

Article citation info:

He X, Su C, Zhang Y, Tang J, Optimal Imperfect Predictive Maintenance Based on Interval Remaining Useful Life Prediction, *Eksploracja i Niezawodność – Maintenance and Reliability* 2025; 27(4) <http://doi.org/10.17531/ein/203458>

## Optimal Imperfect Predictive Maintenance Based on Interval Remaining Useful Life Prediction

Indexed by:



Xiaoliang He<sup>a,b</sup>, Chun Su<sup>a,\*</sup>, Yuru Zhang<sup>a</sup>, Jiuqiang Tang<sup>a</sup>

<sup>a</sup> School of Mechanical Engineering, Southeast University, China

<sup>b</sup> College of Design and Engineering, National University of Singapore, Singapore

### Highlights

- A PdM framework combines interval RUL prediction with maintenance policy optimization
- BiTCN with multi-head attention significantly improves RUL prediction accuracy
- ABKDE is used to estimate the prediction uncertainty
- DCS optimizes hyperparameters of RUL prediction model and decision variables of PdM
- Case studies demonstrate outstanding performance of the proposed approach

### Abstract

Predictive maintenance (PdM) is crucial in prognostics and health management (PHM), with remaining useful life (RUL) prediction as the core. To address challenges in deep learning-based RUL prediction models, including hyperparameter tuning, uncertainty, and the application of prediction results, this paper proposes a novel PdM framework that combines interval RUL prediction with maintenance policy optimization. The framework utilizes a bidirectional temporal convolutional network (BiTCN) with multi-head attention (MHA) and an improved kernel density estimation method for RUL prediction, with a physical model supporting maintenance decision-making. Meanwhile, the differential creative search (DCS) algorithm is introduced to optimize model hyperparameters and PdM decision variables. Experimental results show that the proposed model reduces RMSE by 3.20% and 3.68% on the FD002 and FD004 datasets, respectively. Moreover, the DCS-optimized model reduces RMSE and MAPE by 5.94% and 11.11%, respectively, compared to the BiTCN-MHA model. Sensitivity analysis also demonstrate its strong robustness despite the variations in maintenance costs or times.

### Keywords

differential creative search (DCS), bidirectional temporal convolutional network (BiTCN), adaptive bandwidth kernel density estimation, interval RUL prediction, predictive maintenance (PdM)

This is an open access article under the CC BY license (<https://creativecommons.org/licenses/by/4.0/>)

### 1. Introduction

Prognostics and health management (PHM) is a comprehensive solution that integrates advanced information and artificial intelligence technologies to monitor equipment health status and predict potential faults, thereby significantly improving operational and maintenance efficiency [1, 2]. Among them, remaining useful life (RUL) prediction and maintenance optimization are two crucial parts of PHM.

Maintenance operations can preserve the intended

functionality of a product and restore its working condition. Moreover, its value can be measured with the economic loss caused by insufficient or delayed maintenance operations [3, 4]. With the increase in the complexity and value of the equipment, the risks and loss due to failure-induced downtime will also increase. Therefore, the importance of maintenance is growing. Commonly-used maintenance strategies include corrective maintenance (CM) [5], time-based maintenance (TBM) [6] and

(\*) Corresponding author.

E-mail addresses:

X. He (ORCID: 0009-0002-4983-3137) [230219115@seu.edu.cn](mailto:230219115@seu.edu.cn), C. Su (ORCID: 0000 0002 5523 1469) [suchun@seu.edu.cn](mailto:suchun@seu.edu.cn), Y. Zhang (ORCID: 0000-0002-5768-1455) [ru1094405891@163.com](mailto:ru1094405891@163.com), J. Tang (ORCID: 0000-0002-9155-3721) [jiuqiangtang1998@163.com](mailto:jiuqiangtang1998@163.com)

condition-based maintenance (CBM) [7]. CM refers to the maintenance activities carried out after a failure occurs. As for important equipment or system, CM may lead to significant economic losses. The determination of maintenance intervals in TBM often lacks theoretical support, thus it often leads to over-maintenance or under-maintenance. Based on the results of condition monitoring and fault prediction, CBM can enhance maintenance precision and effectiveness. However, the above maintenance strategies cannot well meet the diversified maintenance demands.

In recent years, new maintenance strategies, such as predictive maintenance (PdM), opportunistic maintenance (OM), and performance-based maintenance (PBM), have gained significant attention [8, 9]. PdM can generally be categorized into model-based approaches (MBA) and data-driven approaches (DBA) [10]. MBA involves constructing performance degradation models in either discrete or continuous time. Discrete-time models are typically built using historical reliability data or known system state transitions, with the Markov process being frequently applied. MBA constructs performance degradation models in discrete or continuous time. Discrete-time models are established on historical reliability data or known system state transitions, and Markov process is frequently used. Meanwhile, stochastic processes are adopted in continuous-time models to represent the degrading process. However, the above models still face challenges including system dynamics and environmental randomness. Unlike MBA, DBA do not depend on the degradation mechanisms, where the performance and condition data are adopted for decision-making. With the increase of installed sensors, DBA have received much attention.

Health state assessment and RUL prediction provide the basis for developing maintenance plans. With advancements in sensor technology and the Internet of Things (IoT), recent data-driven deep learning approaches have gained considerable attention for RUL prediction. Methods such as long short-term memory (LSTM) networks, random forests (RF), graph neural network (GNN) and Transformers have shown particular promise in these fields [11-16]. Liu et al. [17] proposed an LSTM-based RUL framework, while Cho et al. [18] introduced the gated recurrent unit (GRU), an efficient variant of LSTM. Wang et al. [19] proposed a gated graph convolutional network

with multi-sensor signals for remaining useful life prediction. Li et al. [20] developed the simple convolutional neural network (CNN) architecture to predict the aero engine RUL. Yang et al. [21] proposed a dual CNN for bearing RUL prediction. Moreover, by combining frequency-domain analysis, CNN variants like ResNet [22] and temporal convolutional network (TCN) [23] are also commonly used. Meanwhile, in order to enhance temporal feature selection, attention mechanisms are also integrated into TCN. For example, Cao et al. [24] developed a TCN with residual self-attention mechanisms for predicting bearing RUL. Deep learning models possess the powerful capabilities of feature extraction and representation learning, while they also face inevitable challenges. CNN may not perform as well as RNN-based methods in handling sequential features; LSTM excels in modeling long-term dependencies but may suffer from vanishing or exploding gradients; GRU may underperform in some complex sequential tasks; TCN has advantages in processing sequential data and can parallelize training for faster execution, but it may overfit when predicting long sequences.

Although many studies have been done on deep learning models for RUL prediction, the uncertainty has not been adequately addressed. In practical applications, deep learning predictions are often influenced by factors such as random noise and modeling parameters, which significantly reduce the usability of RUL predictions and may lead to equipment failure due to delayed maintenance. Therefore, accurate interval prediction of RUL is crucial for understanding the degradation process and supports effective risk analysis and maintenance decision-making.

To address the need for uncertainty quantification in deep learning-based RUL prediction models, a variety of advanced interval prediction models have been proposed, including bootstrap, local uncertainty, stochastic process, Bayesian, and quantile regression [25, 26]. Huang et al. [27, 28] introduced bootstrap CNN architecture to extract both the series-based and image-based features, and it was embedded in a bootstrap framework for RUL interval prediction. Guo et al. [29] used bidirectional LSTM (BiLSTM) to extract features for rolling bearings, and the uncertainty was quantified with the Bootstrap method. Chen et al. [30] transformed the RUL prediction interval into a Gaussian-based probability distribution, and

a cost-rate function was developed to optimize the aero engine's maintenance decisions. Chen et al. [31] proposed a method for lithium-ion battery capacity prediction, where the quantile regression was adopted, and a one-dimensional CNN and BiLSTM were integrated. Zhang et al. [32] developed a distributed RUL prediction method by using deep network quantile regression, and confidence intervals were identified via the cumulative distribution function to quantify RUL uncertainty.

Moreover, with the increase of the complexity in deep learning models, the number of hyperparameters grows exponentially. It has posed a great challenge for traditional manual tuning, how to achieve optimal network parameters while improving the accuracy and efficiency in RUL prediction becomes a critical issue. To address this, intelligent optimization algorithms have been introduced into deep learning, such as genetic algorithms (GA), grey wolf optimizers (GWO), and newer methods like the whale optimization algorithm (WOA) [33-35]. Among them, the differential creative search (DCS) algorithm is a novel meta-heuristic and it has shown excellent properties, including strong evolutionary capability, fast search speed, and superior optimization performance [36]. Therefore, incorporating DCS into RUL prediction is beneficial to improve the accuracy of deep learning models.

As a whole, there are still shortcomings in existing studies, some of them are as follows: (1) Most studies focus on fault diagnosis and RUL prediction, with limited research on the integration of predictive maintenance and maintenance decision optimization. (2) Deep learning methods (such as LSTM, GRU, CNN, TCN, Transformer, etc.) perform well in handling sequential data, but also face some challenges. (3) Existing methods still face challenges in uncertainty analysis, data processing, and feature selection, particularly in handling confidence intervals and model uncertainty in RUL prediction. (4) The hyperparameter optimization remains a difficult issue in deep learning models.

To address the challenges in RUL prediction and PdM optimization, this study proposes a novel DCS-BiTCN-PdM method. In detail, the major contributions are as follows: (1) BiTCN is used to capture global and local sequence features, while the attention mechanism can enhance long-range

dependency modeling. (2) ABKDE performs probability density estimation for RUL predictions which address data non-uniformity. (3) The interval RUL predictions are applied to PdM to obtain optimal maintenance strategy to minimize costs. (4) The DCS algorithm is employed to optimize the hyperparameters of the BiTCN-MHA model and maintenance interval and levels of PdM.

The rest of the paper is organized as follows. Section 2 outlines the framework of the proposed DCS-BiTCN-PdM method and provides a detailed description of each component. Section 3 demonstrates the method's effectiveness and advantages using the C-MPASS aero engine dataset and PdM optimization. In Section 4, a sensitivity analysis is conducted. Finally, Section 5 offers conclusions and suggests future research directions.

## 2. Methodology

In this study, the BiTCN-MHA model is adopted for RUL prediction. Meanwhile, the optimization of PdM strategies is conducted by using interval RUL predictions and DCS algorithm to minimize the maintenance costs. The framework for the proposed interval RUL prediction and PdM optimization is illustrated in Fig. 1.

