Reliability Engineering and System Safety 237 (2023) 109357

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

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

Deep learning-based intelligent multilevel predictive maintenance framework considering comprehensive cost

Kai-Li Zhou [a,1], De-Jun Cheng [a,b,1,*], Han-Bing Zhang [a], Zhong-tai Hu [a], Chun-Yan Zhang [a]

[a] School of Mechanical Engineering, Jiangsu University of Science and Technology, Zhenjiang, 212003, China

[b] School of Power and Energy, Northwestern Polytechnical University, Xi'an, 710072, China

ARTICLE INFO

Keywords:
Intelligent multilevel predictive maintenance
Series-parallel multi-state system
Deep learning
Remaining useful life
Comprehensive cost

ABSTRACT

Due to the increase in the series-parallel multi-state system (MSS) complexity caused by the nonlinear change of parameters, the traditional model-based maintenance methods are becoming less effective and obsolete. This study proposes a novel deep learning-based intelligent multilevel predictive maintenance (MPM) framework for series-parallel MSS considering comprehensive cost. A new adaptive convolution-transformer (C-Transformer) was constructed to predict component remaining useful life (RUL) through extracting features adaptively. Based on this, the component failure probability was obtained through convolutional neural network (CNN). Then, to directly reflect the operating conditions of MSSs, multilevel maintenance was customized with multilevel failure through the trial-and-error learning method. During the intermission breaks, an intelligent dynamic decision-making optimization model was proposed by introducing multilevel maintenance to improve the system's state in a future mission, which was solved by a new artificial bee colony algorithm (called MDU-ABC-K) to minimize the comprehensive cost under economic dependence and critical component constraints, thus simultaneously balancing maintenance time and cost. The proposed approach was compared with other models through turbofan engine data set by NASA. The comparison results indicate that the proposed intelligent MPM framework can offer a more reasonable and superior maintenance strategy.

1. Introduction

Currently, the scale, performance, and automation of the series-parallel multi-state systems (MSSs) are rising sharply, increasing the difficulty of maintenance decisions when the complex system is shut down due to component aging or failure [1], which leads to the traditional model-based maintenance methods are becoming less effective and obsolete. Therefore, most studies examined predictive selective maintenance (PdSM) due to the revolution of Industry 4.0.

Generally, the PdSM framework consists of two key parts: RUL prediction and decision-making. There are two existing approaches (model-driven and data-driven) for accomplishing the remaining useful life (RUL) prediction of machinery systems. Model-driven approaches mostly use a mathematical model to calculate the component's lifetime, which presents a functional relationship [2,3]. However, for complex series-parallel MSSs, the mathematical model is less accurate and cannot be directly applied in engineering scenarios due to operation variables. Data-driven approaches have emerged in recent years to address the problems that have arisen with model-driven approaches, which are used to learn multi-scale features automatically from massive data for RUL prediction. Data-driven PdSM frameworks are mostly developed based on machine learning (ML) and deep learning (DL), requiring sufficient data to predict the RUL [4] instead of using the degradation mechanism and prior knowledge, greatly reducing expert domain knowledge dependence [5]. However, the ML approaches are less effective in dealing with massive data because the algorithm is relatively simple and less sensitive to missing data [6]. Compared to ML, DL is a complex machine learning algorithm, which has been widely studied in the studies of RUL prediction and performs far beyond previous related techniques in ML prognostics [7]. In which, transformer is a specific network based on multi-head attention mechanism that is different from other DL architectures, which is widely used in fault detection, sequence prediction and machine health monitoring [8,9].

The second part of PdSM focuses on maintenance decision-making, in reality, the system is usually maintained to improve the performance and prolong the useful life. The maintenance can be classified

* Corresponding author.
E-mail address: chengdejun0126@just.edu.cn (D.-J. Cheng).

[1] Both authors contribute equally.

https://doi.org/10.1016/j.ress.2023.109357
Received 9 August 2022; Received in revised form 6 March 2023; Accepted 3 May 2023
Available online 4 May 2023
0951-8320/

into three categories based on the degree of maintenance, i.e., minor maintenance, imperfect maintenance, as well as perfect maintenance. In actual, maintenance activities are usually considered as imperfect maintenance, and the system can only be restored to a certain state between as-good-as new state and as-bad-as old state [10,11]. In addition, considering the optimization objects, the decision-making optimization models can be divided into three categories [4,12-14]. The first aims at maximizing system reliability with time cost and other maintenance resources as constraints. The second aims to minimize overall maintenance costs with time and system reliability constraints. Finally, the third aim is to minimize overall maintenance time with system reliability as a constraint. From the decision-makers' perspective, the objectives are determined based on the feasibility of maintenance resources and industrial demand. Moreover, for complex series-parallel MSSs, the dynamic decision-making optimization model should be customized under mutual restriction between maintenance cost and time to allocate maintenance resources more rationally, maximize the system life, and avoid the MSSs failure.

