Contents lists available at ScienceDirect

Reliability Engineering and System Safety

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

A prognostic driven predictive maintenance framework based on Bayesian deep learning

Liangliang Zhuang [a,b], Ancha Xu [a,b,*], Xiao-Lin Wang [c,*]

[a] Department of Statistics, Zhejiang Gongshang University, Hangzhou 310018, China
[b] Collaborative Innovation Center of Statistical Data Engineering, Technology & Application, Zhejiang Gongshang University, Hangzhou 310018, China
[c] Business School, Sichuan University, Chengdu 610065, China

ARTICLE INFO

Keywords:
Predictive maintenance
Bayesian neural network
Deep learning
Remaining useful life
Spare parts

ABSTRACT

Recent years have witnessed prominent advances in predictive maintenance (PdM) for complex industrial systems. However, the existing PdM literature predominately separates two inter-related stages—prognostics and maintenance decision making—and either studies remaining useful life (RUL) prognostics without considering maintenance issues or optimizes maintenance plans based on given/assumed prognostic information. In this paper, we propose a prognostic driven dynamic PdM framework by integrating the two stages. In the prognostic stage, we characterize the latent structure between degradation features and RULs through a Bayesian deep learning model. By doing so, the framework is capable of generating a predictive RUL distribution that can well describe prognostic uncertainties. In the maintenance decision-making stage, we dynamically update maintenance and spare-part ordering decisions with the latest predictive RUL information, while satisfying operational constraints. The advantage of the proposed PdM framework is validated by comparison with several benchmark polices, based on the famous C-MAPSS turbofan engine data set.

1. Introduction

Maintenance is of vital importance to sustain continuous, cost-effective operations of complex industrial systems. In recent years, the prominent developments of low-cost sensing and monitoring technologies make more industrial systems equipped with on-board sensors to monitor their health conditions. The collected sensory measurements can be used to predict system failures and thus guide maintenance scheduling. This leads to a shift in maintenance paradigm towards predictive maintenance (PdM). In essence, PdM consists of two key stages: remaining useful life (RUL) prognostics and maintenance decision making. The existing PdM studies predominately separate the two inter-related stages, either studying RUL prognostics without considering downstream maintenance issues or optimizing maintenance schedules with given/assumed prognostic information.

On the one hand, many methods—physics-based, statistical, data-driven, and hybrid—have been proposed for RUL prognostics (see Lei et al. [1], Vrignat et al. [2], and Kordestani et al. [3] for overviews). Among them, data-driven methods become prevailing because they can automatically extract and construct useful information from sensory data without domain knowledge [1,3]. In particular, deep learning (DL) is gaining in popularity in data-driven prognostics due to its superior performance; it is capable of learning high-level representations behind data and predicting patterns through stacking multiple processing layers in hierarchical structures [4,5]. Li et al. [6] presented a novel deep convolutional neural network for RUL prognostics. Ellefsen et al. [7] developed a semi-supervised deep architecture that incorporates unsupervised pre-training into supervised learning for RUL prognostics for turbofan engines. Song et al. [8] proposed a bi-level long short-term memory (LSTM) model to capture degradation patterns (lower-level problem) and predict RULs (upper-level problem). Liu et al. [9] proposed a convolutional-vector fusion network by dynamically updating multi-feature weights. More research on DL-based prognostics can be found in Yu et al. [10,11], Xiang et al. [12], Siahpour et al. [13], and Xu et al. [14], among others.

Despite the progresses in DL-based prognostics, the mainstream methods only generate point estimates of RUL values and cannot provide information about the reliability of their prognostics. A critical drawback of these methods is their incapability of capturing uncertainties in prognostics. In general, prognostic uncertainties in deep learning can be divided into two types: (1) *epistemic uncertainty* that is associated with the lack of knowledge on the true model and can be reduced by acquiring more information (the more data we collect, the more certainty we are of the correct model), and (2) *aleatoric*

[*] Corresponding author.
E-mail addresses: xuancha@mail.zjgsu.edu.cn (A. Xu), xiaolinwang@scu.edu.cn (X.-L. Wang).

https://doi.org/10.1016/j.ress.2023.109181
Received 7 October 2022; Received in revised form 14 February 2023; Accepted 16 February 2023
Available online 23 February 2023

uncertainty that is concerned with random, uncontrollable disturbances in sensory data such as measurement errors [15]. Bayesian technique is widely used for uncertainty quantification [16]. Some recent studies have incorporated uncertainty quantification into machine health prognostics through Bayesian method. Peng et al. [17] extended DL models into Bayesian neural network to quantify epistemic uncertainty in RUL prognostics. Zhu et al. [18] combined active learning with Bayesian deep learning (BDL) for RUL prognostics with uncertainty quantification. Kim and Liu [19] proposed a novel BDL framework that captures general features of degradation processes and provides interval estimates of RULs. Aizpurua et al. [20] proposed a probabilistic RUL prediction framework under uncertainty integrating data-driven prognostics models with expert knowledge.

On the other hand, a large body of literature in the maintenance field has studied PdM planning with given/assumed RUL model. We thus confine our attention to recent publications. Huynh [21] developed an adaptive PdM model for single-unit systems with an inverse Gaussian degradation process. Nguyen et al. [22] presented an artificial-intelligence-based approach to optimizing maintenance policies for multi-state component systems. Wang et al. [23] studied data-driven reliability analysis and replacement policy optimization for a two-phase Wiener degradation process. Zhou et al. [24] developed a reinforcement learning (RL) algorithm to optimize maintenance of multi-component systems where parameters of the degradation process are given. Hu et al. [25] proposed a novel RL algorithm-driven maintenance strategy to optimize long-term maintenance decisions where prognostic information is known.

However, there are limited studies integrating RUL prognostics and maintenance decision making. To our knowledge, Nguyen and Medjaher [26] were the first to study prognostic driven PdM problems. They utilized an LSTM network to predict system failure probabilities at future time intervals, which, in turn, drive maintenance and spares ordering decisions upon periodic inspections. Chen et al. [27] developed a hybrid model that can obtain risk-averse point RUL predictions and then use this information to determine an exact PdM time. Chen et al. [28] presented an ensemble model to estimate system health states and failure probabilities; on this basis, the optimal time to perform maintenance was determined under a periodic inspection policy. de Pater et al. [29] proposed a PdM framework for a fleet of aircrafts with periodically updated RUL predictions. Lee and Mitici [30] utilized the RL technique to optimize maintenance actions based on estimated RUL distribution. Despite the valuable contributions, there are several shortcomings in the aforementioned studies.

- First, their prognostic methods only produce point estimates of RULs or failure probabilities in future time intervals [26,28,29], which does not reflect the uncertainties in RUL predictions. Without uncertainty quantification, these deep learning-based methods may find it difficult to provide information on the reliability and creditability of their prognostics [15,16]. Furthermore, maintenance decisions based on single-point RUL estimates may lead to suboptimal or even dangerous outcomes [17], especially for safety-critical applications, e.g., aircraft and ships.
- Second, they focus primarily on short-term maintenance scheduling and cannot recommend long-term maintenance plans with evolving prognostics [26,27,30]. For example, Nguyen and Medjaher [26] predicted the failure probability of the system in fixed time intervals, which is based on periodic inspection. Based on this information, only the maintenance strategy for the current moment can be given, without accurate prediction of future system failures. For industrial applications, a long-term maintenance schedule is also necessary for decision makers to help plan other operations & maintenance activities accordingly.
- Third, an underlying assumption in all the PdM studies mentioned above is that maintenance decisions can be freely executed. However, in practice, maintenance execution is largely constrained by operating schedules. For example, most aircraft maintenance activities cannot be executed when an aircraft is in the air; ships need to be in port for repair or replacement of damaged and worn parts. A more realistic inspection scenario is thus needed.

To overcome the shortcomings discussed previously, this paper takes a further step towards prognostic driven PdM decision making, by considering both types of prognostic uncertainties and constraints on maintenance execution. In the prognostic stage, we adopt the BDL-based framework in Kim and Liu [19] to characterize the prognostic uncertainties and produce a predictive RUL distribution. Specifically, a bidirectional LSTM (BiLSTM) network and a feedforward neural network (FNN) are trained in a unified manner, under Bayes' theorem, to quantify epistemic and aleatoric uncertainties, respectively. In the maintenance decision-making stage, we dynamically update maintenance and spares ordering decisions with the latest prognostic information, while satisfying operational constraints on maintenance execution. The performance of the proposed PdM policy is validated by comparison with several benchmark polices, based on the C-MAPSS turbofan engine data set. The rest of the paper is structured as follows. Section 2 presents the prognostic driven dynamic PdM framework that consists of the RUL prognostic stage and the maintenance decision-making stage. Section 3 conducts a case study to validate the proposed methodology. Finally, Section 4 concludes this paper.

2. Prognostic driven dynamic predictive maintenance framework

The proposed prognostic driven dynamic PdM framework consists of offline and online phases (Fig. 1). The offline phase is devoted to training a BDL-based model—using run-to-failure data—that maps historical condition monitoring (CM) data to a RUL distribution. For this purpose, after preprocessing the collected CM data, a BDL-based model is trained to address prognostic uncertainties and produce a predictive RUL distribution (see Section 2.1 for details).

At the online phase, CM data $\boldsymbol{x}_j^*$ at time t_j are gradually collected and combined into a section of trajectories $\boldsymbol{x}_{1:j}^*$. After pre-processing, the selected data are fed into the trained network to generate a predictive RUL distribution. Based on the prognostic results and operational constraints, both short-term and long-term decisions regarding maintenance and spare part ordering are recommended to system operators (see Section 2.2 for details). Such decisions are then dynamically updated and adjusted with consecutive data acquisition.