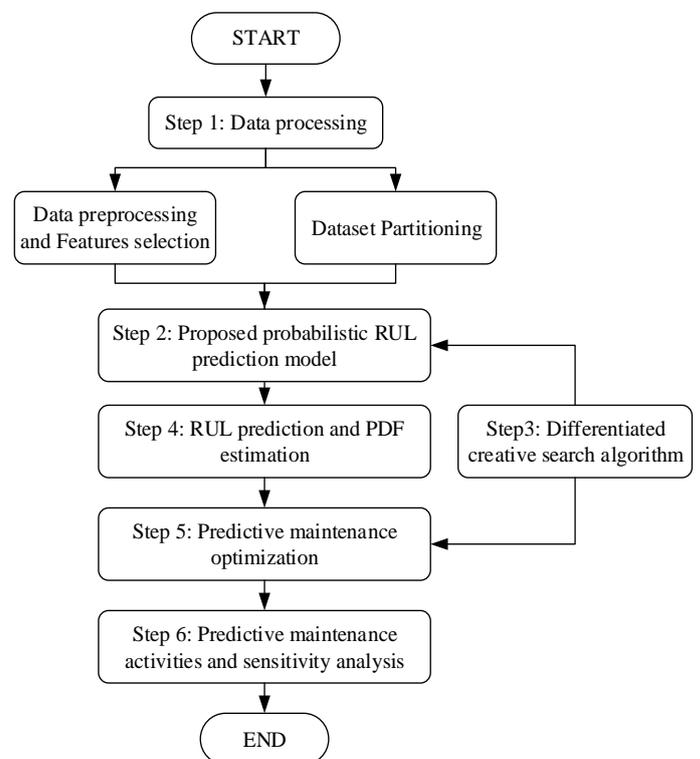


Fig. 1. The framework for the proposed DCS-BiTCN-PdM method.

In this study, the DCS-BiTCN-PdM method implementation follows these steps: (1) It starts with data processing, including preprocessing, feature selection, and dataset partitioning, as shown in Step 1. (2) A probabilistic RUL prediction model is then developed (i.e. Step 2). (3) Integration of the DCS algorithm to optimize RUL prediction in Step 3. (4) In Step 4, RUL prediction and probability density function (PDF) estimation are conducted. (5) PdM optimization is performed based on the RUL results in Step 5. (6) Finally, the optimized PdM is implemented and a sensitivity analysis is conducted (i.e. Step 6).

## 2.1. Data processing and partitioning

### 2.1.1 Normalization

To facilitate training and testing, the data is normalized. The Min-Max normalization is applied and transforms the data to the range  $[-1,1]$ . The normalization process can be expressed as [37]:

$$x_{\text{norm}}^{w,z} = \frac{x^{w,z} - x_{\text{min}}^w}{x_{\text{max}}^w} \quad (1)$$

where  $x_{\text{norm}}^{w,z}$  represents the normalized value;  $x^{w,z}$  is the  $z$ -th data of the  $w$ -th sensor;  $x_{\text{min}}^w$ ,  $x_{\text{max}}^w$  represent the minimum and maximum data of the  $w$ -th sensor, respectively.

### 2.1.2. Sliding window

Fig. 2 shows the flow of sliding time window. This approach creates sliced samples to incorporate more degradation information into each sample. It enables the model to capture temporal dependencies within equipment degradation data. Local data segments are extracted by sliding a fixed-size time window across the sequence. Two key parameters, i.e. the window size and stride length, play a crucial role in determining the accuracy and stability of the analysis [38, 39]. In addition, 80% of the data is randomly allocated for model training, while the remaining 20% is reserved for validation and hyperparameter tuning.

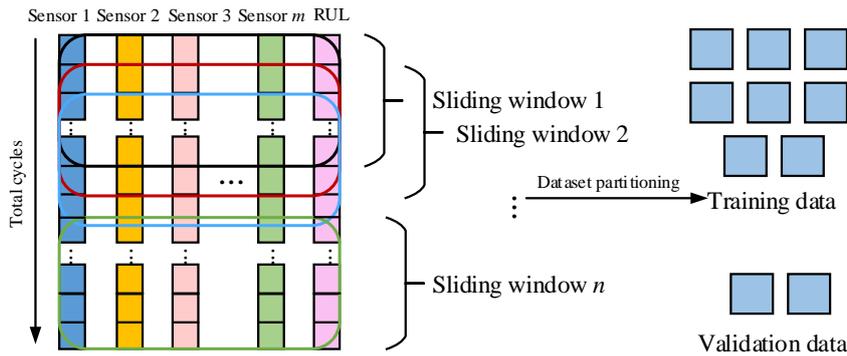


Fig. 2. The flow of sliding time window.

## 2.2. Deep learning model for interval prediction of RUL

### 2.2.1. BiTCN network and MHA mechanism

By integrating the merits of CNN and RNN, TCN uses dilated causal convolutional layers that ensure consistent input and output lengths [23]. The causal convolution mechanism is illustrated in Fig. 3. Traditional TCN can only perform convolutions in the forward direction, limiting feature extraction and ignoring information from the reverse direction. This limitation can hinder the model's ability to capture dependencies that span both past and future inputs. In contrast, BiTCN is an extension of TCN that captures hidden features in both the forward and backward directions. This bidirectional approach enhances its ability to model long-term dependencies by considering both past and future information. This ability of

BiTCN improves its extraction performance, especially in tasks where future data is crucial for accurate predictions. The network structure of the BiTCN is illustrated in Fig. 4.

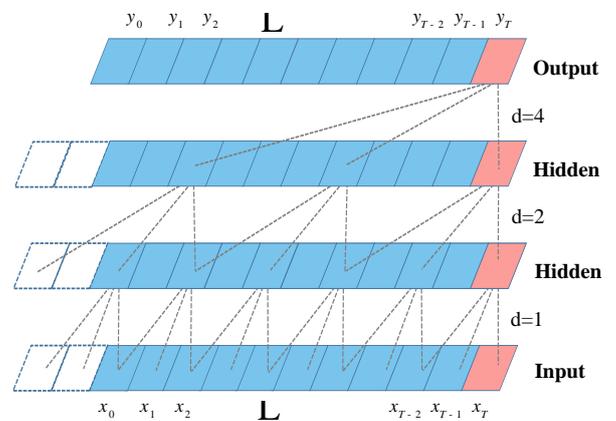


Fig. 3. The concept of causal convolution.

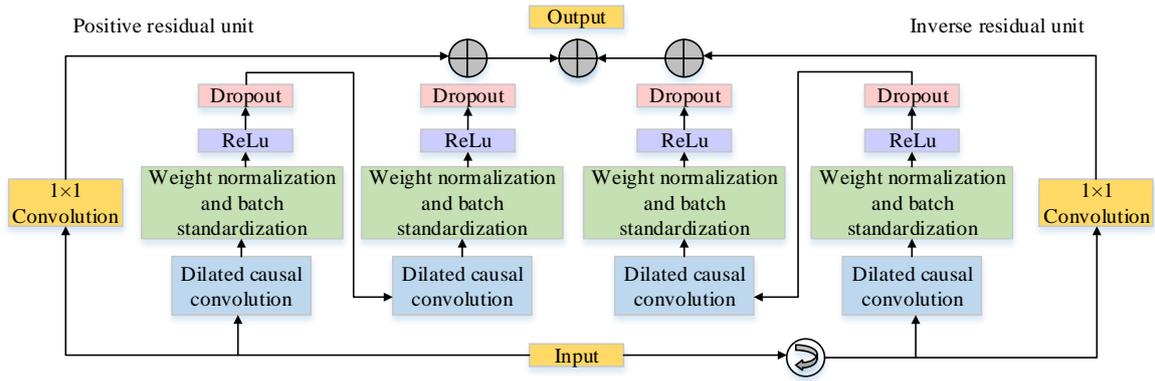


Fig. 4. The network structure of BiTCN.

To further enhance the model's capability, the MHA is integrated into BiTCN. The MHA mechanism is inspired by the Transformer architecture, which allows the model to focus on different parts of the input sequence simultaneously. Unlike traditional attention mechanisms, which focus on a single attention score, MHA splits the attention mechanism into multiple heads, each learning a different aspect of the input sequence. These multiple attention heads are then combined to form a richer representation, allowing the model to capture more complex relationships between different parts of the sequence. The MHA can be expressed as:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O \quad (2)$$

where  $\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$

where  $Q$ ,  $K$ , and  $V$  represent queries, keys, and values,  $\{QW_i^Q, KW_i^K, VW_i^V\}$  represents the trainable weight matrix for  $Q$ ,  $K$ , and  $V$  in  $i$ -th attention head, and  $W^O$  is the weight matrix.

### 2.2.2. Improved adaptive bandwidth kernel density estimation

By using varying bandwidths at different locations, adaptive bandwidth kernel density estimation (ABKDE) improves upon traditional kernel density estimation (KDE), which can better capture the characteristics of local data. To derive the formula for ABKDE, we first review the traditional KDE and then introduce the adaptive bandwidth. The form of traditional KDE is as follows [40]:

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x-x_i}{h}\right) \quad (3)$$

where  $\hat{f}(x)$  represents density estimate at position;  $K(\cdot)$  is Gaussian kernel function;  $h$  is the bandwidth; and  $x_i$  is the  $i$ -th observed data point.

During traditional KDE, the bandwidth is a global constant,

and it remains the same for all data points. It means that the degree of smoothing is consistent across all the locations. Meanwhile, the bandwidth (i.e.  $h$ ) of ABKDE is no longer a global constant, while it will dynamically adjust according to the position of the data points. For each data point  $x_i$ , the bandwidth can be modified according to the local structure of the data, referred to as the adaptive bandwidth  $h_i$ . Therefore, the ABKDE can be expressed as:

$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^n \frac{1}{h_i} K\left(\frac{x-x_i}{h_i}\right) \quad (4)$$

The adaptive bandwidth can be selected using various methods. In this paper, the adaptive bandwidth is chosen with the local weighted regression method, where the bandwidth is adjusted according to the density of local data. For denser regions, smaller bandwidths should be adopted, while for sparser regions larger bandwidths are adopted. Therefore, the adaptive bandwidth can be computed as:

$$h_i = h_0 \cdot \left(\frac{\hat{f}(x_i)}{\hat{f}_{\text{global}}}\right)^{-\alpha} \quad (5)$$

where  $h_0$  is the initial global bandwidth;  $\hat{f}(x_i)$  is the local density estimate at data point;  $\hat{f}_{\text{global}}$  is the global density estimate; and  $\alpha$  is the adjustment parameter, typically  $\alpha$  is set to 0.5.

If the local density around a point  $x_i$  is high (indicating a dense concentration of data points), the bandwidth  $h_i$  decreases to capture finer local details. Conversely, when the local density is low,  $h_i$  increases to allow for smoother estimation in sparser regions.

Thus, by incorporating the equation (4), the ABKDE can be expressed as:

$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^n \frac{1}{h_0 \left( \frac{\hat{f}(x_i)}{f_{\text{global}}} \right)^{-\alpha}} K \left( \frac{x-x_i}{h_0 \left( \frac{\hat{f}(x_i)}{f_{\text{global}}} \right)^{-\alpha}} \right) \quad (6)$$

### 2.3. Maintenance strategy optimization

#### 2.3.1. Imperfect preventive maintenance

Imperfect preventive maintenance (PM) can partially restore a system's functionality, while it cannot return the state to its original condition, as shown in Fig. 5 [4].

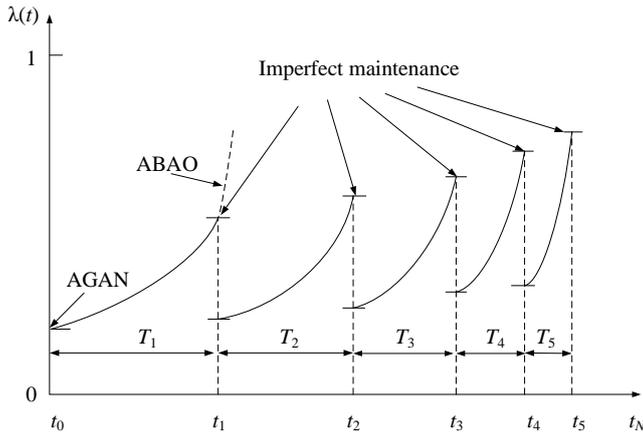


Fig. 5. Different types of imperfect maintenance.