The remainder of this paper is constructed as follows: Section 2 reviews some significant existing works to highlight the paper's contribution. Section 3 illustrates the description of the series-parallel MSS. Section 4 proposes an intelligent MPM framework consists of data-driven prognostics and dynamic decision-making optimization. In prognostics, we develop a new adaptive C-Transformer network and CNN to predict the RUL and component failure probability respectively. For decision-making, the optimal dynamic strategies are output with the combination of dynamic optimization model and a new MDU-ABC-K algorithm under the comprehensive cost. In Section 5, experiment results and four illustrative cases are presented to verify the proposed approach. A conclusion is drawn in Section 6.

2. Literature review

2.1. Related works

The concept of SM was initially proposed by Rice et al. [15]. A theoretical programming model was developed to address the SM problem of systems consisting of identical components. Only a portion of maintenance actions that meet existing maintenance demand can be implemented in the system to ensure the completion probability of the future mission among all the feasible maintenance actions. Thereafter, Maillart et al. [3] and Schneider et al. [2] extended it to the maintenance of complex systems, assuming that each component's RUL is exponentially distributed. In many SM studies, parametric methods are applied to randomly model the system degradation process to represent components' RUL [16]. Besides, Markov models are also widely used in the RUL prediction of components to improve the prediction accuracy [5]. Ingeborg de Pater and Mihaela Mitici [17] develop a model-based RUL prediction approach, which is updated over time, as more sensor data become available. The above model-driven approaches usually need to establish the mathematical RUL components model. In addition, model-driven approaches include the reliability model to model the probability of successful system maintenance in future missions [18–20]. Generally, the components and systems are assumed to be binary, perfect functioning and completely failed [21]. However, the binary system assumptions cannot be applied in complex systems due to operation variables. Thus, incasing attention has been paid to MSS reliability models. Extended decision diagram-based methods [22,23], the universal generating functions [24,25], and the recursive algorithm [26] are designed to promote the system reliability assessment. In addition, Tao Jiang and Liu Y [5,18] introduced the Gaussian quadrature, Riemann sum, and the Bayesian networks in completion probability assessment. However, the above model-driven approaches need to use measured data to update mode parameters and assume that the components' lifetime follows a parametric process model, which cannot be effectively applied in industry.

Unlike model-driven approaches, the data-driven approaches are based on extracting features through real-time monitoring conditions. Thus, the data-driven approaches-based PdSM frameworks have been developed rapidly in recent years, focusing on RUL prediction and decision-making. The ML approaches are less effective in processing massive data, because the algorithm is relatively simple and less sensitive to missing data [6]. In ML, feature selection needs expert knowledge, difficult data processing and feature extraction in the early stage. On the contrary, in DL, an approach in which the computer automatically learns multi-scale features to integrate feature learning into the model building process, reducing the defects caused by artificial feature design [27,28]. The long short-term memory (LSTM) network [29,30] is widely used in the RUL prediction. Based on the LSTM network, Khanh T.P. Nguyen and Kamal Medjaher [31] and Hadis Hesabi et al. [4] developed a PdSM framework of the series system, which can predict the RUL and the failure probability of each component simultaneously. Juseong Lee and Mihaela Mitici [32] estimate the distribution of the RUL using CNN with Monte Carlo dropout, and applied into the maintenance planning problem. Chong Chen et al. [33] propose a merged-long-short term memory (M-LSTM) network for health indicator and RUL estimation. A fault mode-assisted gated recurrent unit (FGRU) life prediction method is proposed to guide the predictive maintenance initiation time of all machines [34].

The above mentioned studies more focus on the prognostics and the maintenance decisions of MSSs, separately. In decision-making, most studies build an optimization model equivalent to the max-min optimization model to minimize maintenance cost/time or maximize the probability of completion [18] of future missions during a limited break between the adjacent missions. For example, Nourelfath and Chatelet [35] investigate the preventive maintenance planning problem with the objective of minimizing the total production and maintenance cost. Tao Jiang [5] proposes a selective maintenance model for MSSs with the objective of maximizing the probability of completion in future missions subject to maintenance cost constraints. Moreover, many considerations are incorporated into the decision-making for better execution of maintenance actions, such as economic dependence [13], structural dependence [36], variable loading conditions [37] and critical components [25]. To improve the prediction accuracy of optimization models, diverse heuristic algorithms are widely used to output the optimal maintenance strategy. K. Chaabane et al. [14] develop a heuristic method based on the genetic algorithm, which is to minimize the total maintenance and labor costs for a maintenance plan that guarantees a given reliability threshold. Pravin P [36] presents a selective maintenance decision model with the objective of minimizing the total cost of maintenance decision. Meanwhile, the simulated annealing algorithm is conducted to solve the proposed model.

Although above mentioned methods have been successfully developed, three are still three crucial problems: (1) Due to the system maintenance strategy depends on failure probability and the current state of each component within the MSSs, it is unreasonable to perform maintenance without considering the multistate of systems. (2) The MSS complexity increases rapidly due to the nonlinear change of parameters, and the difficulty of decision-making also increases. However, most works consider the single maintenance capacity instead of multilevel maintenance. (3) For selecting a superior instantaneous maintenance strategy, the dynamic optimization model should consider multiple options and evaluate the comprehensive cost to balance the maintenance resource.