2.1. Bayesian deep learning-based prognostics

After pre-processing the collected CM data through, for example, data labeling and normalization [26], we adopt a probabilistic DL model to quantify prognostic uncertainties. For this purpose, we first discuss two types of uncertainty in prognostics.

2.1.1. Two types of uncertainty

Consider the degradation features as input $\boldsymbol{x}$ and the corresponding RUL as target y. Then, an end-to-end mapping from $\boldsymbol{x}$ to y can be expressed as

$$y = f(\boldsymbol{x}; \boldsymbol{\Theta}) + \epsilon, \tag{1}$$

where $f(\cdot; \boldsymbol{\Theta})$ represents a functional mapping with parameters $\boldsymbol{\Theta}$ and $\epsilon \sim N\left(0, \eta^2\right)$ is a Gaussian noise term with mean zero and variance η^2. According to Eq. (1), DL exhibits two types of uncertainty in the RUL prognostic problem: (i) epistemic uncertainty arising from the unknown weights $\boldsymbol{\Theta}$ due to inadequate knowledge, and (ii) aleatoric uncertainty captured by η^2, which is not the property of the model but rather an inherent property of the data distribution; hence it is irreducible [16]. Both types of uncertainty would contribute to divergence between prognosis and actual results.

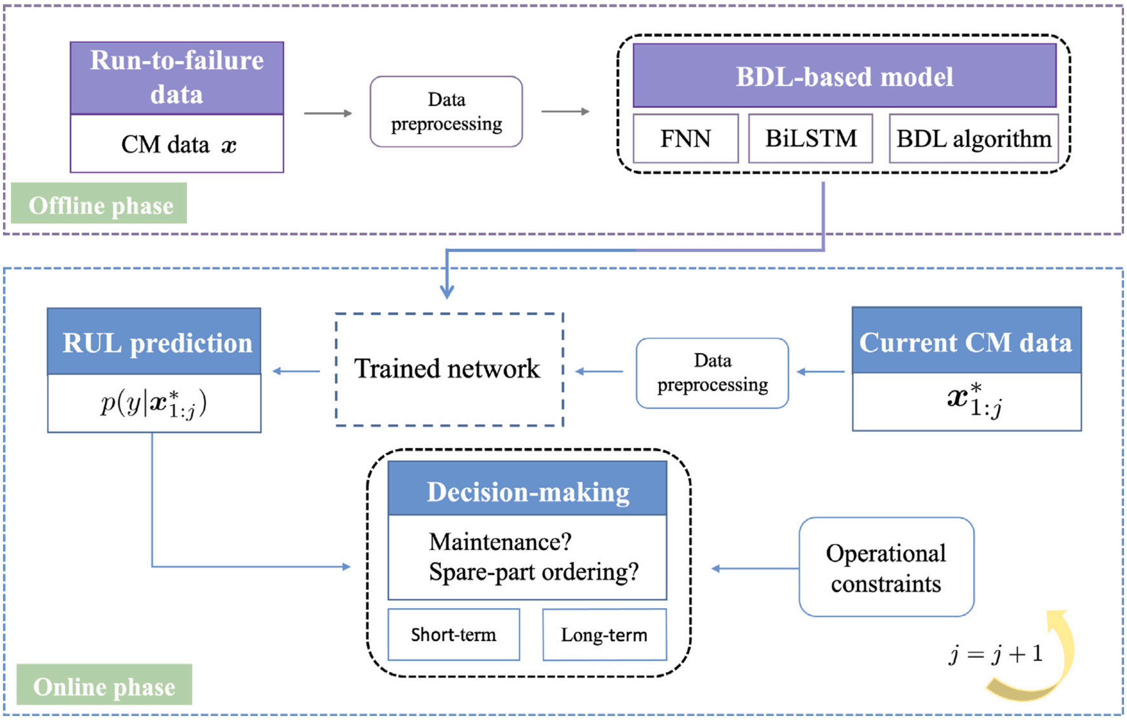


Fig. 1. Proposed prognostic driven dynamic PdM framework.

2.1.2. BDL-based network for RUL prognostics

We adopt the BDL-based framework proposed by Kim and Liu [19] for RUL prognostics (Fig. 2). Based on Eq. (1), we first use a BiLSTM network to map historical CM data to RULs, estimated by $f(\boldsymbol{x};\boldsymbol{\Theta})$. Since aleatoric uncertainty is mainly caused by measurement errors or several variabilities, it is reflected in the variance of Gaussian noise term (i.e., η^2). Thus, we input the real RUL as data into the trained network to characterize this uncertainty, namely, characterize the relationship between RUL and η^2. Since both are scalars, a simple FNN is used to describe them. Other methods (e.g., regression models, convolutional networks, among others) can be used to construct them as well. As for the epistemic uncertainty arising from the unknown weights $\boldsymbol{\Theta}$, we incorporate a Bayesian approach into the BiLSTM network by replacing deterministic weights $\boldsymbol{\Theta}$ with probabilistic ones. Then, we combine the outputs of both networks to obtain the loss; in this manner, the two networks are trained in a unified fashion to optimize prognostic performance. It should be noted that at the test time, because true RUL values are unknown, the estimated RULs from the trained BiLSTM network will serve as the FNN's inputs.

2.1.3. BiLSTM network

As stated earlier, the BiLSTM network is used to establish a mapping between CM data and RUL values, estimated by $f(\boldsymbol{x};\boldsymbol{\Theta})$. Compared to other DL models, it is able to capture long-term dependencies and learn the correlation information in both forward and backward directions. As shown in Fig. 2(a), the BiLSTM network consists of forward, backward, and regression layers. The first two layers—containing several LSTM units—are used to process input sequences bidirectionally to characterize short-term dependencies, and the regression layer captures long-term dependencies with previously extracted features. More details on the BiLSTM network can be found in Peng et al. [17], Chen et al. [28], and Huang et al. [31].

2.1.4. Quantification of aleatoric uncertainty

To quantify aleatoric uncertainty, we construct the relationship between RUL and $\log(\eta^2)$ via the FNN; see Fig. 2(b). Because the input y and output $\log(\eta^2)$ are both scalars, here we use a hidden layer with H neurons to describe their relationship:

$$\log\left(\eta^2\right) = \sum_{h=1}^{H} \gamma_h \cdot g\left(\tau_h y + \tau_{h0}\right) + \gamma_0, \tag{2}$$

where $g(\cdot)$ is a sigmoid activation function; τ_{h0} and τ_h denote the bias and weight between the input and hidden layers, respectively; γ_0 and γ_h are the bias and weight between the hidden and output layers, respectively.

2.1.5. Quantification of epistemic uncertainty

To quantify the epistemic uncertainty, we apply a Bayesian method in the BiLSTM network. Define $\boldsymbol{D} = \{\boldsymbol{X},\boldsymbol{Y}\} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$ as a training data set with inputs $\boldsymbol{x}_i$ and the corresponding RULs y_i. Using Bayes' theorem, the posterior distribution over $\boldsymbol{\Theta}$ is

$$p(\boldsymbol{\Theta}|\boldsymbol{D}) \propto \pi(\boldsymbol{\Theta}) \cdot p(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{\Theta}),$$

where $p(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{\Theta})$ is the model's likelihood, $\pi(\boldsymbol{\Theta})$ is the prior distribution of $\boldsymbol{\Theta}$. Suppose $\boldsymbol{x}^*$ is a new input value and y^* is the corresponding RUL. Then, the predictive distribution of y^* is given by

$$p\left(y^*|x^*,\boldsymbol{D}\right) = \int p(\boldsymbol{\Theta}|\boldsymbol{D}) \cdot p\left(y^*|x^*,\boldsymbol{\Theta}\right) \mathrm{d}\boldsymbol{\Theta}. \tag{3}$$

A challenge here is that the posterior distribution $p(\boldsymbol{\Theta}|\boldsymbol{D})$ is generally intractable. In this work, we adopt variational inference [32] to overcome this challenge. The idea is to approximate $p(\boldsymbol{\Theta}|\boldsymbol{D})$ by a computationally tractable distribution $\phi(\boldsymbol{\Theta})$ such that the distance between the two distributions—measured by the Kullback–Leibler (KL) divergence defined below—is minimized.

$$KL(\phi(\boldsymbol{\Theta}) \parallel p(\boldsymbol{\Theta}|\boldsymbol{D})) = -\int \phi(\boldsymbol{\Theta}) \cdot \ln \frac{p(\boldsymbol{\Theta}|\boldsymbol{D})}{\phi(\boldsymbol{\Theta})} \mathrm{d}\boldsymbol{\Theta}. \tag{4}$$

The Monte Carlo dropout method, which is easy to implement, highly scalable, and computationally efficient [33], can be adopted to minimize the KL divergence. By performing multiple stochastic forward passes with dropouts during the test time, this method is able to reflect uncertainty estimations (see Gal and Ghahramani [33] for details). Thus, if we can find a suitable $\phi(\boldsymbol{\Theta})$ such that the KL divergence in