To describe the changes in failure rate after imperfect PM, the concept of the age reduction factor is introduced. This factor ranges from [0, 1]: “0” means the system is restored to an “as-bad-as-old (ABAO)” condition which indicates it returns to state just before failure; “1” represents an “as-good-as-new

(AGAN)” condition, meaning the system is fully restored to its original state. The actual value of the age reduction factor can reflect the effectiveness of the imperfect PM. Typically, compared to the pre-maintenance levels, after an imperfect PM, the failure rate is decreased while it remains higher than that of a new system, i.e. somewhere between ABAO and AGAN.

Traditional PM strategies are popularly implemented for industrial equipment, as illustrated in Fig. 6. It assumes that the manufacturer provides warranty service for  $Q$  units of equipment over  $W_T$  units of time. It means that the manufacturer is responsible for the maintenance or replacement services within the  $W_T$  warranty period. When a failure occurs between two PM activities, only minimal repair (MR) is performed to restore it.

At  $t=0$ , the equipment starts operation. The first PM is scheduled at time  $T_1$ , followed by maintenance operation at  $T_2, T_3, \dots, T_k$  (where  $k=1, 2, 3, \dots, K$ ) respectively.  $K$  denotes the total number of PM activities performed. The decision variable  $T_k$  denotes the time interval between the end of the  $(k-1)$ -th PM and the beginning of the  $k$ th PM. Each PM activity lasts for a duration of  $T_{pm}$  and incurs a cost of  $C_{pm}$ ; each MR lasts for a duration of  $T_{cm}$  and costs  $C_{cm}$ . Therefore, the total time for  $k$ th PM cycle can be defined as the sum of  $T_k$  and  $T_{pm k}$ , and it represents the time interval between the end of the  $(k-1)$ -th PM and the completion of the  $k$ -th PM.

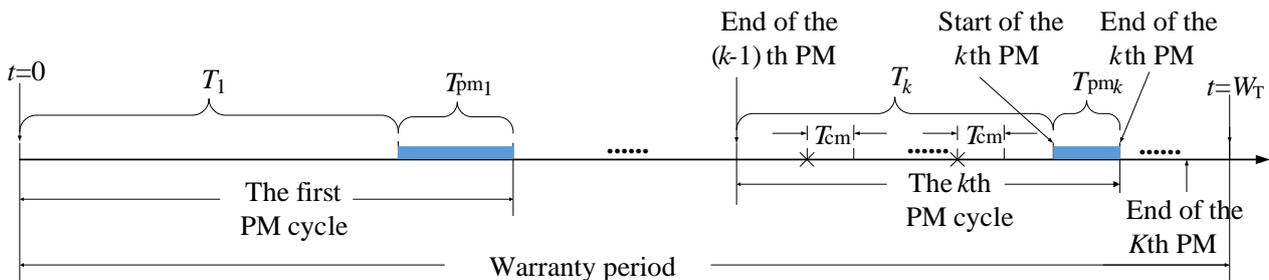


Fig. 6. The traditional PM strategy under the warranty period.

#### 2.3.2. Predictive maintenance

Up to now, the RUL predictions has not been integrated into maintenance planning in most studies. In this study, a PdM strategy is proposed based on interval RUL predictions, as shown in Fig. 7. As illustrated in Fig. 7, the PDF of RUL prediction can be categorized into two scenarios (Fig. 7 (a) and Fig. 7 (b)). In each scenario, the red dashed line denotes actual RUL of equipment, while blue line denotes the average RUL prediction based on the interval estimate. The pink shaded area

denotes the 95% confidence interval of RUL distribution. The red “x” marks the point where actual RUL reaches zero, signaling equipment failure. In scenario Fig. 7 (a), the predicted RUL is relatively large, and it exceeds the interval until the next PM or even the subsequent PMs. In this case, the PM operation proceeds as scheduled. In scenario Fig. 7 (b), the predicted RUL is small, and it falls short of the interval until the next scheduled PM. It suggests that following the traditional PM strategy (as depicted in Fig. 6) may result in equipment failure before the

next planned maintenance, and it will cause significant loss. Therefore, when dangerous signals are detected based on the

RUL predictions, adjusting the PM interval can help to reduce maintenance costs.

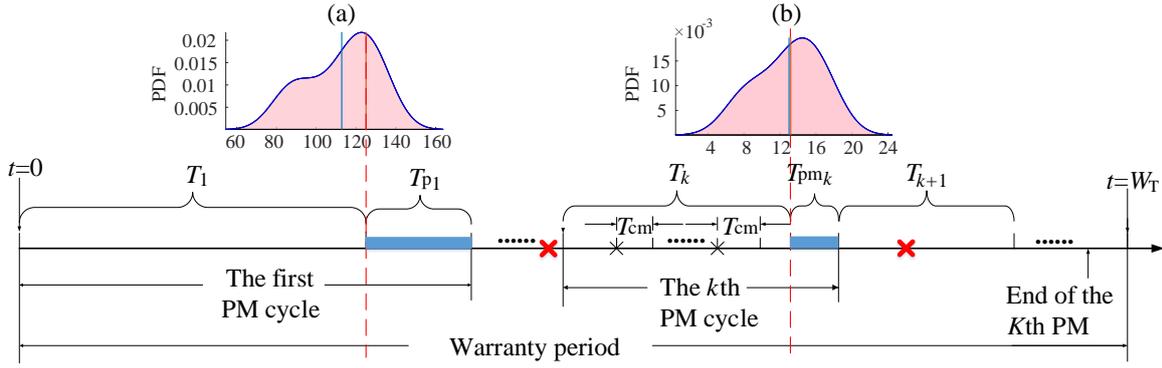


Fig. 7. The proposed PdM strategy under the warranty period.

Moreover, the level of PM plays a crucial role in determining maintenance effectiveness. In this study, when performing maintenance activities, five levels of maintenance can be selected, ranging from level 1 to level 5, and each of them corresponds to a different degree of equipment restoration. Specifically, level 1 maintenance can restore the equipment by 20%, while level 5 denotes a complete replacement of the equipment. Therefore, the choice of maintenance level is also one of the variables that need to be optimized.

### 2.3.3. Physical model

The physical model refers to the failure rate and lifetime distribution model of the equipment. In maintenance decision-making, the physical model determines the frequency of failures during the equipment's operation. Here Weibull distribution is used as a statistical method to analyze the run-to-failure behavior of the equipment. In detail, the Weibull distribution function along with shape parameter  $\beta$  and scale parameter  $\alpha$  as follows [35]:

$$h(t) = \left(\frac{\beta}{\alpha}\right) \left(\frac{t}{\alpha}\right)^{\beta-1} \quad (7)$$

Considering that the item's virtual age will decrease to some extent and the hazard rate will increase after PM action, the hybrid hazard rate model is introduced. The hazard rate function can be described as [41]:

$$h_{i+1}(t) = b_i h_i(t + a_i T_i), t \in [0, T_{i+1}] \quad (8)$$

where  $b_i$  denotes hazard rate adjustment factor; and  $0 \leq b_1 < b_2 < \dots < b_i < \dots$ ;  $a_i$  denotes age reduction factor; and  $h_i(t)$  is hazard rate before the  $i$ th PM:

$$\begin{aligned} h_i(t) &= b_{i-1} h_{i-1}(t + a_{i-1} T_{i-1}) \\ &= b_{i-1} b_{i-2} h_{i-2}(t + a_{i-2} T_{i-2} + a_{i-1} T_{i-1}) \\ &= \dots \\ &= b_{i-1} b_{i-2} \dots b_1 h_1(t + a_1 T_1 + a_2 T_2 + \dots) \\ &= \prod_{i=1}^{i-1} b_i h_1(t + \prod_{i=1}^{i-1} a_i T_i), \\ &t \in [0, T_i] \end{aligned} \quad (9)$$

Moreover, the PM actions performed in the maintenance schedule is non-periodic. For each PM interval, there is a quasi-update coefficient corresponding to it, i.e.  $\gamma_1, \gamma_2, \dots, \gamma_{i-1}$ . The maintenance interval of  $i$ th PM can be defined as:

$$T_i = \gamma_{i-1} T_{i-1} = \gamma_{i-1} \gamma_{i-2} T_{i-2} = \dots = \gamma_{i-1} \gamma_{i-2} \dots \gamma_2 \gamma_1 T_1, i = 1, 2, \dots, I \quad (10)$$

To facilitate the modeling and implementation of maintenance schedule, it is assumed that  $\gamma_1 = \gamma_2 = \dots = \gamma_{i-1} = \gamma$ . Thus, the maintenance interval (i.e.  $T_i$ ) of the  $i$ th PM can be deduced as:

$$T_i = \gamma T_{i-1} = \gamma^2 T_{i-2} = \dots = \gamma^{i-1} T_1, i = 1, 2, \dots, I \quad (11)$$

Besides, to simplify the calculation of the hazard rate model, it is assumed that the virtual age reduction factor and hazard rate adjustment factor in each PM action are supposed to be constants, i.e.  $a_1 = a_2 = \dots = a_{i-1} = a$ ;  $b_1 = b_2 = \dots = b_{i-1} = b$ , respectively.

Combined with equation (7), the hazard rate function in the  $i$ th imperfect PM interval can be expressed as:

$$\begin{aligned} h_i(t) &= b^{i-1} h_1(t + a(T_1 + T_2 + \dots + T_{i-1})) \\ &= b^{i-1} h_1\left(t + a \left(\frac{1-\gamma^{i-1}}{1-\gamma}\right) T_1\right) \end{aligned} \quad (12)$$

Therefore, the number of failures during the  $i$ th imperfect PM interval can be derived as:

$$H_i(T_i) = \int_0^{T_i} h_i(t) dt \quad (13)$$

where  $H_i(T_i)$  is number of failures during the  $i$ th imperfect PM

interval;  $h_i(t)$  is hazard rate prior to the  $i$ th PM.

### 2.3.4. Total maintenance cost

Based on the above maintenance schedule, the total number of PM actions (i.e.  $I$ ) can be deduced as:

$$I = \max\{i \mid \sum_{j=1}^i (T_j + T_{pmj}) < L\} \quad (14)$$

and the length of the  $I+1$  PM interval (i.e.  $T_{I+1}$ ) is denoted as:

$$T_{I+1} = L - \sum_{i=1}^I (T_i + T_{pmi}) \quad (15)$$

Moreover, the total warranty cost (i.e.  $C_T$ ) consists of maintenance cost (i.e.  $C_M$ ) and downtime cost (i.e.  $C_D$ ) during the warranty period, and it can be defined as:

$$C_T = C_M + C_D \quad (16)$$

The maintenance cost includes PM cost (i.e.  $C_P$ ) and the MR cost (i.e.  $C_R$ ), which can be expressed by:

$$C_M = C_P + C_R \quad (17)$$

Moreover,  $C_P$  can be calculated by:

$$C_P = \sum_{i=1}^I C_{pmi} T_{pmi} \quad (18)$$

where  $C_{pmi}$ ,  $T_{pmi}$  are maintenance cost and time of the  $i$ th PM action.