2.2. Paper contribution

In this study, a novel intelligent MPM framework for the RUL prediction and dynamic maintenance decision in multi-state MSSs under multilevel failure conditions is proposed, as shown in Fig. 1. In prognostics, we develop a new adaptive C-Transformer network and CNN to predict the RUL and component failure probability respectively. For

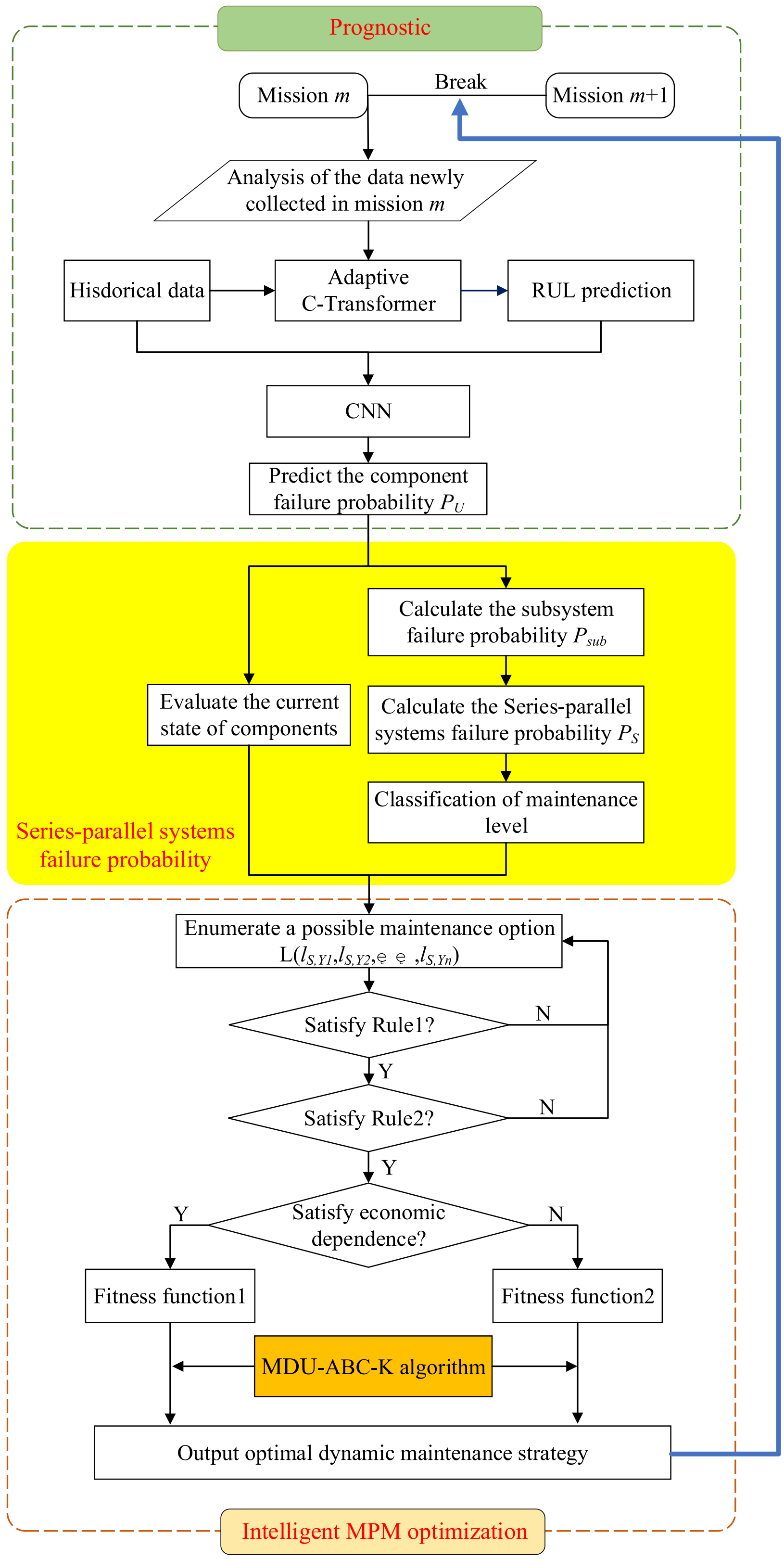

Fig. 1. Framework of proposed intelligent MPM.

decision-making, the optimal dynamic strategies are output with the combination of dynamic optimization model and a new MDU-ABC-K algorithm under the comprehensive cost. The proposed MPM framework not only can consider the real-work condition but also provide accurate information to improve multi-state MSSs reliability. The main contributions of this study are as follows:

(1) A new adaptive C-Transformer network is proposed, which can effectively predict component RUL through extracting features adaptively varied to dataset. Comparison results indicate that the proposed model presents better performance than other models.
(2) A novel deep learning-based MPM framework for series-parallel MSSs is proposed. The current state of multi-state components and multilevel maintenance of the series-parallel system are considered to output the maximum step size spanned by the component state before and after maintenance by introducing multilevel maintenance, thus reducing the computational complexity and expense of the decision-making strategy.
(3) The intelligent decision-making optimization model for the series-parallel MSS is customized under mutual restrictions between maintenance cost and time to allocate maintenance resources more rationally and avoid wasting time. Furthermore, the optimum dynamic maintenance strategies are output through the new MDU-ABC-K algorithm under economic dependence and critical components constraints with minimum comprehensive cost as the optimization objective by assigning appropriate weights to maintenance time and cost.