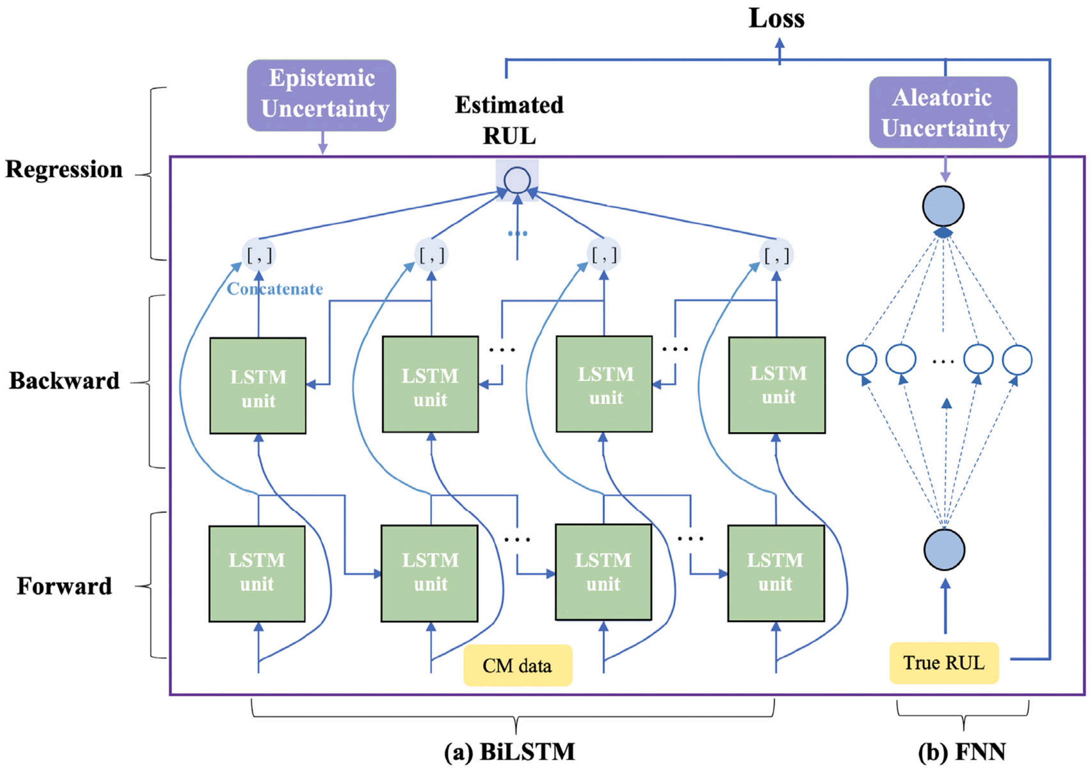


Fig. 2. BDL-based RUL prognostics at the training time.

Eq. (4) is minimized, then the predictive distribution in Eq. (3) can be approximated by

$$p\left(y^*|x^*,\boldsymbol{D}\right) \approx \int \phi(\boldsymbol{\Theta}) \cdot p\left(y^*|x^*,\boldsymbol{\Theta}\right) \mathrm{d}\boldsymbol{\Theta}. \tag{5}$$

2.1.6. *Loss function*

Unlike most existing DL literature in which the objective is to minimize the sum of squared errors, $\{f(\boldsymbol{x};\boldsymbol{\Theta})-y\}^2$, with variance σ^2 being fixed or ignored, our work aims at minimizing the entire negative log-likelihood while capturing aleatoric uncertainty through the FNN:

$$-\log p\left(y|f\left(\boldsymbol{x};\boldsymbol{\Theta}\right)\right) \propto \frac{1}{2\eta^2}\left\{f\left(\boldsymbol{x};\boldsymbol{\Theta}\right)-y\right\}^2+\frac{1}{2}\ln\eta^2.$$

In addition, we add an l_2-regularization term of η^2 to the loss function in order to avoid exploration of unconstrained η^2:

$$\frac{1-\omega}{2N}\sum_{n=1}^{N}\left\{\frac{\left\{f\left(\boldsymbol{x}_n;\boldsymbol{\Theta}\right)-y_n\right\}^2}{\eta_n^2}+\ln\eta_n^2\right\}+\omega\sum_{n=1}^{N}\left|\eta_n^2\right|^2,$$

where η_n^2 captures the aleatoric uncertainty of sample n, and $\omega \in (0,1)$ is a tuning parameter that can be determined by cross validation.

2.2. *Prognostic driven maintenance decision making*

Based on the predictive RUL information, we propose a prognostic driven dynamic PdM framework in which maintenance and spares ordering decisions can be dynamically updated, while satisfying operational constraints on maintenance execution. The framework imposes the following assumptions:

(i) Maintenance is perfect through replacing systems with new identical spares.
(ii) Spare parts are ordered only when needed so as to minimize inventory holding costs, and the lead time is a constant, denoted by $\mathcal{L}$.
(iii) Maintenance can be executed when spare parts are unavailable, but it incurs an extra out-of-stock cost.
(iv) Maintenance activities can only be executed in a series of time windows $\mathcal{S} = \{[t_{d_1},t_{e_1}],\ldots,[t_{d_s},t_{e_s}]\}$ and can be completed within a single period, whereas spare parts can be ordered at any time.

2.2.1. *Tentative PdM scheduling with operational constraints*

Assume that CM data are collected at equidistant epochs $t_1,t_2,\ldots$, with $t_1-t_0=t_2-t_1=\cdots=\Delta t$ and $t_0=0$ corresponding to the time when a new (or replaced) system starts operating. Given the current time t_j, the probability density function of RUL, $p(y|\boldsymbol{x}^*_{1:j})$, for any y can be obtained via the prognostic framework in Section 2.1. Based on this information, the operator can decide on whether to replace the degraded system (R) or do nothing (DN) at the current and any future moments, so as to minimize the average cost rate. To facilitate our presentation, let c_p be the preventive maintenance cost, c_c the corrective maintenance cost, c_{os} the out-of-stock cost when spare parts are not available, c_f the cost of wasting a unit of system RUL, and c_q the spare-part holding cost per unit time.

Suppose that at time t_j, the operator decides to replace the system at a future moment t_{j+k}, $k \in \{0,1,\ldots\}$. At this stage, we assume that the spare part is pre-ordered at $t_{j+k}-\mathcal{L}$ and can arrive exactly at t_{j+k}. Then, two scenarios should be considered. If the system fails before t_{j+k}, then a corrective replacement will be performed upon failure; in this case, there will be no available spare part and an out-of-stock cost will be incurred. Otherwise, a preventive replacement will be executed at t_{j+k}; in this case, spare part is available but the system's RUL will be wasted. Therefore, at time t_j, choosing to replace the system at t_{j+k} yields the following cost rate:

$$\begin{aligned} C^{\mathrm{R}}_{j,j+k} = & \int_{0}^{t_{j+k}-t_j} p(y|\boldsymbol{x}^*_{1:j})\frac{c_c+c_{os}}{t_j+y}\mathrm{d}y \\ & + \int_{t_{j+k}-t_j}^{+\infty} p(y|\boldsymbol{x}^*_{1:j})\frac{c_p+c_f(y-t_{j+k}+t_j)}{t_{j+k}}\mathrm{d}y. \end{aligned} \tag{6}$$

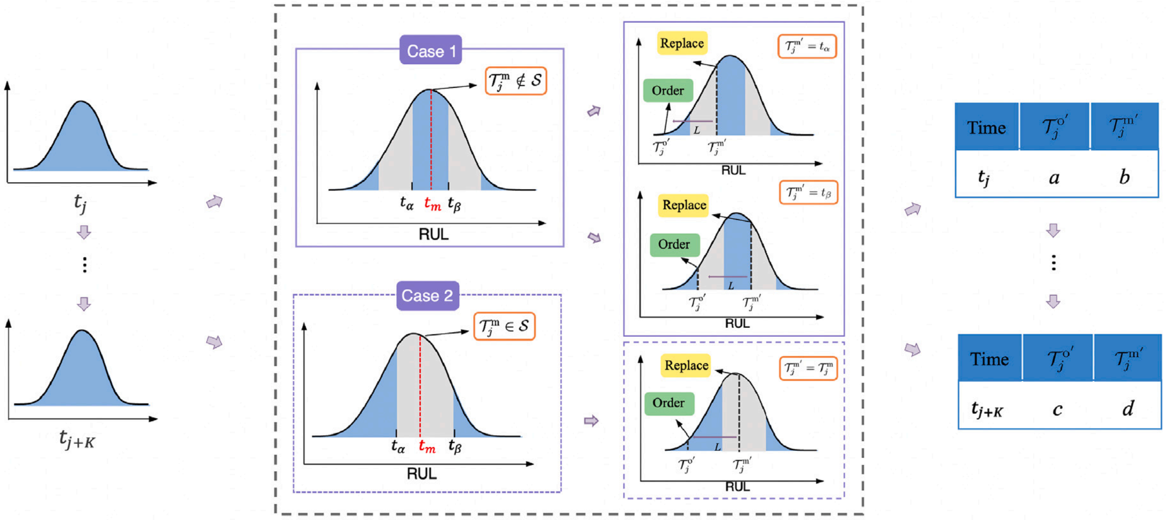

t_j
t_{j+K}
Case 1
$\mathcal{T}_j^{\rm m} \notin \mathcal{S}$
t_α t_m t_β
RUL
Case 2
$\mathcal{T}_j^{\rm m} \in \mathcal{S}$
t_α t_m t_β
RUL
Replace
Order
$\mathcal{T}_j^{\rm m'} = t_\alpha$
$\mathcal{T}_j^{\rm o'}$ $\mathcal{T}_j^{\rm m'}$
L
RUL
Replace
Order
$\mathcal{T}_j^{\rm m'} = t_\beta$
RUL
Replace
Order
$\mathcal{T}_j^{\rm m'} = \mathcal{T}_j^{\rm m}$
RUL
Time $\mathcal{T}_j^{\rm o'}$ $\mathcal{T}_j^{\rm m'}$
t_j a b
Time $\mathcal{T}_j^{\rm o'}$ $\mathcal{T}_j^{\rm m'}$
t_{j+K} c d

Fig. 3. Tentative PdM scheduling with operational constraints (the gray areas indicate available maintenance windows).