$C_R$  can be obtained with:

$$C_R = C_{cm} H T_{cm} \quad (19)$$

where  $C_{cm}$  is the maintenance cost per MR action;  $H$  is the system's total number of failures during total maintenance schedule, which can be expressed as:

$$H = \sum_{i=1}^{I+1} H_i(T_i) \quad (20)$$

where  $H_i(T_i)$  is number of failures during  $i$ th PM interval;  $I$  is total number of PM actions;  $I+1$  is the last PM to the end of warranty.

In addition, PM and MR actions will incur shutdown of system and downtime loss (i.e.  $C_D$ ).  $C_D$  can be calculated as:

$$C_D = C_0 (\sum_{i=1}^I T_{pmi} + H T_R) \quad (21)$$

where  $C_0$  is the downtime cost per unit of shutdown time.

### 2.4. DCS optimization algorithm

The flowchart of the DCS process is shown in Fig. 8[36]. Inspired by differentiated knowledge acquisition and creative realism, the DCS algorithm combines knowledge acquisition with creative realism to optimize strategies. It adopts a dual-strategy approach, balancing divergent and convergent thinking, and uses differentiated knowledge acquisition based on individual performance to foster continuous learning and adaptability.

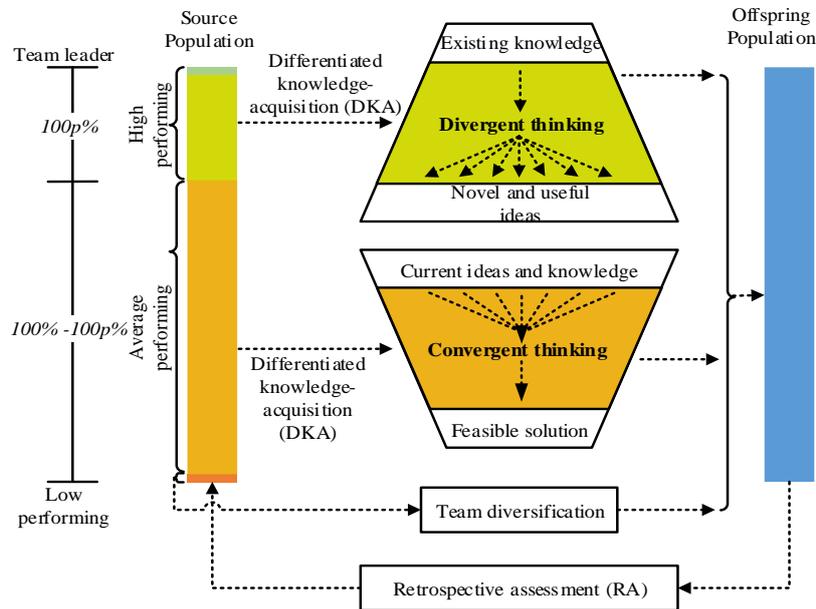


Fig. 8. The flowchart of the DCS.

In this paper, the optimization process using the DCS algorithm is divided into two parts: the optimization of the hyperparameters of the deep learning BiTCN-MHA model for RUL prediction, and the optimization of the maintenance intervals for PdM strategies. Therefore, two optimization models are established as follows.

#### (1) Hyperparameter optimization model

$$\begin{aligned} \min_{lr, n, l2} \quad & MSE = \frac{1}{n} \sum_{i=1}^n (y_p^i - y_t^i)^2 \\ \text{s.t.} \quad & lr \in [10^{-5}, 10^{-1}], \\ & n \in \mathbb{Z}^+, \\ & l2 \geq 0 \end{aligned} \quad (22)$$

where  $MSE$  represents the mean squared error during the training process;  $N$  is the number of samples;  $y_i$  is the true label

of the  $i$ th sample;  $\hat{y}_i$  is the predicted value for the  $i$ th sample;  $lr$  is the learning rate;  $n$  is the number of filters;  $l2$  is the regularization coefficient.

The optimal objective is to minimize the loss during the training process. Additionally, it is essential to ensure that the model achieves high prediction accuracy and robustness.

(2) PdM optimization model

$$\min C_T(T_i, L_i) \quad \text{s.t.} \quad \begin{cases} T_i > 0 \\ L_i \in M \end{cases} \quad (23)$$

where  $C_T$  is the total cost of the proposed PdM strategy;  $T_i$  is the

$i$ th maintenance interval of PdM activity;  $L_i$  is the  $i$ th maintenance level of PdM activity; the set  $M$  represents the available options of PdM levels.

The optimal objective is to minimize the total maintenance cost, including both the preventive and corrective maintenance costs.

## 2.5. Overall framework of the proposed model

As a summary, the framework of proposed DCS-BiTcN-PdM method is illustrated in Fig. 9.

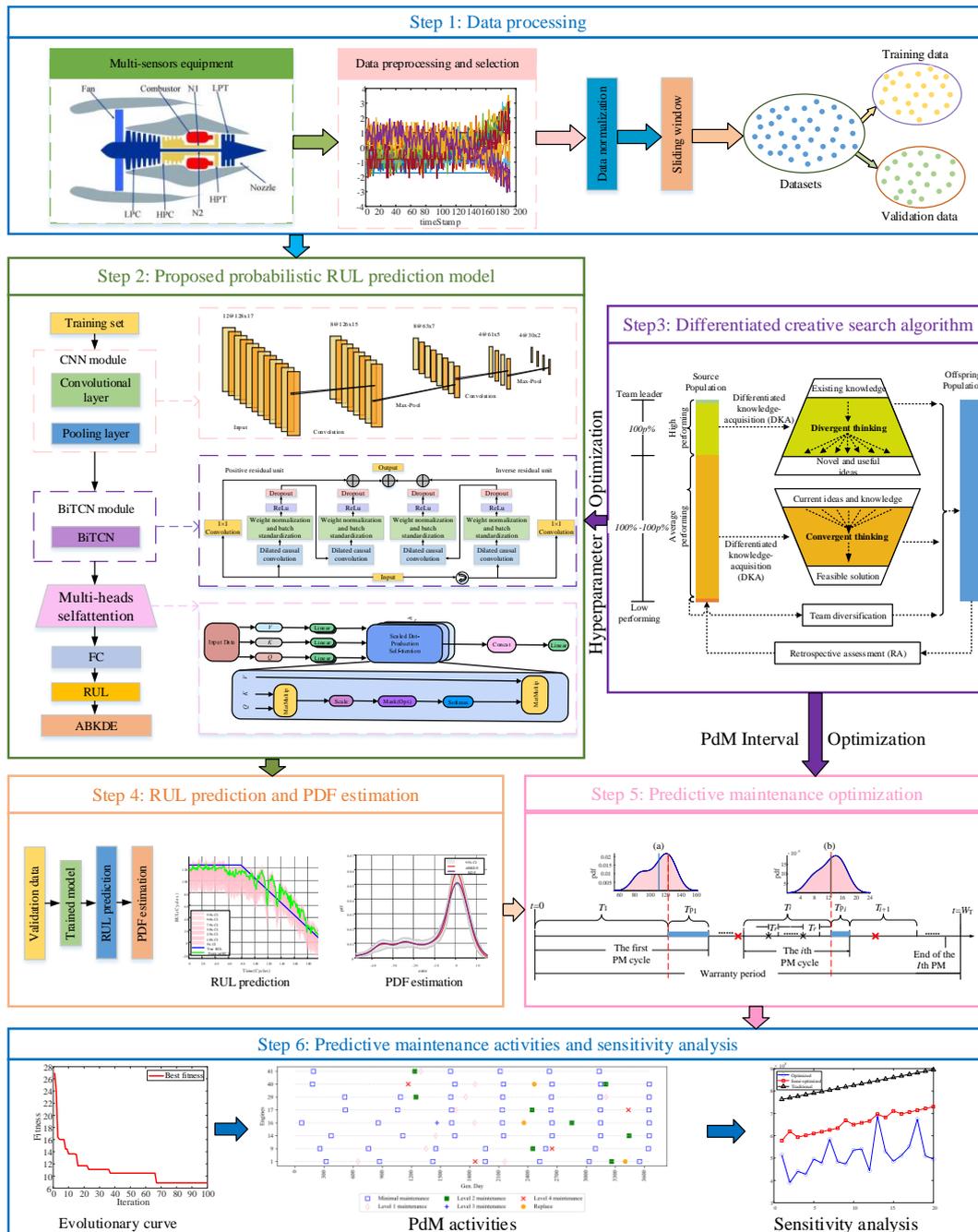


Fig. 9. The framework of the proposed DCS-BiTcN-PdM method.

In it, the training process begins with normalizing and partitioning the time series data, which is then input into the BiTCN model to extract local features. Moreover, an MHA mechanism captures long-term dependencies; the DCS algorithm is applied to obtain optimal hyperparameters in the BiTCN-MHA model, and the ABKDE method is applied to estimate the PDF. Finally, the RUL predictions are incorporated into the PdM, with the aim of minimizing total maintenance cost through an optimization algorithm. In order to provide a robust solution for PdM, deep learning is combined with the optimization techniques in the proposed framework.

### 3. Case study

To evaluate the effectiveness of proposed DCS-BiTCN-PdM method, numerical experiments are conducted on the C-MAPSS dataset of aero engines, and the results are compared with some Table 1. Description of the C-MAPSS dataset.

Dataset Description	Number of engines for training	Number of engines for testing	Minimum cycle for training	Minimum cycle for testing
FD001	100	100	128	31
FD002	260	259	156	21
FD003	100	100	145	38
FD004	249	248	128	19

of the recently published methods.

### 3.1. Dataset description

In the case study, the dataset used is provided by NASA, and it is a part of the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) [42]. Overall it consists of four subsets, each of which represents different operational conditions and failure modes of a turbofan engine. Moreover, each subset includes a training set, a test set, and a labelled prediction result set. The training set comprises aero engines of the same type with varying levels of initial wear and installation faults. The sensor data can capture the complete lifecycle of each engine, from start-up to final failure. The test set contains partial lifecycle data fragments from same engines. More detailed information about the dataset is presented in Table 1.

### 3.2. Evaluation indicators

In this study, two types of metrics are used to evaluate the predictive performance of the model. They are root mean square error (RMSE) and mean absolute percentage error (MAPE), respectively. Their definitions are as follows:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_p^i - y_t^i)^2} \quad (24)$$

$$MAPE = \frac{1}{n} \sum_{i=1}^n \frac{|y_p^i - y_t^i|}{y_t^i} \times 100\% \quad (25)$$

where  $n$  denotes the total number of samples in the testing set;  $y_p^i, y_t^i$  denote the predicted and true RUL values for the  $i$ -th sample, respectively.

### 3.3. Hyperparameter and maintenance activity set

The settings of some general hyperparameters and the structure of BiTCN-MHA are presented in Table 2.

Table 2. Settings of the hyperparameters and structure of BiTCN-MHA.