3. Problem description

3.1. Series-parallel mss

In this work, the series-parallel system includes M independent subsystems, as illustrated in Fig. 2, where subsystem i ($i = 1,2,\ldots,M$) consists of N_i identical components in parallel. In engineering scenarios, each component might be in multi states (from a completely failed state to a perfect functioning state) due to component degradation. Here, a set of S is used to describe the $K+1$ possible states of each component, $S=\{0,1,\ldots,K\}$, where state 0 represents completely fail, state K represents perfectly function, and others represent the intermediate states.

For each component, X represents the current state vector of the system and Y represents the system state when the maintenance operation is completed. Eq. (1) and Eq. (2) can express both, respectively, where $0 \le X_{ij} \le K$ and $X_{ij} \le Y_{ij} \le K$, because maintenance activities are unlikely to result in the performance degradation of the component and system.

$$X = \{X_{ij}\}, (i = 1, 2, \ldots, M;\ j = 1, 2, \ldots, N_i) \tag{1}$$

$$Y = \{Y_{ij}\},\ (i = 1, 2, \ldots, M;\ j = 1, 2, \ldots, N_i) \tag{2}$$

3.2. Comprehensive cost of the system

The comprehensive cost represents the maintenance cost and time of improving the state of any component in the same subsystem from X_{ij} to Y_{ij}, which can be presented by $c_i(X_{ij}, Y_{ij})$ and $t_i(X_{ij}, Y_{ij})$. If the component remains in the same condition after the maintenance activities, then $X_{ij} = Y_{ij}$, therefore $c_i(X_{ij}, Y_{ij}) = 0$ and $t_i(X_{ij}, Y_{ij}) = 0$. On the contrary, if $Y_{ij} = K$, the faulty component has been replaced with the new one. Suppose the maintenance cost of components in the same subsystem can be presented by the same matrix, and the maintenance time of components is the same too, because the components in the same subsystem are identical. Therefore, the possible $c_i(X_{ij}, Y_{ij})$ and $t_i(X_{ij}, Y_{ij})$ of each component form cost C_i and time T_i matrix of $(K+1) \times (K+1)$ respectively, as written in Eqs. (3) and (4)

$$C_i = \begin{bmatrix} 0 & c_i & \cdots & c_i(0, K) \\ 0 & 0 & \cdots & c_i(1, K) \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & c_i(K-1, K) \\ 0 & 0 & \cdots & 0 \end{bmatrix} \tag{3}$$

$$T_i = \begin{bmatrix} 0 & t_i & \cdots & t_i(0, K) \\ 0 & 0 & \cdots & t_i(1, K) \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & t_i(K-1, K) \\ 0 & 0 & \cdots & 0 \end{bmatrix} \tag{4}$$

To calculate the maintenance cost and time of the entire system, it is usually assumed that the maintenance of each component is independent. Thus, the total maintenance cost and time during the maintenance period is the sum of all components, which can be computed using Eqs. (5) and (6).

$$C_M(\mathrm{X}) = \sum_{i=1}^{M} \sum_{j=1}^{N_i} c_i\ (X_{ij},\ Y_{ij}) \tag{5}$$

$$T_M(\mathrm{X}) = \sum_{i=1}^{M} \sum_{j=1}^{N_i} t_i\ (X_{ij},\ Y_{ij}) \tag{6}$$

This study introduces the comprehensive cost to allocate maintenance resources more rationally and avoid wasting time. Thus, we assign the appropriate weights k_1 and k_2 to maintenance cost and time, respectively, as expressed in Eq. (7).

$$C_T(\mathrm{X}) = k_1 C_M(\mathrm{X}) + k_2 T_M(\mathrm{X}) \tag{7}$$

In which, $k_1 + k_2 = 1$, and the actual resource distribution determine the value of k_1 and k_2. If the manager pays more attention to saving maintenance cost, the value of k_1 will be greater than k_2; On the other hand, if the manager prefers to save more maintenance time, k_2 is greater than k_1.

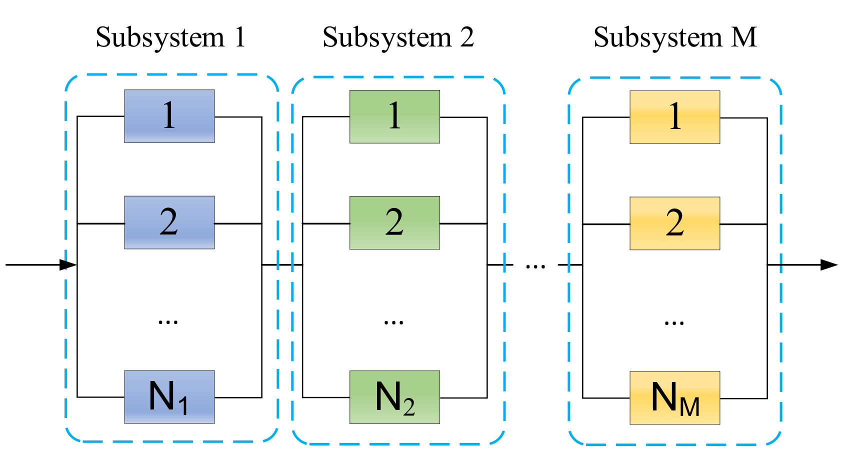

Fig. 2. Series-parallel system.