Because the predictive RUL information is obtained at discrete epochs, the cost rate can be approximated by

$$C^{\rm R}_{j,j+k} \approx \sum_{h=0}^{k-1} p_{h|j}\frac{c_c+c_{os}}{t_{j+h}} + \sum_{h=k}^{+\infty} p_{h|j}\frac{c_p+c_f(h-k)\Delta t}{t_{j+k}}, \tag{7}$$

where $p_{h|j}$ is the probability mass function that a system will fail h periods ahead given that the current period is j.

On the contrary, if the operator decides to do nothing at t_{j+k}, then the system may fail before the $(j+k)$th period, or in the $(j+k+1)$th period, resulting in an unexpected failure in either case. The cost rate of the DN-option is thus

$$C^{\rm DN}_{j,j+k} = \int_0^{t_{j+k+1}-t_j} p(y|\boldsymbol{x}^*_{1:j})\frac{c_c+c_{os}}{t_j+y}\,{\rm d}y \approx \sum_{h=0}^{k+1} p_{h|j}\frac{c_c+c_{os}}{t_{j+h}}. \tag{8}$$

The optimal action at time t_{j+k} is thus the one with a lower cost rate:

$$\mathcal{A}_{j,j+k} = \begin{cases} R, & \text{if } C^{\rm R}_{j,j+k} \le C^{\rm DN}_{j,j+k}, \\ DN, & \text{if } C^{\rm R}_{j,j+k} > C^{\rm DN}_{j,j+k}. \end{cases} \tag{9}$$

As can be seen, the framework can not only make an instantaneous maintenance decision at the current time t_j, but also recommend a long-term maintenance plan for any future moment t_{j+k}. In particular, the first time when the cost of R-option becomes lower than that of DN-option, while satisfying operational constraints, can be regarded as the tentative maintenance time:

$$\mathcal{T}_j^{\rm m'} = \mathcal{T}_j^{\rm m}\cdot \mathbb{1}_{\{\mathcal{T}_j^{\rm m}\in\mathcal{S}\}} + \tilde{\mathcal{T}}_j^{\rm m}\cdot\left(1-\mathbb{1}_{\{\mathcal{T}_j^{\rm m}\in\mathcal{S}\}}\right), \tag{10}$$

where $\mathbb{1}_{\{\cdot\}}$ is the indicator function that equals to 1 if the argument is true, and 0 otherwise. In this expression, $\mathcal{T}_j^{\rm m} = \inf_{k\in\{0,1,\dots\}}\{t_{j+k} : C^{\rm R}_{j,j+k} \le C^{\rm DN}_{j,j+k}\}$ is the tentative maintenance time when it is in the window $\mathcal{S}$, while $\tilde{\mathcal{T}}_j^{\rm m} = \inf_{\zeta\in\{\alpha,\beta\}}\{t_\zeta : C^{\rm R}_{j,\zeta} \le C^{\rm DN}_{j,\zeta}\}$ is the tentative maintenance time when $\mathcal{T}_j^{\rm m} \notin \mathcal{S}$. Here, α is the time slot at the end of the last window before $\mathcal{T}_j^{\rm m}$, and β is the one at the beginning of the first window after $\mathcal{T}_j^{\rm m}$ (see Fig. 3).

Accordingly, the tentative ordering time $\mathcal{T}_j^{\rm o'}$ can be calculated by subtracting the constant lead time $\mathcal{L}$ from $\mathcal{T}_j^{\rm m'}$. To reduce the fluctuation due to dynamic updating, the predictions $\mathcal{T}_j^{\rm m'}$'s for the recent Q cycles are averaged to determine $\mathcal{T}_j^{\rm o'}$:

$$\mathcal{T}_j^{\rm o'} = \left\lfloor \frac{\sum_{q=0}^{\min\{j-1,Q-1\}}\mathcal{T}_{j-q}^{\rm m'}}{\min\{j,Q\}} \right\rfloor - \mathcal{L}. \tag{11}$$

where $\lfloor x \rfloor = \max\{n\in\mathbb{Z} \mid n \le x\}$.

2.2.2. Dynamic PdM updating and adjusting

The aforementioned PdM scheduling is tentative because RUL prognostics for a long period of time would have a low accuracy. Fortunately, as more CM data are progressively collected, the proposed BDL-based framework is expected to generate a more accurate prognostic result and thus enables us to update maintenance and spares ordering plans.

With successively updated $(\mathcal{T}_j^{\rm m'}, \mathcal{T}_j^{\rm o'})$, the spare part should be ordered when $\mathcal{T}_j^{\rm o'}$ is no later than the current time t_j for the first time. That is, the predicted optimal ordering time is

$$\mathcal{T}^{\rm o^*} = \inf_{j\in\mathbb{Z}^+}\{t_j : \mathcal{T}_j^{\rm o'} \le t_j\}. \tag{12}$$

Similarly, the predicted maintenance time is given by

$$\mathcal{T}^{\rm m'} = \inf_{t_j\in\mathcal{S}}\{t_j : \mathcal{T}_j^{\rm m'} \le t_j\}. \tag{13}$$

Note that $\mathcal{T}^{\rm m'}$ is not necessarily the final maintenance moment. This is because we still need to update the RUL prognosis after the spare part is ordered, especially when the lead time $\mathcal{L}$ is long. As a result, there might be a mismatch between spare arrival and maintenance execution (i.e., $\mathcal{T}^{\rm o^*} + \mathcal{L} \ne \mathcal{T}^{\rm m'}$), resulting in spare-part holding or shortage costs. Let $t_a = \mathcal{T}^{\rm o^*} + \mathcal{L}$, and $t_b = \mathcal{T}^{\rm m'}$. When reaching the time point $\min\{t_a, t_b\}$, we need to assess a series of average cost rates over the interval $[\min\{t_a,t_b\}, \max\{t_a,t_b\}]$, so as to adjust the final maintenance decision. For this purpose, three scenarios should be considered:

(i) $t_a < t_b$. In this scenario, the spare part has arrived at t_a, so there is no shortage cost. We need to evaluate the average cost rates for discrete moments $\{t_a,\dots,t_b\}$ at the current time t_a, denoted as $\{C_{a,a},\dots,C_{a,b}\}$, respectively. The optimal maintenance time can thus be determined by

$$\mathcal{T}^{\rm m^*} = \mathop{\arg\min}_{t_{a+j}\in\mathcal{S}}\left\{C_{a,a},\dots,C_{a,a+j},\dots,C_{a,b}\right\}, \tag{14}$$

where

$$C_{a,a+j} = \sum_{h=0}^{j-1} p_{h|a}\frac{c_c+c_q h}{t_{a+h}} + \sum_{h=j}^{+\infty} p_{h|a}\frac{c_p+c_q j+c_f(h-j)}{t_{a+j}}$$

for $j = 0,1,\dots,b-a$. Here, we use $t_b = \mathcal{T}_a^{\rm m'}$, because $\mathcal{T}_a^{\rm m'}$ is the latest tentative maintenance time updated at t_a.

(ii) $t_a > t_b$. In this scenario, the spare part will arrive later than the predicted maintenance execution time, the shortage cost should be considered if the maintenance is performed before t_a. The optimal maintenance time can thus be determined by

$$\mathcal{T}^{\rm m^*} = \mathop{\arg\min}_{t_{b+j}\in\mathcal{S}}\left\{C_{b,b},\dots,C_{b,b+j},\dots,C_{b,a}\right\}, \tag{15}$$

where

$$C_{b,b+j} = \sum_{h=0}^{j-1} p_{h|b}\frac{c_c + c_{os}}{t_{b+h}} + \sum_{h=j}^{+\infty} p_{h|b}\frac{c_p + c_{os}\cdot \mathbb{1}_{\{j<a-b\}} + c_f(h-j)}{t_{b+j}}$$

for $j = 0, 1, \ldots, a-b$.

(iii) $t_a = t_b$. The optimal maintenance time is simply

$$\mathcal{T}^{\mathrm{m}^*} = \mathcal{T}^{\mathrm{m}'}. \tag{16}$$

2.2.3. Performance evaluation of the dynamic PdM policy

We now focus on performance evaluation of the dynamic PdM policy. Suppose the system runs R life cycles, and denote $\mathcal{X}_r = \min\{\mathcal{T}_r^{\mathrm{m}^*}, \mathcal{T}_r^{\mathrm{f}}\}$, where $\mathcal{T}_r^{\mathrm{m}^*}$ and $\mathcal{T}_r^{\mathrm{f}}$ are predicted optimal maintenance time and actual failure time of the rth life cycle, respectively. The actual cost rate of the rth life cycle can be calculated by

$$CR_r = \begin{cases} \dfrac{c_p + c_{os}\delta_r + c_q\kappa_r(1-\delta_r) + c_f\sum_{h=1}^{+\infty} hp_{h|\psi_r}}{\mathcal{T}_r^{\mathrm{m}^*}}, \mathcal{X}_r = \mathcal{T}_r^{\mathrm{m}^*}, \\ \dfrac{c_c + c_{os}\delta_r + c_q\kappa_r(1-\delta_r)}{\mathcal{T}_r^{\mathrm{f}}}, \mathcal{X}_r = \mathcal{T}_r^{\mathrm{f}}, \end{cases} \tag{17}$$

where κ_r is the number of periods to store the spare part and δ_r is a direct function (i.e., $\delta_r = 1$ when a spare part is unavailable), and ψ_r is the discrete period that $\mathcal{T}_r^{\mathrm{m}^*}$ locates in. The average cost rate for all life cycles is thus given by

$$\overline{CR} = \frac{\sum_{r=1}^{R}\mathcal{X}_r \cdot CR_r}{\sum_{r=1}^{R}\mathcal{X}_r}. \tag{18}$$

The implementation procedures of the prognostic driven dynamic PdM framework are summarized in Algorithm 1.