Hyperparameter	Value
<b>Hyperparameter -architecture</b>	
Window size	30
Batch size	256
Convolutional layers	2
Number of filters	$n$
Kernel size	5
Spatial Dropout rate	0.15
Layer number of BiTCN layer	2
<b>Hyperparameter -optimization</b>	
Number of epochs	500
Initial learning rate	$lr$
L2 regularization coefficient	$l2$
Dropout rate	0.1
Training-Validation split	7:3

Note: The  $n, lr$  and  $l2$  are hyperparameters to be optimized by DCS.

The software of MATLAB 2024a® is used for training and testing deep learning of the proposed method. The programs are

run on a computer with the following configuration: Intel(R) Core(TM) i5-12400F CPU@ 3.5GHz, NVIDIA RTX 4060

Table 3. Parameters of maintenance cost and time.

$C_{cm}(\$/\text{day})$	$C_{pm\text{level}}(\$/\text{day})$					$T_{cm}(\text{day})$	$T_{pm\text{level}}(\text{day})$				
	1	2	3	4	5		1	2	3	4	5
2700	80	250	400	620	750	3	5	8	10	13	20

The parameters of Weibull distribution can be estimated through various techniques. Among them, maximum likelihood estimation is popularly used due to its numerical stability. By using the above method, the parameters of the dataset are estimated, with the obtained shape parameter  $\beta=4$  and scale parameter  $\alpha=225$ .

### 3.4. Results and discussion

#### 3.4.1. DCS evolutionary curve

The hyperparameters in the BiTCN-MHA model are optimized by using the DCS algorithm. In detail, the corresponding parameters are set in the Table 4.

Table 4. Hyperparameters of BiTCN-MHA model.

Hyperparameters of BiTCN-MHA	Range
Learning rate	$[10^{-5}, 10^{-1}]$
Number of filters,	$[10, 100]$
Regularization parameter	$[10^{-6}, 10^{-2}]$

The optimal objective is to minimize the model's training loss, with focuses on the learning rate, number of filters, and regularization parameter. Moreover, the maximum number of iterations is set to 1000. The evolutionary curve is illustrated in Fig. 10 and the results of optimization is show in Table 5.

Table 5. Optimized hyperparameters of BiTCN-MHA model.

Hyperparameters of BiTCN-MHA	value
Learning rate	$10^{-3}$
Number of filters,	64
Regularization parameter	$10^{-4}$

Graphics Card, 32GB RAM. Moreover, the parameters of maintenance cost and time are shown in Table 3.

As shown in Fig. 10 and Table 5, the fitness of the optimal objective decreases gradually with the increase of the iteration and eventually converges to a lower value, and the best optimized hyperparameters are obtained. The trend indicates that the DCS optimization can effectively minimize the error in the RUL predictions generated by the BiTCN-MHA model, thereby the prediction accuracy is improved and the optimal hyperparameters is identified.

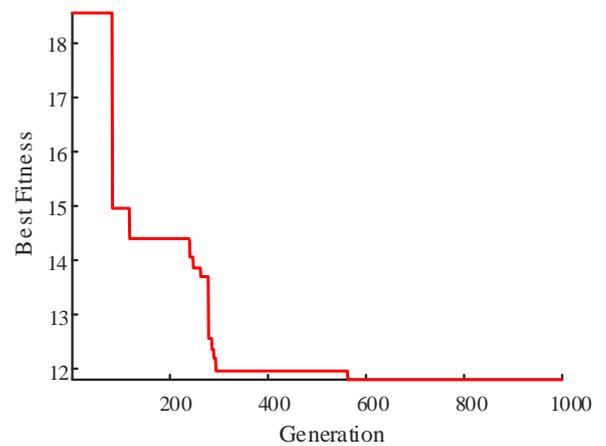
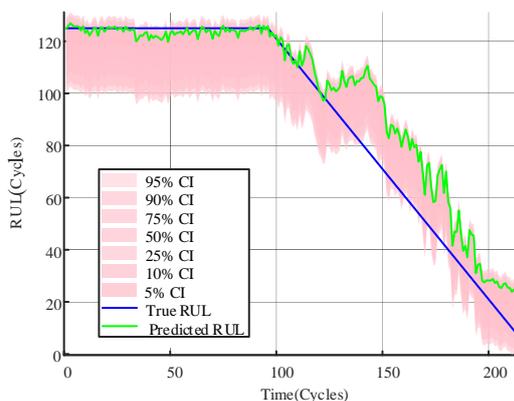


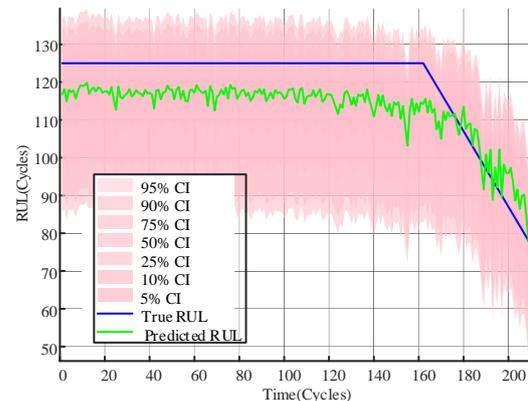
Fig. 10. DCS evolutionary curve of the BiTCN-MHA.

#### 3.4.2. RUL prediction results

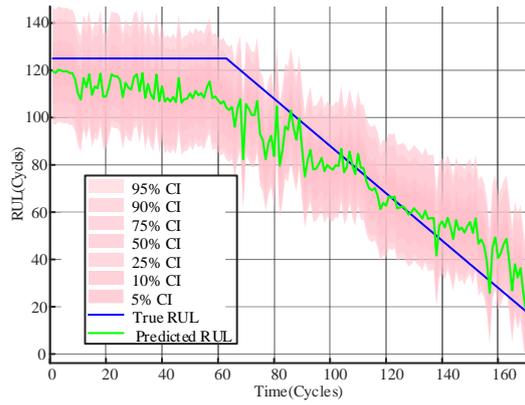
The hyperparameters optimized by using DCS are applied to the BiTCN-MHA model, and then they are re-trained and tested. Due to the space limitations, Fig. 11 presents only the RUL prediction results for four randomly selected aero-engines from the datasets FD001 to FD004.



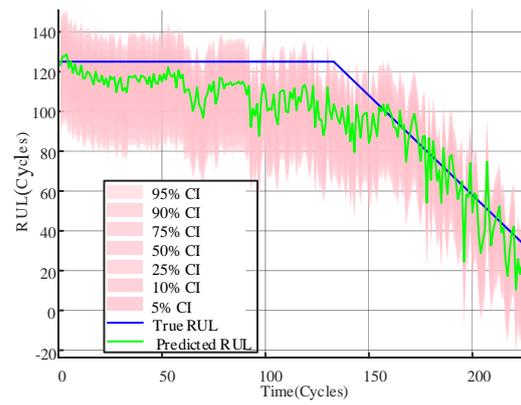
(a) FD001 #81



(b) FD002 #24



(c) FD003 #79



(d) FD004 #47

Fig. 11. RUL prediction results of C-MAPSS aero-engines.

In addition, the four datasets also provide a comparison between the actual RUL (green line) and the predicted RUL (blue line), as shown in Fig. 11. Overall, the blue lines closely follow the green lines. It demonstrates that the prediction ability of the proposed model is quite accurate across different datasets. Detailly, the pink areas represent the confidence intervals, where wider intervals indicate higher prediction uncertainty. In the early stages, the confidence intervals are narrow and they will gradually widen over time. This widening becomes more pronounced when the RUL starts to decline sharply. It indicates

that the model's uncertainty increases as the equipment nears the end of its lifespan. Additionally, for the FD003 dataset, the confidence interval is comparatively wider, and it indicates a relatively higher level of prediction uncertainty for this dataset.

To further verify the effectiveness of the proposed model, a performance comparison is conducted with several state-of-the-art (SOTA) methods. The comparison assesses the prediction results in terms of RMSE and MAPE, with the findings listed in Table 6.

Approaches	RMSE				MAPE			
	FD001	FD002	FD003	FD004	FD001	FD002	FD003	FD004
Auto encoder(2019) [16]	14.74	22.07	17.48	23.49	-	-	-	-
BiGRU (2019) [17]	-	26.54	-	29.13	-	-	-	-
CNN+LSTM (2019) [43]	16.13	20.46	17.12	23.26	15.34%	21.31%	<u>11.50%</u>	<u>15.21%</u>
MCLSTM (2021) [14]	13.71	-	-	23.81	-	-	-	-
BiGRU-TSAM (2022) [44]	12.56	18.94	12.45	20.47	14.95%	<u>20.50%</u>	<b>11.01%</b>	<b>15.02%</b>
CACNN+Transformer (2022) [45]	12.25	<u>17.08</u>	13.39	19.86	-	-	-	-
Transformer (2023) [46]	13.32	19.83	13.92	21.88	-	-	-	-
PI-DCNN+LSTM (2023) [47]	<b>10.45</b>	19.43	<b>11.30</b>	<u>19.03</u>	-	-	-	-
BiTCN-MHA (Proposal)	12.22	17.12	13.24	19.42	<u>14.29%</u>	21.23%	18.23%	23.32%
DCS-BiTCN-MHA (Proposal)	<u>11.78</u>	<b>16.55</b>	<u>12.80</u>	<b>18.33</b>	<b>12.86%</b>	<b>19.27%</b>	17.46%	22.03%

Table 6. Results of the proposed method and the SOTA methods.

It can be observed that the proposed DCS-optimized BiTCN-MHA model demonstrates superior performance. Detailly, it can achieve four optimal metrics and two sub-optimal metrics. The other metrics also show a favorable performance level. Furthermore, compared to the SOTA methods, the optimal performance of the proposed model shows a reduction of 3.20% at least in RSME on FD002 and 3.68% at least in RSME on FD004. Additionally, compared to the BiTCN-MHA model without DCS optimization, the RMSE and

MAPE for the DCS-BiTCN-MHA model are reduced by 5.94% and 11.11%, respectively. The above results confirm the proposed model's efficiency and superiority.

To validate the effectiveness and computational cost/time of the proposed framework, as referenced in [48, 49], we conducted RUL prediction experiments using five models: BiTCN, CNN-LSTM, Transformer, and the proposed methods (BiTCN-MHA and DCS-BiTCN-MHA). Each experiment was repeated five times, and the average results are listed in Table 7.

Table 7. RUL prediction results under five different approaches.

Approaches	Training time [s]	Inferring time [s]
BiTCN	289.23	8.24
CNN-LSTM	272.84	7.98
Transformer	321.45	8.92
BiTCN-MHA (Proposal)	340.27	9.26
DCS-BiTCN-MHA (Proposal)	334.86	9.15

As shown in the results of Table 7, compared to the BiTCN and CNN-BiLSTM methods, the proposed method increases the model training time by approximately 50s. However, compared to the Transformer method, the increase in training time of the proposed model is negligible. After the model training, the

inferring time for all five methods does not show significant differences, remaining within 10s. These results indicate that the proposed method has high time efficiency in practical applications.

Moreover, to validate the robustness of the proposed method, RUL prediction is performed under limited data samples [50, 51]. Taking the subset FD001 as an example, where the training and testing samples are 100 each, we first set the training samples to 10 and train and predict using the proposed DCS-BiTCN-MHA model. Then, the number of training samples is increased by 10 at each step, until reaching 100. The prediction performance of the proposed method under limited samples is shown in Figure 12.