3.3. The economic dependence on maintenance

In most series-parallel MSS, the maintenance strategy will select multiple components for maintenance, especially for the components in the same condition in each subsystem, saving time and costs. In subsystem i, the components in the same condition usually need to adopt the same maintenance activities to improve performance. Furthermore, during maintenance, time and cost can be saved by reusing the technology and equipment, and the repair dependence of the same components can be expressed by the saving factor $f_c^i(\mathbf{a},\mathbf{b})$. The preparation for each component is fixed before repair, which requires similar initial setting up, equipment, and labor. A fixed amount of "set-up cost" and "set-up time" can be expressed by Δc_s and Δt_s, respectively. If N_r identical components are repaired, the total of saved maintenance cost and time due to the share of setting up will be $(N_r - 1) \times \Delta c_s$ and $(N_r - 1) \times \Delta t_s$. Thus, the maintenance cost of the first component is equal to the cost of independent maintenance of the component. For the next component, maintenance costs $c_i^{'}(a, b)$ can be calculated by using Eq. (8).

$$c_i^{'}(a, \mathbf{b}) = \begin{cases} f_c^i(\mathbf{a}, \mathbf{b}) c_i(\mathbf{a}, \mathbf{b}) - \Delta c_s, & \textbf{If there is identical repair} \\ c_i(\mathbf{a}, \mathbf{b}) - \Delta c_s, & \textbf{Otherwise} \end{cases} \tag{8}$$

Similarly, the maintenance time of the first component is equal to the time of independent component maintenance. For the next component, maintenance time $t_i^{'}(a, b)$ can be obtained by using Eq. (9).

$$t_i^{'}(a, \mathbf{b}) = \begin{cases} f_c^i(\mathbf{a}, \mathbf{b}) t_i(\mathbf{a}, \mathbf{b}) - \Delta t_s, & \textbf{If there is identical repair} \\ t_i(\mathbf{a}, \mathbf{b}) - \Delta t_s, & \textbf{Otherwise} \end{cases} \tag{9}$$

Based on the above analysis, considering the economic dependence on maintenance, the total maintenance cost and maintenance time can be calculated by Eqs. (10) and (11), respectively.

$$C'_M(\mathrm{X}) = \sum_{i=1}^{M} \sum_{j=1}^{N_i} c'_i\left(X_{ij}, Y_{ij}\right) \tag{10}$$

$$T'_M(\mathrm{X}) = \sum_{i=1}^{M} \sum_{j=1}^{N_i} t'_i\left(X_{ij}, Y_{ij}\right) \tag{11}$$

3.4. Critical component

From the perspective of the system structure, components must be maintained in the series-parallel MSS structure to improve the system performance [20]. Critical components are defined as components that contribute significantly to improving the system performance. Thus, the critical components constraint means that the component must be maintained to the preferred value during the period of maintenance. The system cannot complete the next mission if critical components are unrepaired or not reached the preferred state.

4. Deep learning-based intelligent mpm

4.1. Deep learning predictive model for component failure probability

The above description of the PdSM problems demonstrates that the composition of series-parallel MSS is complex and contains various information from different sensors, which has an important impact on the prediction results. In the actual industrial scenarios, there are a lot of miscellaneous information in the data collected by sensors due to the variety of equipment. To make better use of the data-driven approach to estimate the RUL of the device, the existing studies [38,39] often select some fixed feature for network training. However, due to the features are always varied to the dataset, traditional fixed feature selection method cannot extract effective features adaptively in response to different training data, which leads to the deterioration in RUL prognostics issue.

To address this issue, a new adaptive C-Transformer is proposed to extract feature automatically varied to dataset. Based on this, a deep learning architecture is constructed, which can predict the RUL and component failure probability simultaneously and clarify their inherent correlation, as shown in Fig. 3. First, the original data are fed into self-attention C-Transformer network, which can effectively predict component RUL through extracting features adaptively. Then, through relabeling the original data and the predicted RUL to form new training data, the CNN network is applied to predict the component failure probability accurately.

4.1.1. The architecture of the adaptive C-Transformer

The transformer is a model for sequence-to-sequence tasks, which entirely lies in self-attention mechanisms and can reduce training time and performance decline due to long-term dependence [39]. In this work, a new adaptive C-Transformer is proposed for solving sequence-to-sequence tasks, which entirely lies in self-attention mechanisms, and reduce training time and performance decline due to long-term dependence. The proposed adaptive C-Transformer includes feature extraction layer, input layer, an encoder block, a decoder block, and an output layer, detailed as follows.

A Feature extraction layer

The feature extraction layer consists of the convolutional layer, dense layer and the ReLu activation function. In order to lightweight the feature extraction layer, we abandon the pooling operation and only use the convolution operation. Through using the convolutional kernel, the convolutional layer extracts the features from the input data. During data training process, the convolution kernel will periodically scan the input features. Then, ReLU activation function is applied to better fit the complex nonlinear feature data. Finally, the features are input into dense layer for linear change to effectively obtain the feature extraction results. After unifying the dimensions, the features are fed into the input layer of transformer for RUL network training.