Algorithm 1: Prognostic driven dynamic PdM framework

```
Input: $x^*, c_c, c_p, c_{os}, c_q, c_f, \mathcal{S}, \mathcal{L}, Q$.
Output: $\overline{CR}$, $\mathcal{T}_r^{\mathrm{o}^*}$, $\mathcal{T}_r^{\mathrm{m}^*}$, and $CR_r, r = 1, \ldots, R$.
obtain the well-trained DBL model in Section 2.1.
for $r = 1$ to $R$ do
    while no maintenance performed do
        if the system is working then
            collect new data $x_j^*$;
            compute $\{p_{h|j}\}_{h=0}^{+\infty}$ ;
            for $k = 0$ to $+\infty$ do
                compute $C_{j,j+k}^{\mathrm{R}}$ and $C_{j,j+k}^{\mathrm{DN}}$ by (7) and (8),
                  respectively;
                determine $\mathcal{A}_{j,j+k}$ by (9);
            end
            determine $\mathcal{T}_r^{\mathrm{o}^*}$ by (12);
            determine $\mathcal{T}_r^{\mathrm{m}^*}$ by (14), (15) or (16) ;
            if $t_j = \mathcal{T}_r^{\mathrm{m}^*}$ then
                preventive maintenance.
            end
        end
        else
            corrective maintenance;
            set $\mathcal{T}_r^{\mathrm{f}} = t_j$.
        end
        $j = j + 1$
    end
    compute $CR_r$ by (17).
end
compute $\overline{CR}$ by (18).
```

3. Numerical experiment

The proposed method is applied to the turbofan aircraft engine data set. In Section 3.1, an overview of the data set and pre-processing process is provided. Section 3.2 demonstrates the prognostic accuracy in comparison with other prognostic methods. Section 3.3 introduces our proposed dynamic PdM framework, which consists of two parts: (i) tentative PdM scheduling with operational constraints; (ii) dynamic PdM updating and adjusting. In Section 3.4, the performance of the proposed policy is compared with five benchmark policies to assess its effectiveness.

3.1. Case description

We adopt the famous C-MAPSS dataset [34] to validate the effectiveness of the prognostic driven dynamic PdM framework. This dataset is made available by the NASA Ames Prognostics Center of Excellence. In this study, CM data for 200 engines in the “FD001” sub-dataset are used, which are obtained under the same operational condition. The CM data for 100 engines with run-to-failure data are treated as training data, and those for 100 working engines are test data.

In order to provide an end-to-end solution, we do not take any feature extraction step. All sensor signals, operational variables, and cycle times are used as inputs. To increase the amount of training data, a sliding time window approach is used [19]; in particular, 25 consecutive points are used as one input sample. Each sample is pre-processed by min–max normalization. For label rectification, a linear RUL function with a maximum value of 125 is utilized for each training sample [6,19].

Using the Keras and TensorFlow libraries in Python, we create a DBL-based network (Fig. 2), which is implemented in the Google Collaboratory environment using GPU accelerators. The parameters of this constructed network are as follows. FNN has 1 hidden layer and 5 neurons. BiLSTM has 2 hidden layers and each layer has 20 neurons. RMS-Prop is used to train model with a learning rate of 0.01 and a dropout probability of 0.2.

3.2. Discussion of prognostic accuracy

3.2.1. Evaluation criteria

Suppose that there are M test systems. Let d_m be the difference between estimated and actual RUL values for the mth system, $m = 1, 2, \ldots, M$. To assess the prognostic performance, the following three criteria are considered:

Score (SC) allows to penalize a late prediction more severely than an early prediction.

$$\mathrm{SC} = \sum_{m=1}^{M} s_m, \text{ where } s_m = \begin{cases} e^{-\frac{d_m}{13}} - 1, & \text{if } d_m < 0, \\ e^{\frac{d_m}{10}} - 1, & \text{if } d_m \geq 0. \end{cases}$$

Root mean square (RMSE) penalizes both late and early predictions equally.

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{m=1}^{M}(d_m)^2}{M}}.$$

Accuracy (AC) measures the percentage of RUL differences d_m’s that fall within the tolerance interval $[-13, 10]$.

$$\mathrm{AC} = \frac{100}{M}\sum_{m=1}^{M} a_m, \text{ where } a_m = \begin{cases} 1, & \text{if } d_m \in [-13, 10], \\ 0, & \text{if } d_m \notin [-13, 10]. \end{cases}$$

Note that a smaller SC, a smaller RMSE, or a larger AC indicates a better RUL prediction performance.

Table 1
Comparison of point prediction with other methods.

	SC	RMSE	AC
MODBNE [35]	334.2	15.0	–
MCLSTM [12]	315.0	13.7	–
TSCG [36]	468.5	17.4	–
DCNN [6]	273.7	12.6	–
BDL-LSTM [19]	267.2	**12.2**	–
GA-RBM-LSTM [7]	231.0	12.6	–
DBNBP-IPF [37]	543.0	–	51%
DBN-IPF [37]	314.0	–	63%
BiLSTM-ED [10]	273.0	14.7	57%
SBI-EN [11]	**228.0**	13.6	67%
Proposed method	234.9	12.7	**70**%

Table 2
RUL prediction with uncertainty quantification for test engine unit #24.

Cycle	Real value	Point estimation	95% Confidence interval
164	42	47.536	$[36.153, 56.624]$
167	39	41.929	$[34.598, 51.443]$
170	36	41.586	$[32.424, 50.940]$
173	33	35.206	$[28.262, 42.342]$
176	30	26.876	$[21.728, 33.511]$
179	27	27.309	$[21.617, 33.009]$
182	24	24.886	$[20.036, 31.302]$
185	21	22.655	$[17.637, 27.351]$

3.2.2. Comparison with other prognostic methods

To illustrate the performance of the BDL-based prognostic method, a comparison of point estimates of RULs for different methods is shown in Table 1. For the proposed method, we implement 1000 stochastic forward passes during the test process and utilize the mean values of these predicted RULs as the point estimation; the simulation is repeated for 20 times. Table 1 shows that the proposed method is among the best ones and can achieve a comparative performance. Specifically, despite that it is not the best model in terms of RMSE and SC, it is indeed the best one according to the AC.

Notably, most methods considered above only output a single predicted RUL value. However, our method can obtain interval predictions of RULs that can quantify the uncertainty of each individual engine. For example, following Li et al. [6], Fig. 4 shows the RUL interval estimates for 4 test engine units under different cycles. We can see that our method works well on these engines with high precision, especially when there are many samples (units #24, #34 and #81); with an increase in sample size, the prediction performance improves gradually. Fig. 5 and Table 2 further show the RUL predictions with uncertainty quantification for engine unit #24. Decision makers can use kernel distributions to obtain the system's failure probability in a future moment, which facilitates maintenance decision making.

3.3. Dynamic predictive maintenance framework

To compare the performance of our PdM framework with that of others, we have to rely on complete information on the engine states during the entire life cycle, that is, from the beginning of operations to failure, so that we can evaluate the maintenance costs under different frameworks. For this purpose, the FD001 training set is divided into two parts. The first part consists of 80 engines for network training, and the second contains the remaining 20 units for performance validation. The same setting has been widely adopted in various PdM studies [26,28,30].

Based on the predictive RUL distribution, we first show how the process of tentative PdM scheduling works, under operational constraints. Then, we illustrate how to dynamically update maintenance and spares ordering decisions. For this purpose, we set $c_p = 100$, $c_c = 500$, $c_{os} = 10$, $c_f = 1$, $c_q = 0.1$, $\mathcal{L} = 20$ and $Q = 6$ [26]. Three maintenance-window cases are considered: (i) $\mathcal{S} = \{1, 2, \dots\}$. (ii) $\mathcal{S} = \{[10, 20], [30, 40], \dots\}$. (iii) $\mathcal{S} = \{10, 20, \dots\}$, which is a periodic inspection. A visualization of the three cases is shown in Fig. 6.

Table 3
Cost rates for engine unit #81.