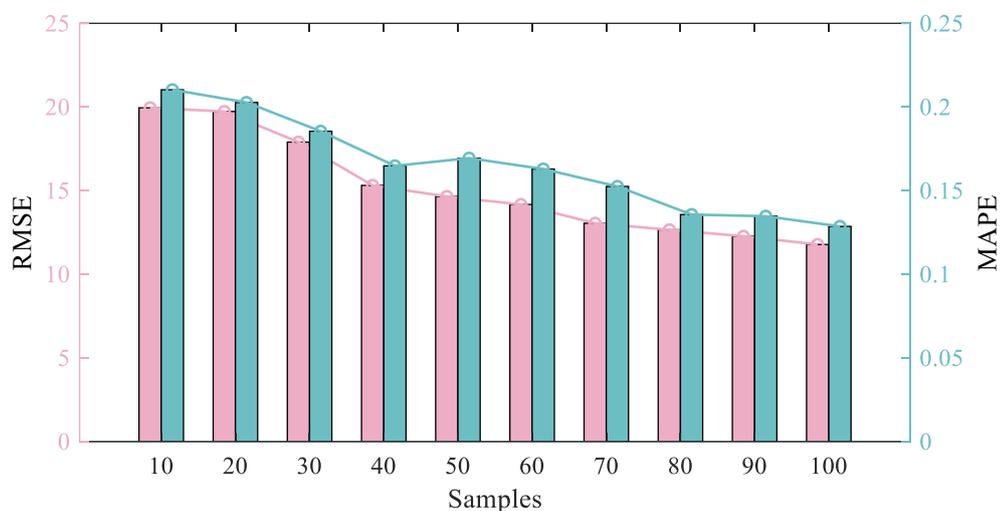


Fig. 12. Prediction performance of the DCS-BiTCN-MHA model under limited samples.

As shown in Fig. 12, the experimental results indicate that the DCS-BiTCN-MHA method demonstrates high prediction performance for each available sample size, with RMSE values all below 20. As the number of available samples in the training set increases, both the RMSE and MAPE predicted by the DCS-BiTCN-MHA model show a decreasing trend. Furthermore, when 70 training samples are available, the proposed method achieves an RMSE of 13.05, fully demonstrating the superiority of the model.

### 3.4.3. Adaptive bandwidth kernel density estimation

In this paper, the ABKDE is used to evaluate the kernel density estimation of RUL predictions. Compared to the fixed bandwidth KDE, the ABKDE gains a more adaptive and flexible estimation, as shown in Fig. 13.

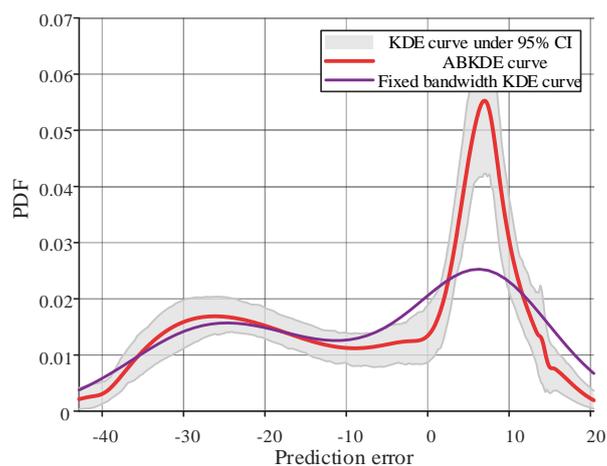


Fig. 13. The kernel density estimation curves.

It can be found that the prediction error is concentrated near 0, and it indicates that the model's overall predictive performance is relatively accurate. Meanwhile, the ABKDE curve follows the actual error distribution more closely in the

main error region, while the fixed bandwidth KDE curve shows fluctuations in the high-error region.

To quantify uncertainty, several key metrics are introduced: prediction interval normalized average width (PINAW), prediction interval coverage probability (PICP), and continuous ranked probability score (CRPS). Among them, PINAW is used to measure the width of the prediction interval and normalize it for comparability across different models or datasets; PICP is

applied to demonstrate the reliability of the prediction interval by indicating the proportion of actual values that fall within it, and CRPS is employed to evaluate the closeness of the predicted probability distribution to the actual observed values by comparing the predicted distribution to the real outcomes. As a whole, the above metrics are used to assess the uncertainty in RUL prediction results. The results are illustrated in Table 7 and Fig. 14 respectively.

Table 7 PINAW, PICP and CRPS of  $\alpha$  for RUL prediction for C-MAPSS training sets.

$\alpha$	Index	FD001	FD002	FD003	FD004
0.05	PINAW	0.003	0.0186	0.013	0.013
	PICP	0.005	0.0190	0.029	0.049
	CRPS	0.083	0.175	0.084	0.099
0.10	PINAW	0.006	0.037	0.026	0.029
	PICP	0.009	0.029	0.052	0.107
	CRPS	0.083	0.178	0.084	0.101
0.25	PINAW	0.043	0.096	0.069	0.089
	PICP	0.085	0.062	0.201	0.272
	CRPS	0.083	0.181	0.085	0.104
0.50	PINAW	0.100	0.329	0.139	0.330
	PICP	0.300	0.367	0.394	0.584
	CRPS	0.084	0.199	0.087	0.132
0.75	PINAW	0.164	0.647	0.241	0.452
	PICP	0.563	0.943	0.600	0.683
	CRPS	0.086	0.234	0.089	0.146
0.90	PINAW	0.215	0.914	0.383	0.565
	PICP	0.803	0.990	0.941	0.799
	CRPS	0.088	0.265	0.093	0.158
0.95	PINAW	0.244	1.064	0.455	0.619
	PICP	0.887	0.995	0.982	0.830
	CRPS	0.089	0.282	0.096	0.166

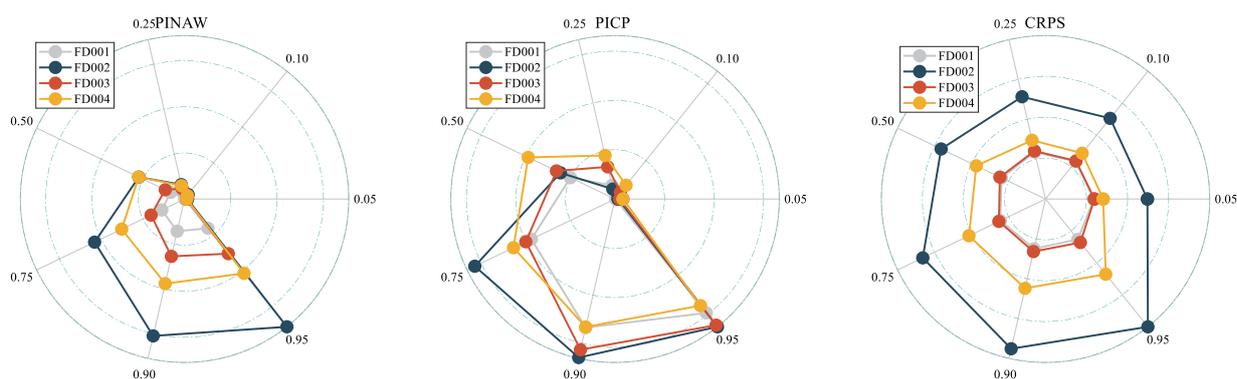


Fig. 14. The radar diagram of PINAW, PICP and CRPS.

As shown in Table 5 and Fig. 14, when  $\alpha=5\%$ , the PINAW value is relatively small. It indicates a narrow prediction interval and lower prediction uncertainty. The PICP value is slightly below the expected 5%. It suggests that the model may not fully cover the actual values within the 5% confidence level. With the

increase of  $\alpha$ , all of PINAW, PICP and CRPS show an upward trend. At the 95% confidence interval, the PINAW value is higher, and it reflects a wider prediction interval. The PICP value is close to 1, which means that most actual values fall within the prediction interval. In addition, the CRPS remains at

a relatively low value, indicating that the predicted RUL distribution is close to the actual values. Hence, the proposed model performs well at higher confidence levels.

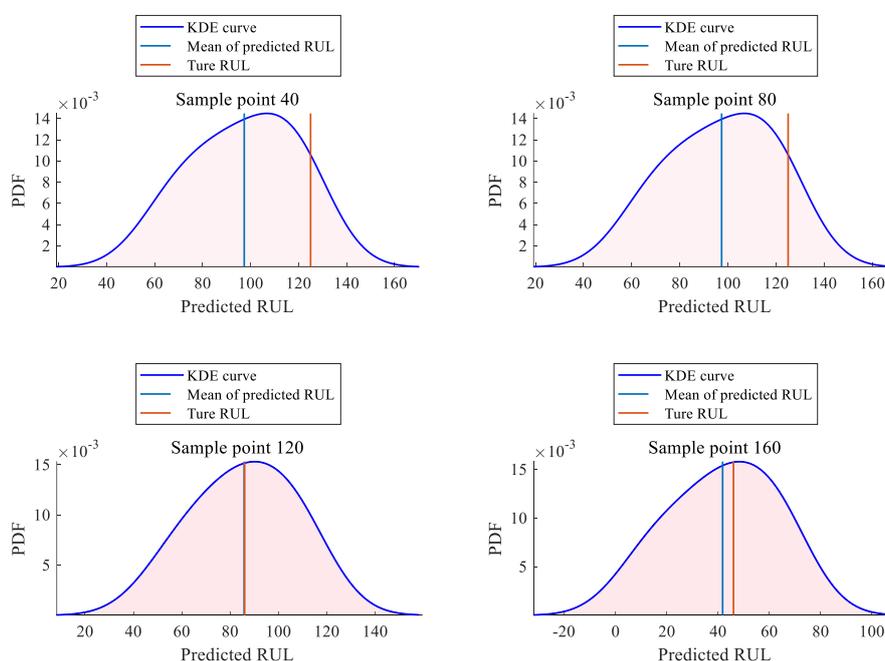


Fig. 15. The predicted PDF of engine 24's RUL in subset FD001.

As shown in Fig. 15, the PDF of engine 24's RUL is concentrated around the actual RUL at all four moments, with predicted RUL generally being slightly lower than the actual RUL. As the actual RUL decreases, the average predicted RUL becomes increasingly aligned with the actual value. It indicates that by using the predicted interval RUL, the equipment's operational status can be assessed promptly, therefore maintenance activities can be performed proactively to prevent possible failures.

### 3.4.4. Predictive maintenance optimization

Considering that the aircraft engines in the dataset had already been running for some time before the data collection, it is essential to take into account this factor when conducting actual industrial applications in optimizing maintenance activities. Therefore, we assume that at the beginning of the dataset, the engine has already been running for  $k$  cycles. It implies that the actual life of the aircraft engine is  $k + \text{RUL}$ . The predictive maintenance optimization is then performed on engines 24, 29, 34, 56, and 67 from the FD001 dataset, with an assumption of a ten-year warranty period (3650 days) for the aero engines.

The PM intervals of the PdM strategy are optimized by using

Fig. 15 shows the PDF of predicted RUL of engine 24 in the FD001 subset (as an example) for operating times of 40, 80, 120, and 160 flight cycles, respectively.

the DCS algorithm, with the evolutionary curve illustrated in Fig. 16. The objective function is to minimize the total maintenance cost, with the focus on the  $i$ th PM interval and PM level, which are set within the ranges of (100, 1000) and (1,5). The maximum number of the iterations is set to 1000.