A Input and output layer

The input of the transformer is a data sequence. The input layer includes the input embedding layer, output embedding layer, and position encoding layer. In particular, the processing steps of the decoder layer inputs are the same as the encoder layer outputs. The embedding layer maps the input data to vectors of dimension d_{model}. It is worth noting that after input embedding, it is necessary to add position encoding to the word vector of each word. We must inject some information about the absolute position of tokens so that the network can take full advantage of the sequence order. In this study, sine and cosine functions of different frequencies were selected for position encoding:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right) \tag{12}$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right) \tag{13}$$

Where pos is the position, and i is the sensor dimension.

Compared to the input layer, the composition of the output layer is relatively simple, consisting of linear transformations and sigmoid activation. The RUL prediction can be obtained through linear transformation and sigmoid activation.

A Encoder block

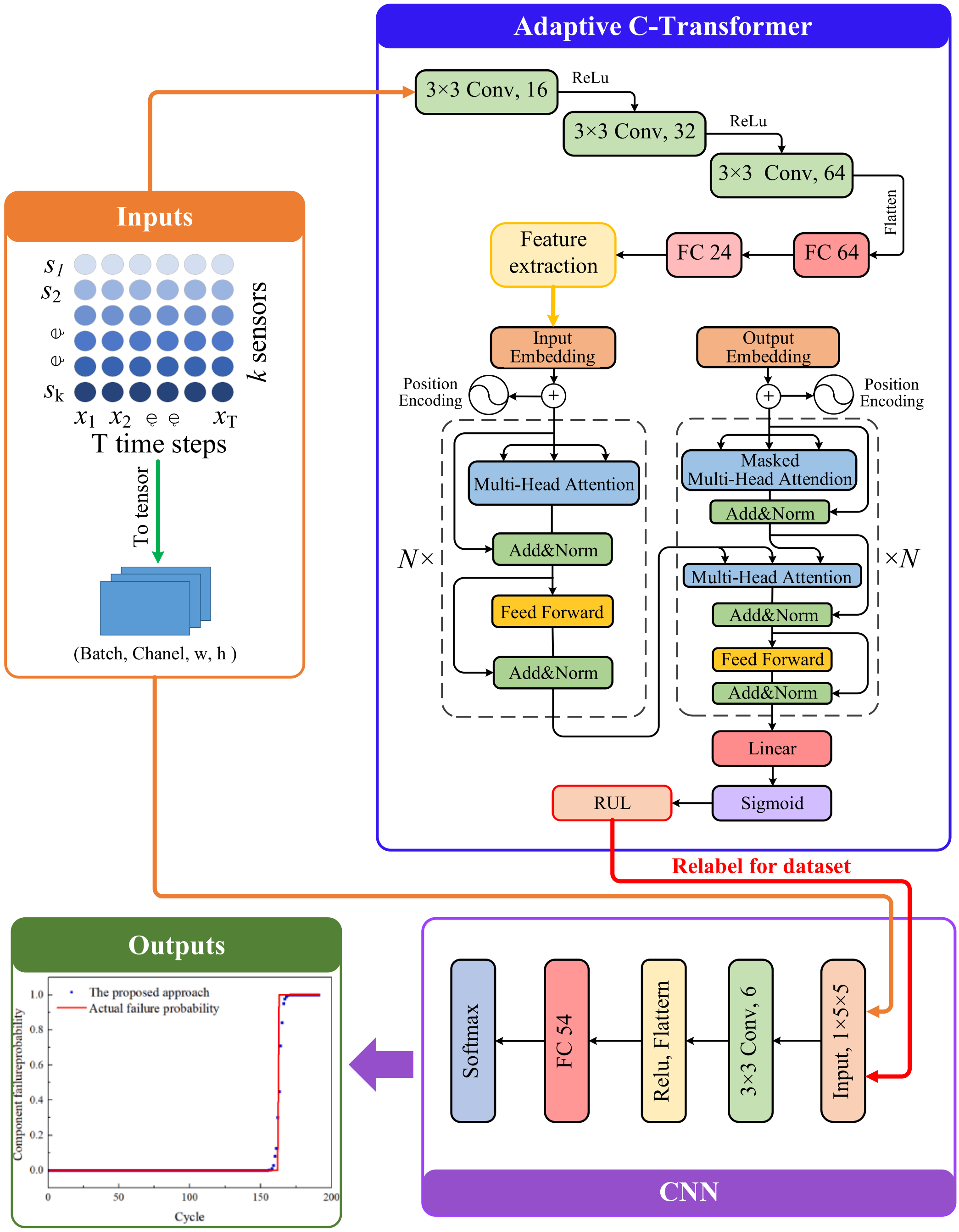

Fig. 3. The architecture of proposed deep learning network for component failure probability.

The encoder block is formed by stacking N identical layers. Each layer has two sub-layers, multi-head attention (MHA) mechanism, and a feed-forward network (FFN). Residual connection and layer normalization are used after each sublayer. The residual connection and layer normalization purposes alleviate the difficulty of training deep neural networks and making the model converge faster. The encoder layer automatically learns to pay attention to the higher-weight features through the multi-head self-attention mechanism instead of the training process with human experience. The MHA and FFN of the encoder are described in the following.

