Data-driven prognostic method based on bayesian approaches for direct remaining useful life prediction. J Intell Manuf 2016;27(5):1037–48.
[15] Tobon-Mejia DA, Medjaher K, Zerhouni N, Tripot G. A data-driven failure prognostics method based on mixture of gaussians hidden markov models. IEEE Trans Reliab 2012;61(2):491–503.
[16] Medjaher K, Tobon-Mejia DA, Zerhouni N. Remaining useful life estimation of critical components with application to bearings. IEEE Trans Reliab 2012;61(2):292–302.
[17] Mosallam A, Medjaher K, Zerhouni N. Nonparametric time series modelling for industrial prognostics and health management. Int J Adv Manuf Technol 2013;69(5–8):1685–99.
[18] Deutsch J, He D. Using deep learning-based approach to predict remaining useful life of rotating components. IEEE Trans Syst, Man, Cybern 2018;48(1):11–20.
[19] Tamilselvan P, Wang P. Failure diagnosis using deep belief learning based health state classification. Reliab Eng & Syst Saf 2013;115:124–35.
[20] Li C, Sanchez R-V, Zurita G, Cerrada M, Cabrera D, Vasquez RE. Gearbox fault diagnosis based on deep random forest fusion of acoustic and vibratory signals. Mech Syst Signal Process 2016;76–77:283–93.
[21] Lin Y, Li X, Hu Y. Deep diagnostics and prognostics: an integrated hierarchical learning framework in PHM applications. Appl Soft Comput (2018) 2018.
[22] Zhang J, Wang P, Yan R, Gao RX. Long short-term memory for machine remaining life prediction. J Manufact Syst (2018) 2018.
[23] Wu Y, Yuan M, Dong S, Lin L, Liu Y. Remaining useful life estimation of engineered systems using vanilla LSTM neural networks. Neurocomputing 2018;275:167–79.
[24] Zhao R, Yan R, Wang J, Mao K. Learning to monitor machine health with convolutional bi-directional LSTM networks. Sensors (Basel) 2017;17(2).
[25] Ellefsen AL, Bjorlykhaug E, Aesoy V, Ushakov S, Zhang H. Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture. Reliab Eng Syst Saf 2019;183:240–51.
[26] Liu J, Li Q, Chen W, Yan Y, Qiu Y, Cao T. Remaining useful life prediction of pemfc based on long short-term memory recurrent neural networks. Int J Hydrogen Energy 2018;In Press, Corected Proof.
[27] Ma R, Yang T, Breaz E, Li Z, Briois P, Gao F. Data-driven proton exchange membrane fuel cell degradation predication through deep learning method. Appl Energy 2018;231:102–15.
[28] Chrtien S, Herr N, Nicod J-M, Varnier C. Post-prognostics decision for optimizing the commitment of fuel cell systems. IFAC-PapersOnLine 2016;49(28):168–73. 3rd IFAC Workshop on Advanced Maintenance Engineering, Services and Technology AMEST 2016
[29] Skima H, Varnier C, Dedu E, Medjaher K, Bourgeois J. Post-prognostics decision making in distributed mems-based systems. J Intell Manuf 2017:1–12.
[30] Christer A, Wang W, Sharp J. A state space condition monitoring model for furnace erosion prediction and replacement. Eur J Oper Res 1997;101(1):1–14.
[31] Nguyen KT, Fouladirad M, Grall A. Model selection for degradation modeling and prognosis with health monitoring data. Reliability Engineering & System Safety 2018;169:105–16.
[32] Son KL, Fouladirad M, Barros A, Levrat E, Lung B. Remaining useful life estimation based on stochastic deterioration models: a comparative study. Reliab Eng Syst Saf 2013;112:165–75.
[33] Wang ZQ, Hu CH, Wang W, Si XS. An additive wiener process-based prognostic model for hybrid deteriorating systems. IEEE Trans Reliab 2014;63(1):208–22.
[34] Asgarpour M, Srensen JD. Bayesian based prognostic model for predictive maintenance of offshore wind farms. Int J Prognostics Health Manage 2018;10:1–9.
[35] Yuan M, Wu Y, Lin L. Fault diagnosis and remaining useful life estimation of aero engine using LSTM neural network. 2016 IEEE International Conference on Aircraft Utility Systems (AUS). 2016. p. 135–40.
[36] Chollet F. Deep Learning with Python. 1st Greenwich, CT, USA: Manning Publications Co.; 2017. ISBN 1617294438, 9781617294433
[37] Nguyen KTP, Amor K, Medjaher K, Picot A, Maussion P, Tobon D, et al. Analysis and comparison of multiple features for fault detection and prognostic in ball bearings. Proceedings of the European Conference of the PHM Society 2018;4(1).
[38] Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 2014;15:1929–58.
[39] Kingma DP, Ba J. Adam: a method for stochastic optimization. CoRR abs/14126980 2014. http://arxiv.org/abs/1412.6980
[40] Nguyen KTP, Fouladirad M, Grall A. New methodology for improving the inspection policies for degradation model selection according to prognostic measures. IEEE Trans Reliab 2018:1–12.
[41] Ramasso E, Saxena A. Review and Analysis of Algorithmic Approaches Developed for Prognostics on CMAPSS Dataset. Tech. Rep. SGT Inc Moffett Field United States, SGT Inc Moffett Field United States; 2014.
[42] Ramasso E, Rombaut M, Zerhouni N. Joint prediction of observations and states in time-series based on belief functions. IEEE Trans Syst, Man, Cybern, Part B: Cybernetics 2013;43(1):37–50.
[43] Liu K, Gebraeel NZ, Shi J. A data-level fusion model for developing composite health indices for degradation modeling and prognostic analysis. IEEE Trans Autom Sci Eng 2013;10(3):652–64.
[44] Ramasso E. Investigating computational geometry for failure prognostics in presence of imprecise health indicator: results and comparisons on C-MAPSS Datasets. 5. 2014. p. 1–13.