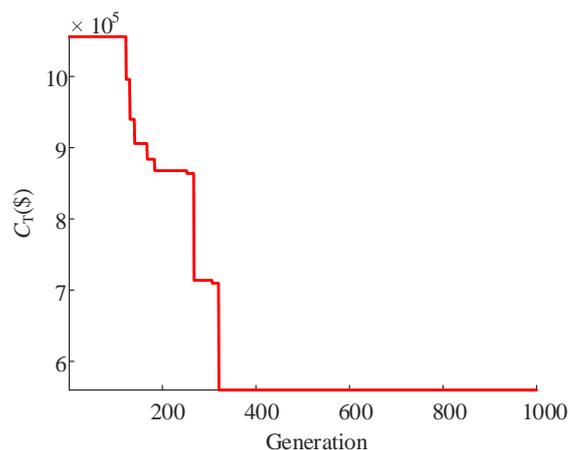


Fig. 16. The evolutionary curve of the PdM.

It can be seen from Fig. 16 that the fitness of the objective function decreases with the increasing number of iterations and eventually converges to a smaller value. This trend indicates that the DCS optimization can reduce total maintenance cost in the PdM strategy.

Moreover, the details of the maintenance plan of optimized PdM are shown in Fig. 17.

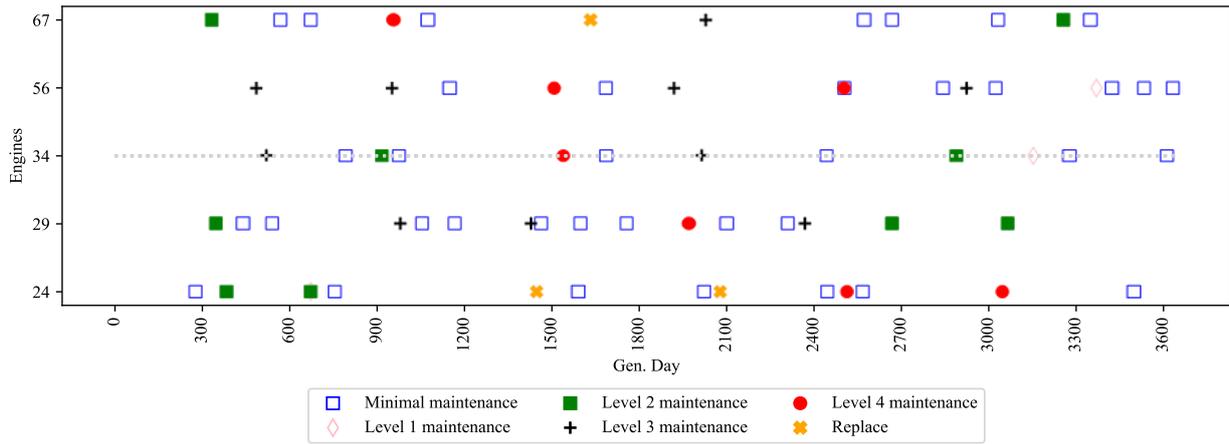


Fig. 17. The detail maintenance plan of optimized PdM strategy.

From Fig. 17, it can be found that all devices undergo multiple MR activities due to sudden failures caused by environmental factors, component malfunctions, or other electrical issues. These failures are inevitable, and it requires MR to restore the device to an ABAO state. Meanwhile, in the early stages of the equipment operation, level 2 and level 3 maintenance occur most frequently in the optimal maintenance plan. With the increase of the operating time, the frequency of level 4 repairs begins to rise. Additionally, the frequency of level 1 and level 5 maintenance remains the lowest. It can be explained by the fact that during the early stages of operation, the equipment has a relatively high RUL, therefore lower-level maintenance strategies can be employed. Meanwhile, level 1 maintenance is the least frequent. Furthermore, the replacement will incur fairly high maintenance cost for the aero engines. In order to minimize the number of replacements and restore the RUL to a higher level, the optimal PdM strategy suggests

scheduling one or two preventive maintenance actions in the early stage of the engine's operation.

#### 4. Sensitivity analysis

In this section, sensitivity analysis is conducted on the maintenance cost and time in the PdM strategy so as to guide the maintenance decision-makers. In addition, for comparison, three types of maintenance strategies are set up: (1) The proposed PdM strategy optimizes both the maintenance interval and level (called optimized strategy); (2) The maintenance interval is fixed and the maintenance level is optimized (called semi-optimized strategy); (3) The traditional PM strategy has pre-defined maintenance intervals and maintenance levels. Specifically, the maintenance cost and time of PM are set into 20 groups, with an increase of \$100 per group in maintenance cost and 0.3 days per group in maintenance time on the basis of the original values.

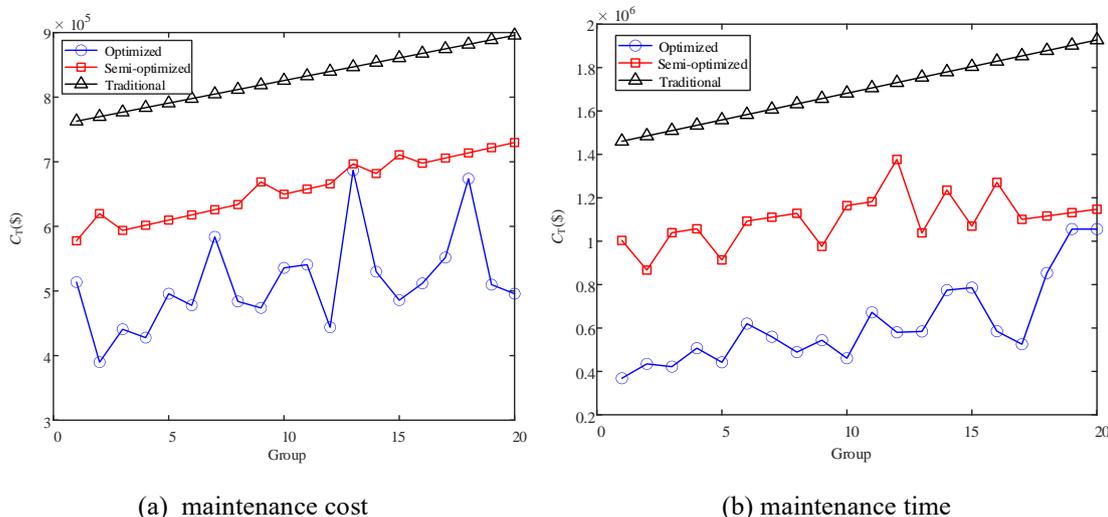


Fig. 18. Sensitivity analysis of the PdM strategy.

As shown in Fig. 18, the impact trends of maintenance cost and maintenance time on total maintenance cost show similar results. In detail, with the increase of maintenance cost or maintenance time, all three types of total maintenance costs show an increasing trend. At the same time, the traditional maintenance strategy has the highest cost, while the proposed PdM strategy has the lowest total cost, and the semi-optimized strategy has a total maintenance cost between the two. With the increase of maintenance cost and maintenance time, the total maintenance cost increases linearly. However, even with an increase in maintenance costs or time, a lower total cost may still be achieved in the proposed PdM strategy. This is because the intelligent DCS algorithm continuously optimizes the PdM strategy under varying conditions to attain the lowest possible total maintenance cost for the current scenario. Moreover, the total cost of proposed PdM strategy can be reduced by up to 47.15% and 33.34% compared to traditional PM and semi-optimized PdM strategy, respectively.

## 5. Conclusion

This study aims to address key challenges in RUL prediction for PdM by introducing a novel framework that integrates interval RUL prediction with maintenance policy optimization. Meanwhile, a case study was conducted using the C-MPASS

aircraft engine. Compared to some SOTA methods, the proposed model demonstrates at least a 3.20% improvement in RMSE on FD002 and a 3.68% improvement on FD004. Additionally, compared to the BiTCN-MHA model without DCS optimization, the DCS-BiTCN-MHA model reduces RMSE and MAPE by 5.94% and 11.11%, respectively. By using the predicted RUL interval as a constraint, a PdM model was constructed. Compared to traditional PM strategy and semi-optimized PdM strategy, the proposed optimized PdM strategy achieves the lowest total maintenance cost and offers more flexible maintenance plans. Detailly, the optimized PdM method reduces maintenance costs by 47.15% compared to traditional preventive maintenance and by 33.34% compared to semi-optimized PdM methods. Therefore, the proposed method can provide an optimal maintenance solution for PHM.

However, PdM optimization based on deep learning still faces several challenges. For example, accurate RUL prediction generally requires extensive historical data. Additionally, exploring more accurate and lightweight deep learning models remains crucial. Furthermore, predictive maintenance often needs to account for environmental and resource dependencies, and the increased complexity of these factors presents an ongoing challenge for PdM.

## Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant No. 72471055) and the China Scholarship Council.

## References

1. Zio E. Prognostics and health management (PHM): Where are we and where do we (need to) go in theory and practice. *Reliability Engineering & System Safety* 2022; 218: 108119, <https://doi.org/10.1016/j.ress.2021.108119>.
2. Fang P Y, Sui X X, Zhang A H, et al. Fusion model based RUL prediction method of lithium-ion battery under working conditions. *Eksploatacja i Niezawodność* 2024; 26(3): <https://doi.org/10.17531/ein/186537>.
3. Pinciroli L, Baraldi P, Zio E. Maintenance optimization in industry 4.0. *Reliability Engineering & System Safety* 2023; 234: 109204, <https://doi.org/10.1016/j.ress.2023.109204>.
4. Zhou Z, Bai M, Long Z, Liu J, Yu D. An adaptive remaining useful life prediction model for aeroengine based on multi-angle similarity. *Measurement* 2024; 226: 114082, <https://doi.org/10.1016/j.measurement.2023.114082>.
5. Levitin G, Yu L, Dai Y. Optimizing corrective maintenance for multistate systems with storage. *Reliability Engineering & System Safety* 2024; 244: 109951, <https://doi.org/10.1016/j.ress.2024.109951>.
6. Liu P, Wang G, Tan Z. Calendar-time-based and age-based maintenance policies with different repair assumptions. *Applied Mathematical Modelling* 2024; 129: 592-611, <https://doi.org/10.1016/j.apm.2024.02.013>.
7. Sharma J, Mittal M L, Soni G. Condition-based maintenance using machine learning and role of interpretability: A review. *International Journal of System Assurance Engineering and Management* 2024; 15(4): 1345-1360, <https://doi.org/10.1007/s13198-022-01843-7>.