1) MHA

The most important attention mechanism is the attention function, represented in the transformer as mapping a query vector and a set of key-value vectors to the output. "Scaled Dot-Product Attention" (SDPA) is the particular attention mechanism used in transformer. Queries, keys, and values form the input, and the dimensions of keys and values are d_k and d_v, respectively. In practice, the results of a set of queries computed concurrently by the attention function are packed into matrix Q. Similarly, keys and values are packed together into matrices K and V. Based on the above analysis, Eq. (14) can express the outputs matrix.

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right) \tag{14}$$

Several SDPA layers running in parallel formed MHA. The multi-head self-attention mechanism allows models to focus on the information at different positions, thus improving the prediction accuracy, which can be expressed as:

$$MultiHead(Q, K, V) = Concat(head_1, \dots, head_h)W^O \tag{15}$$

$$head_i = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{16}$$

where h is the number of heads, W_i^Q, W_i^K, and W_i^V are parameter matrices, and $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{model} \times d_v}$, and $W^O \in \mathbb{R}^{hd_v \times d_{model}}$.

1) FFN

The FFN is a two-layer neural network consisting of two linear transformations with a ReLU activation in the middle. The formula of the fully connected layer is as follows:

$$FFN(x) = \max(0, xW_1 + b_1)W_2 + b_2 \tag{17}$$

The purpose of the two layers is to map the input into a higher-dimensional space, filter it through the nonlinear function ReLU, and finally change it back to the original dimension after screening.

A Decoder block

The decoder block is formed by stacking N identical layers. Each layer consists of three sub-layers, an MHA mechanism, a masked MHA mechanism, and FFN. The decoder layer has a different masked MHA mechanism from the encoder layer. However, the MHA is slightly different from the encoder layer except for the masked MHA. The details of MHA and masked MHA of the decoder are described in the following section.

1) Masked MHA

Masked MHA is composed of a padding mask and a sequence mask. Since each batch of input sequence length is not the same, it is necessary to use a padding mask to align the input sequence. The purpose of the sequence mask is to mask out (setting to-∞) all values. Thus, the attention mechanism only depends on the previous output data before time t when we predict the RUL of t.

1) MHA

Unlike the MHA in the encoder layer, the decoder layer is only based on attention, not self-attention. Its input query comes from the masked MHA output, and keys and values come from the output of the last layer in the encoder. The MHA purpose is to obtain the information of the next moment through the current input information, that is, the output information, to predict the output by expressing the relationship between the current input and the feature vector extracted by the encoder.

4.1.2. The architecture of cnn

CNN is commonly used in speech analysis and image recognition and has also been used for classification tasks in recent years. The architecture of CNN is shown at the bottom of Fig. 3. The input layer, convolutional layer, flatten layer, fully connected layer, and softmax activation formed the network. In this work, CNN provides 6 filter kernels in convolutional to learn features well. In which the convolutional kernel is a two-dimensional tensor. The flatten layer is used to turn the data into one dimension to be used as the input of the fully connected layer. The fully connected layer integrates and computes the previously extracted features as the last stage of the whole network. Softmax is an activation function for classification problems, which can be expressed as:

$$\mathbf{Softmax}(z_i) = \frac{\mathbf{e}^{z_i}}{\sum_{c=1}^{C} e^{z_c}} \tag{18}$$

where z_i and c represents the output value of the i th node and the number classification categories, respectively. The output value of multi-class can be transformed into a probability distribution in the range of [0, 1] by the softmax function.

In this work, the output result from CNN is the component failure probability of the series-parallel system, summarized into the following two classes:

Class 0: When the RUL is larger than or equal to the previous preset time window w.
Class 1: When the RUL is less than the previous preset time window w.

It is assumed that the time window w equals to the length of the next mission. Therefore, if the RUL of a component is less than the time window, it belongs to Class 1, and the system is considered as completely failed in the next mission. In contrast, when the RUL of a component is larger than the time window, it belongs to Class 0, and the system is considered as functioning in the next mission.

4.2. Series-parallel system failure probability model

4.2.1. Evaluation of the current state of components

Existing data-driven PdSM frameworks [4,31] focus more on simple series system cases, which are rarely test in complex MSSs. In addition, the predicted component failure by the above PdSMs cannot reflect the current state of a component of the series-parallel MSS, thus increasing the difficulty of maintenance activities. Therefore, to closely integrate the failure probability with the series-parallel MSS, multilevel failure was introduced to determine the component's current state. In this study, the evaluation criterion of the current state of components in response to multilevel failure probability was defined as listed in Table 1.

4.2.2. Classification of maintenance level

In general, we assumed that the series-parallel MSS is formed by M independent subsystems connected in series, and each subsystem i is composed of N_i identical components in parallel. Based on the probabilistic union rule, the failure probability of a multi-component subsystem can be computed by Eq. (19) and (20).

$$\mathrm{P}_{sub_i} = 1 - \mathrm{P}\{(RUL_1 \geq w) \cap (RUL_2 \geq w) \cdots (RUL_{N_i} \geq w)\} \tag{19}$$

$$= 1 - \{P(RUL_1 \geq w) \cdot P(RUL_2 \geq w) \cdots P(RUL_{N_i} \geq w)\} \tag{20}$$

In which, RUL_{N_i} and P_{sub_i} represent the RUL of the component in subsystem i and the failure probability of subsystem i, respectively.