Cycle $(200 + k)$	CMF	DN-cost	R-cost	$\mathcal{A}_{200,200+k}$
200	0	0	0.736	DN
⋮	⋮	⋮	⋮	⋮
234	0.301	1.095	1.031	DN
235	0.442	1.316	1.328	DN
236	**0.529**	**1.609**	**1.490**	***R***
237	0.653	1.803	1.740	R
238	0.730	2.166	1.862	R
⋮	⋮	⋮	⋮	⋮
247	1.000	2.452	2.452	R

3.3.1. Tentative PdM scheduling with operational constraints

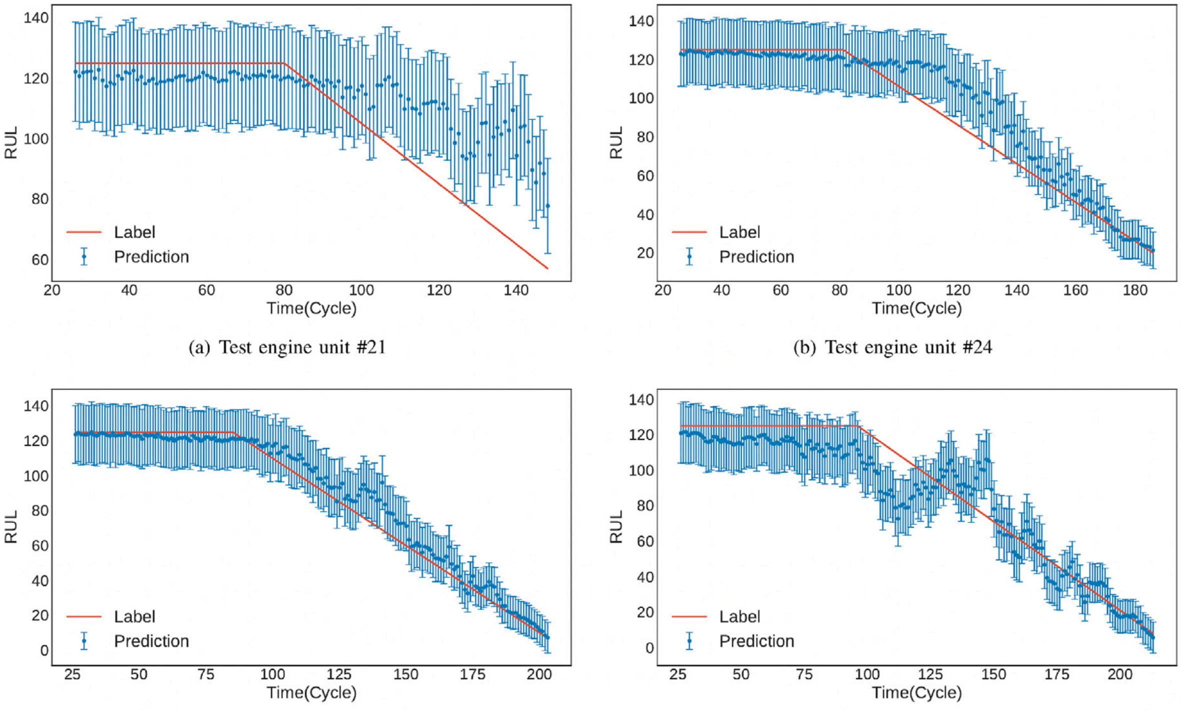
Based on the RUL prognostics, we compute the cost rates of R- and DN-option at the current and any future time points. By doing so, we know what should be done at the current moment and when to perform maintenance in the future.

For illustration, the result for engine unit #81 at time t_{200} is shown in Table 3. We can see that the cost rate of the DN-option is much smaller than that of the R-option at the beginning, so maintenance operation is not needed. As the process evolves, the value of cumulative mass function (CMF) $\sum_{h=0}^{k} p_{h|200}$ becomes larger. When the cost rate of the R-option becomes smaller than that of the DN-option for the first time, the tentative maintenance time $\mathcal{T}_{200}^{\mathrm{m}} = 236$ can be obtained. Then, we use Eqs. (10) and (11) to obtain tentative maintenance and spares ordering time, respectively. The results for the three maintenance-window cases are $(\mathcal{T}_{200}^{\mathrm{m}'}, \mathcal{T}_{200}^{\mathrm{o}'}) = (236, 216), (236, 215)$, and $(230, 212)$, respectively.

3.3.2. Dynamic PdM updating and adjusting

On this basis, the system operator can update and adjust maintenance and spares ordering plans when more CM data are obtained. Table 4 shows the dynamic PdM policy for engine unit #81 in the three cases. With successively updated $(\mathcal{T}_j^{\mathrm{m}'}, \mathcal{T}_j^{\mathrm{o}'})$, the optimal spare-part ordering time $\mathcal{T}^{\mathrm{o}^*}$ can be obtained; the results for the three cases are 219, 218, and 216, respectively. After that, the tentative maintenance time is continuously updated until t_a or t_b is reached. In all the three cases, t_a is reached before t_b; specifically, the spare part arrives at 239, 238, and 236 respectively. We can then seek the optimal maintenance time over the interval $[t_a, t_b]$, which are $[239, 242]$, $[238, 240]$, and $[236, 240]$, respectively. By (14), the optimal maintenance times $\mathcal{T}^{\mathrm{m}^*}$ of these cases are 239, 238, and 240, respectively. The predicted maintenance times using (13) are 242, 240, 240, respectively. We note that the actual useful lifetime of this engine unit is 240. As can be seen, if we do not dynamically update and adjust the optimal maintenance time based on the latest prognostic information, there will be a corrective maintenance cost in case (i). While the optimal maintenance time determined by our dynamic updating method is quite close, yet prior, to the engine's actual lifetime. This implies that we can make full use of the engine's lifetime, while avoiding high corrective maintenance cost. This demonstrates the practical value of our proposed PdM methodology in real-world industrial applications, where cost-effectiveness and efficiency are of utmost importance.

Another advantage of the proposed PdM framework is that decisions can be made quickly based on the predictive RUL distribution. It allows evaluation of cost parameters at decision time, which brings greater flexibility to decision makers to better adapt to cost changes. For an illustration, Table 5 shows the optimal actions under different c_c/c_p for engine unit #81, where LR means lifetime usage rate and $\mathcal{T}^{\mathrm{m}'}$ is the predicted maintenance time. From Table 5, it can be found that the optimal maintenance time exhibits a non-increasing trend with the increase of c_c/c_p. This result is consistent with our intuition that a higher corrective maintenance cost would induce an earlier system replacement. The optimal maintenance actions in case (iii) are relatively stable and not greatly impacted by the cost parameters. This is due



(a) Test engine unit #21

(b) Test engine unit #24

(c) Test engine unit #34

(d) Test engine unit #81

Fig. 4. RUL interval estimates for four test engine units.

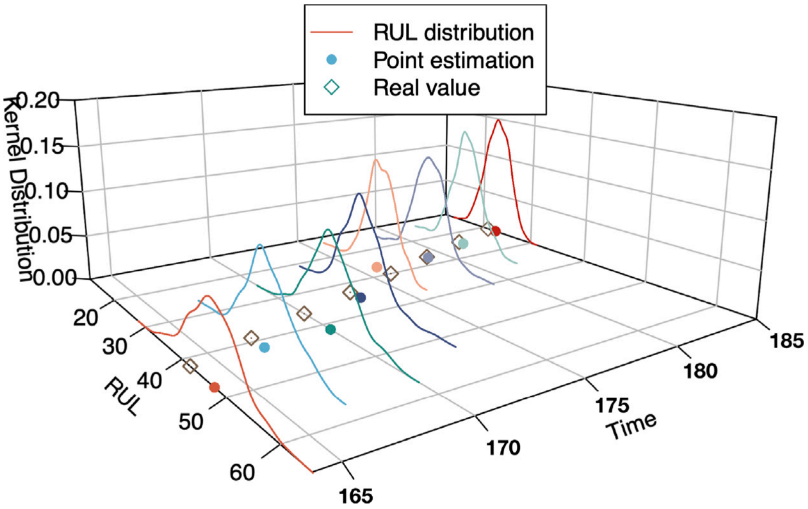


Fig. 5. Uncertainty quantification for test engine unit #24.

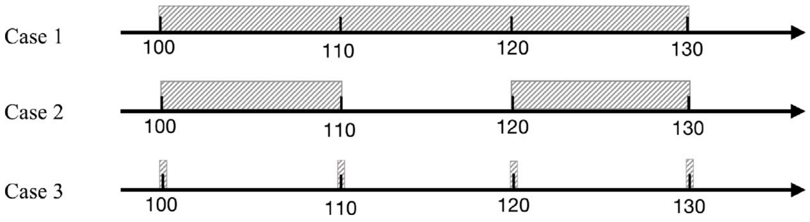


Fig. 6. Three maintenance-window cases (the gray box indicates available maintenance windows).

to the operational schedule constraints which dictate the selection of optimal maintenance time within the designated maintenance window.

In most cases, the optimal maintenance time is slightly less than the unit's lifetime, allowing for avoidance of both unit failure and the waste

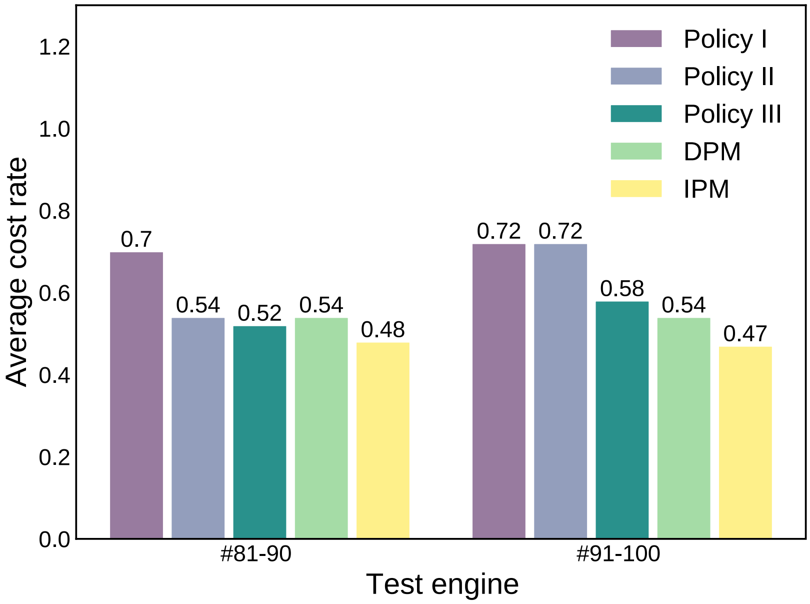


Fig. 7. Average cost rates for test engines under periodic inspection.

Table 4
Dynamic PdM policy of unit #81 in the three cases.