8. Hong G, Song W, Gao Y, Zio E, Kudreyko A. An iterative model of the generalized Cauchy process for predicting the remaining useful life of lithium-ion batteries. *Measurement* 2022; 187: 110269, <https://doi.org/10.1016/j.measurement.2021.110269>.
9. Wang T, Li X, Wang W, Du J, Yang X. A spatiotemporal feature learning-based RUL estimation method for predictive maintenance. *Measurement* 2023; 214: 112824, <https://doi.org/10.1016/j.measurement.2023.112824>.
10. Wang Z, Zhao W, Li Y, Dong L, Wang J, Du W, Jiang X. Adaptive staged RUL prediction of rolling bearing. *Measurement* 2023; 222: 113478, <https://doi.org/10.1016/j.measurement.2023.113478>.
11. Nunes P, Santos J, Rocha E. Challenges in predictive maintenance—A review. *CIRP Journal of Manufacturing Science and Technology* 2023; 40: 53-67, <https://doi.org/10.1016/j.cirpj.2022.11.004>.
12. Cao X, Shi X, Zhao J, Duan Y, Yang X. Dynamic grouping maintenance optimization by considering the probabilistic remaining useful life prediction of multiple equipment. *Eksploatacja i Niezawodność* 2024; 26(3): <https://doi.org/10.17531/ein/187793>.
13. Truong T T, Airao J, Hojati F, Ilvig C F, Azarhoushang B, Karras P, Aghababaei R. Data-driven prediction of tool wear using Bayesian regularized artificial neural networks. *Measurement* 2024; 238: 115303, <https://doi.org/10.1016/j.measurement.2024.115303>.
14. Xiang S, Qin Y, Luo J, Pu H, Tang B. Multicellular LSTM-based deep learning model for aero-engine remaining useful life prediction. *Reliability Engineering & System Safety* 2021; 216: 107927, <https://doi.org/10.1016/j.res.2021.107927>.
15. Li T, Zhou Z, Li S, Sun C, Yan R, Chen X. The emerging graph neural networks for intelligent fault diagnostics and prognostics: A guideline and a benchmark study. *Mechanical Systems & Signal Processing* 2022; 168: 108653, <https://doi.org/10.1016/j.ymsp.2021.108653>.
16. Yu W, Kim IY, Mechefske C. Remaining useful life estimation using a bidirectional recurrent neural network based autoencoder scheme. *Mechanical Systems & Signal Processing* 2019; 129: 764-780, <https://doi.org/10.1016/j.ymsp.2019.05.005>.
17. Liu J Q, Pan C L, Lei F, et al. Fault prediction of bearings based on LSTM and statistical process analysis. *Reliability Engineering & System Safety* 2021; 214: 107646, <https://doi.org/10.1016/j.res.2021.107646>.
18. Zhou J H, Qin Y, Luo J, et al. Dual-thread gated recurrent unit for gear remaining useful life prediction. *IEEE Transactions on Industrial Informatics* 2023; 19(7): 8307-8318, <https://doi.org/10.1109/TII.2022.3217758>.
19. Wang L, Cao H, Xu H, Liu H. A gated graph convolutional network with multi- sensor signals for remaining useful life prediction. *Knowledge-Based Systems* 2022; 252: 109340, <https://doi.org/10.1016/j.knsys.2022.109340>.
20. Li X, Ding Q, Sun J. Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliability Engineering & System Safety* 2018; 172: 1-11, <https://doi.org/10.1016/j.res.2017.01.002>.
21. Yang B, Liu R, Zio E. Remaining useful life prediction based on a double convolutional neural network architecture. *IEEE Transactions on Industrial Electronics* 2019; 66(12): 9521-9530, <https://doi.org/10.1109/TIE.2019.2924605>.
22. Wang P, Zhang X, Zhang G. Remaining useful life prediction of lithium-ion batteries based on ResNet-Bi-LSTM-Attention model. *Energy Storage Science and Technology* 2023; 12(4): 1215, <https://doi.org/10.19799/j.cnki.2095-4239.2022.0652>.
23. He K, Su Z, Tian X, et al. RUL prediction of wind turbine gearbox bearings based on self-calibration temporal convolutional network. *IEEE Transactions on Instrumentation and Measurement* 2022; 71: 1-12, <https://doi.org/10.1109/TIM.2022.3143881>.
24. Cao Y, Ding Y, Jia M, et al. A novel temporal convolutional network with residual self-attention mechanism for remaining useful life prediction of rolling bearings. *Reliability Engineering & System Safety* 2021; 215: 107813, <https://doi.org/10.1016/j.res.2021.107813>.
25. Rahman M S, Colbourne B, Khan F. Conceptual development of an offshore resource centre in support of remote harsh environment operations. *Ocean Engineering* 2020; 203: 107236, <https://doi.org/10.1016/j.oceaneng.2020.107236>.
26. Shen J Y, Cui L R, Ma Y Z. Availability and optimal maintenance policy for systems degrading in dynamic environments. *European Journal of Operational Research* 2019; 276(1): 133-143, <https://doi.org/10.1016/j.ejor.2018.12.029>.
27. Huang C G, Huang H Z, Li Y F, et al. A novel deep convolutional neural network-bootstrap integrated method for RUL prediction of rolling bearing. *Journal of Manufacturing Systems* 2021; 61: 757-767, <https://doi.org/10.1016/j.jmsy.2021.03.012>.
28. Huang C G, Huang H Z, Li Y F, et al. Fault prognosis using deep convolutional neural network and bootstrap-based method. *2020 IEEE 18th International Conference on Industrial Informatics (INDIN) 2021: 742-747*, <https://doi.org/10.1109/INDIN45582.2020.9442148>.
29. Guo J Y, Wang J, Wang Z Y, et al. A CNN-BiLSTM-Bootstrap integrated method for remaining useful life prediction of rolling bearings. *Quality and Reliability Engineering International* 2023; 39(5): 1796-1807, <https://doi.org/10.1002/qre.3314>.
30. Chen C, Shi J T, Lu N Y, et al. Data-driven predictive maintenance strategy considering the uncertainty in remaining useful life prediction.

Neurocomputing 2022; 494: 79-88, <https://doi.org/10.1016/j.neucom.2022.04.055>.

31. Chen C, Tao G Y, Shi J T, et al. A lithium-ion battery degradation prediction model with uncertainty quantification for its predictive maintenance. *IEEE Transactions on Industrial Electronics* 2024; 71(4): 3650-3661, <https://doi.org/10.1109/TIE.2023.3274874>.
32. Zhang M, Wang D, Amaitik N, et al. A distributional perspective on remaining useful life prediction with deep learning and quantile regression. *IEEE Open Journal of Instrumentation and Measurement* 2022; 1: 1-13, <https://doi.org/10.1109/OJIM.2021.3082434>.
33. Erden C. Genetic algorithm-based hyperparameter optimization of deep learning models for PM2.5 time-series prediction. *International Journal of Environmental Science and Technology* 2023; 20(3): 2959-2982, <https://doi.org/10.1109/OJIM.2022.3205649>.
34. Taş G, Bal C, Uysal A. Performance comparison of lithium polymer battery SOC estimation using GWO-BiLSTM and cutting-edge deep learning methods. *Electrical Engineering* 2023; 105(5): 3383-3397, <https://doi.org/10.1007/s13369-022-07586-8>.
35. Che Z, Peng C, Yue C. Optimizing LSTM with multi-strategy improved WOA for robust prediction of high-speed machine tests data. *Chaos, Solitons & Fractals* 2024; 178: 114394, <https://doi.org/10.1016/j.chaos.2023.114394>.
36. Duankhan P, Sunat K, Chiewchanwattana S, et al. The differentiated creative search (DCS): Leveraging differentiated knowledge-acquisition and creative realism to address complex optimization problems. *Expert Systems with Applications* 2024; 123734, <https://doi.org/10.1016/j.eswa.2024.123734>.
37. Zhang Y R, Su C, Wu J J, et al. Trend-augmented and temporal-featured transformer network with multi-sensor signals for remaining useful life prediction. *Reliability Engineering & System Safety* 2024; 241: 109662, <https://doi.org/10.1016/j.res.2023.109662>.
38. Liu L, Song X, Zhou Z. Aircraft engine remaining useful life estimation via a double attention-based data-driven architecture. *Reliability Engineering & System Safety* 2022; 221: 108330, <https://doi.org/10.1016/j.res.2022.108330>.
39. Wang L, Cao H, Xu H, Liu H. A gated graph convolutional network with multisensor signals for remaining useful life prediction. *Knowledge-Based Systems* 2022; 252: 109340, <https://doi.org/10.1016/j.knsys.2022.109340>.
40. Chen C, Shi J T, Shen M Q, et al. A predictive maintenance strategy using deep learning quantile regression and kernel density estimation for failure prediction. *IEEE Transactions on Instrumentation and Measurement* 2023; 72: 1-12, <https://doi.org/10.1109/TIM.2023.3240208>.
41. Shoorkand H D, Noureifath M, Hajji A. A hybrid CNN-LSTM model for joint optimization of production and imperfect predictive maintenance planning. *Reliability Engineering & System Safety* 2024; 241: 109707, <https://doi.org/10.1016/j.res.2023.109707>.
42. de Pater I, Reijns A, Mitici M. Alarm-based predictive maintenance scheduling for aircraft engines with imperfect remaining useful life prognostics. *Reliability Engineering & System Safety* 2022; 221: 108341, <https://doi.org/10.1016/j.res.2022.108341>.
43. Kong Z, Cui Y, Xia Z, Lv H. Convolution and long short-term memory hybrid deep neural networks for remaining useful life prognostics. *Applied Sciences-Basel* 2019; 9(19): 4156, <https://doi.org/10.3390/app9194156>.
44. Zhang J, Jiang Y, Wu S, Li X, Luo H, Yin S. Prediction of remaining useful life based on bidirectional gated recurrent unit with temporal self-attention mechanism. *Reliability Engineering & System Safety* 2022; 221: 108297, <https://doi.org/10.1016/j.res.2021.108297>.
45. Liu L, Song X, Zhou Z. Aircraft engine remaining useful life estimation via a double attention-based data-driven architecture. *Reliability Engineering & System Safety* 2022; 221: 108330, <https://doi.org/10.1016/j.res.2022.108330>.
46. Zhang J, Li X, Tian J, Luo H, Yin S. An integrated multi-head dual sparse self-attention network for remaining useful life prediction. *Reliability Engineering & System Safety* 2023; 235: 109247, <https://doi.org/10.1016/j.res.2023.109096>.
47. Xiong J, Zhou J, Ma Y, Zhang F, Lin C. Adaptive deep learning-based remaining useful life prediction framework for systems with multiple failure patterns. *Reliability Engineering & System Safety* 2023; 235: 109244, <https://doi.org/10.1016/j.res.2023.109244>.
48. Vakharia V, Shah M, Nair P, et al. Estimation of lithium-ion battery discharge capacity by integrating optimized explainable-ai and stacked lstm model. *Batteries* 2023, 9, 125, <https://doi.org/10.3390/batteries9020125>.
49. Wang, S M, Ma C M, Xu Y X, et al. A hyperparameter optimization algorithm for the lstm temperature prediction model in data center. *Scientific Programming* 2022, 6519909, <https://doi.org/10.1155/2022/6519909>.
50. Dave V, Borade H, Agrawal H, et al. Deep learning-enhanced small-sample bearing fault analysis using Q-transform and HOG image features in a GRU-XAI framework. *Machines* 2024, 12, 373, <https://doi.org/10.3390/machines12060373>.
51. Song S S, Zhang S Q, Dong W, et al. Multi-source information fusion meta learning network with convolutional block attention module for bearing fault diagnosis under limited dataset. *Structural Health Monitoring-An International Journal* 2024, 23(2): 818-835, <https://doi.org/10.1177/14759217231176045>.