Cycle	Case (i)		Case (ii)		Case (iii)	
	$\mathcal{T}_j^{m'}$	$\mathcal{T}_j^{o'}$	$\mathcal{T}_j^{m'}$	$\mathcal{T}_j^{o'}$	$\mathcal{T}_j^{m'}$	$\mathcal{T}_j^{o'}$
216	241	219	240	218	240	**216**
217	238	219	238	218	230	216
218	237	219	237	**218**	230	215
219	236	**219**	236	218	230	215
220	238	219	238	218	230	215
⋮	⋮	⋮	⋮	⋮	⋮	⋮
236	240	221	240	219	**240**	217
237	240	221	240	219	–	–
238	243	221	**240**	219	–	–
239	**242**	221	–	–	–	–

Table 5
The optimal actions under different c_c/c_p for engine unit #81.

c_c/c_p	Case (i)				Case (ii)				Case (iii)			
	$\mathcal{T}^{o^*}$	$\mathcal{T}^{m'}$	$\mathcal{T}^{m^*}$	LR	$\mathcal{T}^{o^*}$	$\mathcal{T}^{m'}$	$\mathcal{T}^{m^*}$	LR	$\mathcal{T}^{o^*}$	$\mathcal{T}^{m'}$	$\mathcal{T}^{m^*}$	LR
2	220	246	242	100.83%	219	240	239	99.58%	217	240	240	100%
5	219	242	239	99.58%	218	240	238	99.17%	216	240	240	100%
8	217	241	238	99.17%	217	239	237	98.75%	215	230	230	95.83%
10	216	240	237	98.75%	216	238	237	98.75%	215	230	230	95.83%
15	216	238	236	98.33%	216	237	236	98.33%	215	230	230	95.83%
20	215	238	236	98.33%	215	236	236	98.33%	214	230	230	95.83%

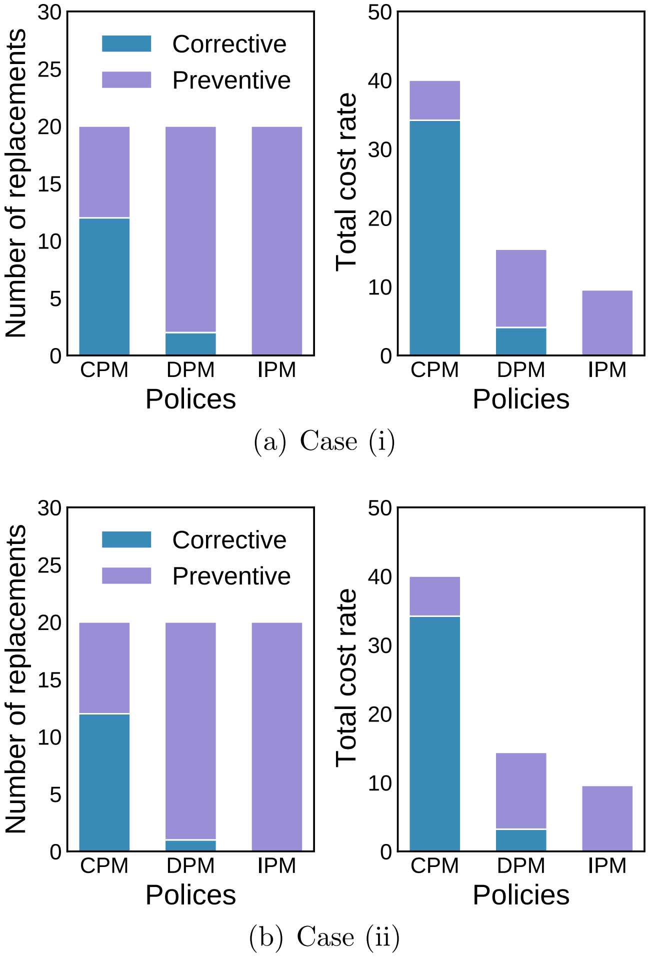
of its lifetime. This highlights the importance of considering both cost and operational factors in industrial maintenance planning to ensure efficient and effective operations.

3.4. Comparison with different maintenance policies

In this subsection, we compare the proposed methodology with several benchmark policies to assess its performance.

3.4.1. Benchmark maintenance policies

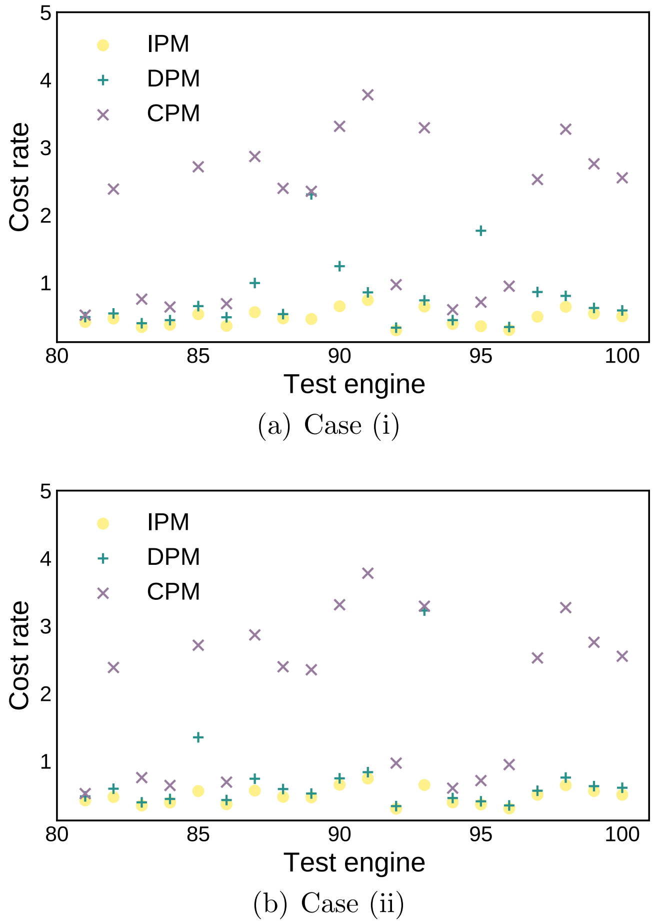
We first consider the following two simplified policies.



(a) Case (i)

(b) Case (ii)

Fig. 8. Performance of the three policies under cases (i) and (ii).



(a) Case (i)

(b) Case (ii)

Fig. 9. Cost rate for each test engine under cases (i) and (ii).

- **Classical PdM policy** (CPM) that is based on historical reliability data. The preventive maintenance time is determined by

$$\mathcal{T}_r^{\dagger} = \bar{\mathcal{T}}^F \cdot \mathbb{1}_{\{\bar{\mathcal{T}}^F \in S\}} + t_{\alpha^{\dagger}} \cdot \left(1 - \mathbb{1}_{\{\bar{\mathcal{T}}^F \in S\}}\right),$$

where $\bar{\mathcal{T}}^F$ is the system's mean time to failure and $\alpha^{\dagger}$ is the time slot at the end of the last window before $\bar{\mathcal{T}}^F$. Let μ denote the discrete period that the preventive maintenance time $\mathcal{T}_r^{\dagger}$ locates in, and $\mathcal{X}_r^{\dagger} = \min\{\mathcal{T}_r^{\dagger}, \mathcal{T}_r^{\mathrm{f}}\}$. The cost rate of the rth life cycle is

$$CR_r^{\dagger} = \begin{cases} \dfrac{c_p}{\mathcal{T}_r^{\dagger}}, & \mathcal{X}_r^{\dagger} = \mathcal{T}_r^{\dagger}, \\ \dfrac{c_c + c_{os}}{\mathcal{T}_r^{\mathrm{f}}}, & \mathcal{X}_r^{\dagger} = \mathcal{T}_r^{\mathrm{f}}. \end{cases}$$

- **Ideal PdM policy** (IPM) that is based on the assumption of perfect predicted failure time $\mathcal{T}_r^P$ with an available spare part. The preventive maintenance time is determined by

$$\mathcal{T}_r^{\ddagger} = \mathcal{T}_r^P \cdot \mathbb{1}_{\{\mathcal{T}_r^P \in S\}} + t_{\alpha^{\ddagger}} \cdot \left(1 - \mathbb{1}_{\{\mathcal{T}_r^P \in S\}}\right),$$

where $\alpha^{\ddagger}$ is the time slot at the end of the last window before $\mathcal{T}_r^P$. The cost rate of the rth life cycle is

$$CR_r^{\ddagger} = \frac{c_p}{\mathcal{T}_r^{\ddagger}}.$$

Then, the average cost rate over all life cycles can be computed by Eq. (18). In addition, three state-of-the-art PdM policies (see Nguyen and Medjaher [26], Zhang and Zhang [38], and Chen et al. [28]) under the periodic inspection policy are also considered, denoted by Policies I, II, and III, respectively. Our proposed dynamic PdM policy is denoted by the DPM policy.

3.4.2. Performance of the proposed framework

The DPM policy is compared with other benchmark polices, under the aforementioned maintenance-window cases. We first compare the average cost rates of the 20 test engines under the periodic inspection policy (i.e., case (iii)); the result is shown in Fig. 7. Though the average cost rate of our policy for engine units #81-90 is higher than that of Policy III, it indeed has the lowest overall cost rate (i.e., 0.54), except the IPM policy. Note that the cost of wasting RUL, c_f, is included in the average cost rate for the DPM policy, but not taken into account for the other policies. After removing the cost c_f, the average cost rate of DPM for the 20 test engines is reduced to 0.52, which is now much closer to that of IPM. In addition, the three policies are only applicable to the periodic inspection scenarios. However, in practice, maintenance execution may be subject to various operational constraints as stated in Introduction, making it important to consider both periodic and aperiodic inspection scenarios. To address this issue, we provide a more comprehensive and effective solution for ensuring the reliability of a system. We take into account operational constraints on maintenance execution and present a practical policy in general inspection scenarios. Performance comparison under the other two maintenance-window cases are shown in Figs. 8 and 9, respectively. We can observe that the cost rate of each test engine obtained by DPM is quite close to the IPM policy. The number of test units undergoing corrective maintenance is close to zero, and the average cost rate is much lower than that of the CPM policy. These findings highlight the superior performance of the DPM strategy in real-world industrial applications.

4. Conclusion

In this work, a new prognostic driven PdM framework is proposed, which provides a comprehensive solution that integrates RUL prognostics and maintenance decision making. In the prognostic stage, we adopt a BDL-based framework to qualify aleatoric and epistemic uncertainties, and output a predictive distribution of RULs. In the maintenance decision-making stage, a practical policy in general inspection scenarios is presented. This model enables prompt evaluation of the cost rates of R- and DN-option at any moment, and produces tentative PdM schedule that satisfies operational constraints. As more CM data are progressively collected, our framework dynamically updates and adjusts maintenance and spare-part ordering decisions to generate a more reliable PdM schedule.

By comparison with several benchmark polices, based on the turbofan aircraft engine data set provided by the NASA Ames Prognostics Center of Excellence, we find that the proposed policy driven by the BDL method can enhance the prognostics result with uncertainty quantification, thereby improving the performance of dynamic PdM decision making. Under both periodic and aperiodic inspection scenarios, the proposed policy results in an average cost rate that is much close to an ideal policy. This study has practical implications for industries, demonstrating the benefits of incorporating uncertainty quantification and operational constraints in PdM policy. The enhanced policy performance leads to better maintenance planning, reducing costs and increasing profitability, while also improving customer satisfaction.

We believe that the proposed prognostics driven dynamic PdM planning framework can be applied to condition-monitored complex systems in other industries as well. Of course, slight adaptions of this framework might be needed; for example, (i) additional industry-specific constraints may be taken into account for maintenance of different types of complex systems; (ii) more realistic data with advanced techniques (e.g., deep RL, active learning) can be considered.

CRediT authorship contribution statement

Liangliang Zhuang: Writing – original draft, Software, Methodology, Investigation, Formal analysis. **Ancha Xu:** Writing – review & editing, Supervision, Conceptualization. **Xiao-Lin Wang:** Writing – review & editing, Validation, Supervision, Methodology, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data is public available.

Acknowledgments

The research is supported in part by the Humanities and Social Sciences Youth Foundation of the Ministry of Education of China (grant number 22YJC630142) and in part by the National Natural Science Foundation of China (grant numbers 72201180, 12171432, and 11671303).

References

[1] 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.

[2] Vrignat P, Kratz F, Avila M. Sustainable manufacturing, maintenance policies, prognostics and health management: A literature review. Reliab Eng Syst Saf 2022;218:108140.

[3] Kordestani M, Saif M, Orchard ME, Razavi-Far R, Khorasani K. Failure prognosis and applications—A survey of recent literature. IEEE Trans Reliab 2019;70(2):728–48.

[4] Zhao R, Yan R, Chen Z, Mao K, Wang P, Gao RX. Deep learning and its applications to machine health monitoring. Mech Syst Signal Process 2019;115:213–37.

[5] Khan S, Yairi T. A review on the application of deep learning in system health management. Mech Syst Signal Process 2018;107:241–65.

[6] Li X, Ding Q, Sun J-Q. Remaining useful life estimation in prognostics using deep convolution neural networks. Reliab Eng Syst Saf 2018;172:1–11.

[7] Ellefsen AL, Bjørlykhaug E, Æsøy 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.

[8] Song T, Liu C, Wu R, Jin Y, Jiang D. A hierarchical scheme for remaining useful life prediction with long short-term memory networks. Neurocomputing 2022;487:22–33.

[9] Liu X, Lei Y, Li N, Si X, Li X. RUL prediction of machinery using convolutional-vector fusion network through multi-feature dynamic weighting. Mech Syst Signal Process 2023;185:109788.

[10] Yu W, Kim IY, Mechefske C. Remaining useful life estimation using a bidirectional recurrent neural network based autoencoder scheme. Mech Syst Signal Process 2019;129:764–80.

[11] Yu W, Kim IY, Mechefske C. An improved similarity-based prognostic algorithm for RUL estimation using an RNN autoencoder scheme. Reliab Eng Syst Saf 2020;199:106926.

[12] Xiang S, Qin Y, Luo J, Pu H, Tang B. Multicellular LSTM-based deep learning model for aero-engine remaining useful life prediction. Reliab Eng Syst Saf 2021;216:107927.

[13] Siahpour S, Li X, Lee J. A novel transfer learning approach in remaining useful life prediction for incomplete dataset. IEEE Trans Instrum Meas 2022;71:1–11.

[14] Xu DY, Qiu HB, Gao L, Yang Z, Wang DP. A novel dual-stream self-attention neural network for remaining useful life estimation of mechanical systems. Reliab Eng Syst Saf 2022;222:108444.

[15] Hüllermeier E, Waegeman W. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. Mach Learn 2021;110(3):457–506.

[16] Abdar M, Pourpanah F, Hussain S, Rezazadegan D, Liu L, Ghavamzadeh M, et al. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. Inf Fusion 2021;76:243–97.

[17] Peng W, Ye Z-S, Chen N. Bayesian deep-learning-based health prognostics toward prognostics uncertainty. IEEE Trans Ind Electron 2019;67(3):2283–93.

[18] Zhu R, Chen Y, Peng W, Ye Z-S. Bayesian deep-learning for RUL prediction: An active learning perspective. Reliab Eng Syst Saf 2022;228:108758.

[19] Kim M, Liu K. A Bayesian deep learning framework for interval estimation of remaining useful life in complex systems by incorporating general degradation characteristics. IISE Trans 2020;53(3):326–40.

[20] Aizpurua J, Stewart B, McArthur S, Penalba M, Barrenetxea M, Muxika E, et al. Probabilistic forecasting informed failure prognostics framework for improved RUL prediction under uncertainty: A transformer case study. Reliab Eng Syst Saf 2022;226:108676.

[21] Huynh KT. An adaptive predictive maintenance model for repairable deteriorating systems using inverse Gaussian degradation process. Reliab Eng Syst Saf 2021;213:107695.

[22] Nguyen V-T, Do P, Vosin A, Iung B. Artificial-intelligence-based maintenance decision-making and optimization for multi-state component systems. Reliab Eng Syst Saf 2022;228:108757.

[23] Wang J, Tan L, Ma X, Gao K, Jia H, Yang L. Prognosis-driven reliability analysis and replacement policy optimization for two-phase continuous degradation. Reliab Eng Syst Saf 2023;230:108909.

[24] Zhou Y, Li B, Lin TR. Maintenance optimisation of multicomponent systems using hierarchical coordinated reinforcement learning. Reliab Eng Syst Saf 2022;217:108078.

[25] Hu Y, Miao XW, Zhang J, Liu J, Pan ES. Reinforcement learning-driven maintenance strategy: A novel solution for long-term aircraft maintenance decision optimization. Comput Ind Eng 2021;153:107056.

[26] Nguyen KT, Medjaher K. A new dynamic predictive maintenance framework using deep learning for failure prognostics. Reliab Eng Syst Saf 2019;188:251–62.

[27] Chen C, Lu N, Jiang B, Wang C. A risk-averse remaining useful life estimation for predictive maintenance. IEEE/CAA J Autom Sin 2021;8(2):412–22.

[28] Chen C, Zhu ZH, Shi J, Lu N, Jiang B. Dynamic predictive maintenance scheduling using deep learning ensemble for system health prognostics. IEEE Sens J 2021;21(23):26878–91.

[29] de Pater I, Reijns A, Mitici M. Alarm-based predictive maintenance scheduling for aircraft engines with imperfect remaining useful life prognostics. Reliab Eng Syst Saf 2022;221:108341.

[30] Lee J, Mitici M. Deep reinforcement learning for predictive aircraft maintenance using probabilistic remaining-useful-life prognostics. Reliab Eng Syst Saf 2023;230:108908.

[31] Huang C-G, Huang H-Z, Li Y-F. A bidirectional LSTM prognostics method under multiple operational conditions. IEEE Trans Ind Electron 2019;66(11):8792–802.

[32] Blei DM, Kucukelbir A, McAuliffe JD. Variational inference: A review for statisticians. J Amer Statist Assoc 2017;112(518):859–77.

[33] Gal Y, Ghahramani Z. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In: International conference on machine learning. 2016, p. 1050–9.

[34] Saxena A, Goebel K, Simon D, Eklund N. Damage propagation modeling for aircraft engine run-to-failure simulation. In: 2008 International Conference on Prognostics and Health Management. 2008, p. 1–9.

[35] Zhang C, Lim P, Qin AK, Tan KC. Multiobjective deep belief networks ensemble for remaining useful life estimation in prognostics. IEEE Trans Neural Netw Learn Syst 2016;28(10):2306–18.

[36] Xu H, Fard N, Fang Y. Time series chain graph for modeling reliability covariates in degradation process. Reliab Eng Syst Saf 2020;204:107207.

[37] Peng K, Jiao R, Dong J, Pi Y. A deep belief network based health indicator construction and remaining useful life prediction using improved particle filter. Neurocomputing 2019;361:19–28.

[38] Zhang L, Zhang J. A data-driven maintenance framework under imperfect inspections for deteriorating systems using multitask learning-based status prognostics. IEEE Access 2020;9:3616–29.