Since a series of subsystems are combined to form a system, the minimum subsystem performance determines the performance rate of the system, which can be written as Eq. (21),

$$P_S = min(P_{sub_1}, P_{sub_2}, \dots, P_{sub_i}) \tag{21}$$

where P_S represents the failure probability of the entire system.

To reduce the computational complexity and maintenance cost during decision-making, the maintenance level considering imperfect maintenance is proposed based on the trial-and-error learning method with the calculated failure probability of the entire system. In this work, five-level maintenance is adopted in response to the system multilevel failure to formulate the system maintenance strategy. Maintenance level indicates the maximum step size of the state span before and after maintenance. In which, the maximum step size is used to determine the maximum state that can be repaired without wasting redundant maintenance resources under the system current state. Such as, Level 4 means that the maximum step size of the state span before and after maintenance is 5, and Level 3 means that the maximum step size is 4. Then Levels 1 and 2 analogize in turn, and Level 0 means no maintenance is

Table 1
Evaluation criterion of current state of component.

State	0	1	2	3	4	5	6
P_U	[0.85,1.0]	[0.7,0.85)	[0.55,0.7)	[0.4,0.55)	[0.25,0.4)	[0.1,0.25)	[0,0.1)

done. The trial-and-error testing process is shown in Fig. 4. The state of each component after the repair is taken as a criterion that the highest state cannot be higher than the optimal state plus 1 at most, and the lowest state cannot be lower than the intermediate state. As the illustrated MSSs in Section 5.4, a set of component failure probability values are randomly selected from the training data (FD001 dataset), which corresponds to component states. Then, a maintenance level value is assumed based on the calculated system failure probability by Eqs. (19)-(21), which is reversed corrected through the trial-and-error process. Due to the same system failure probability will have multiple sets of different component failure probabilities. Thus, the system failure probability boundaries of maintenance levels can be extracted after repeating the above process by many experiments. The evaluation criteria of maintenance level in response to various failure probabilities of the entire system are listed in Table 2.

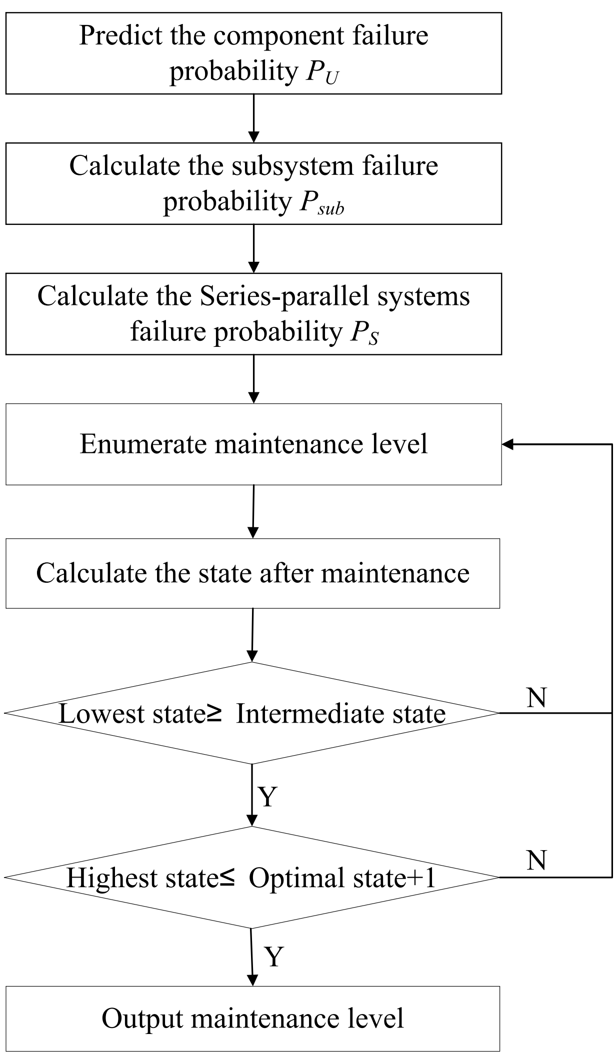

Fig. 4. The trial-and-error testing process of maintenance level.

Table 2
Evaluation criterion of maintenance level.

Maintenance level	Level 0	Level 1	Level 2	Level 3	Level 4
P_s	[0,0.4)	[0.4,0.6)	[0.6,0.75)	[0.75,0.9)	[0.9–1]

4.3. Intelligent dynamic decision-making optimization

4.3.1. Dynamic optimization model

The series-parallel MSS systems are assumed to have the following characteristics: (1) All the components in a subsystem are identical. (2) Considering the imperfect maintenance activities, the system can only be restored to a certain state between as-good-as new state and as-bad-as old state. (3) The resource requirements for a single maintenance activity are identified and known, and the time and cost available for maintenance before the next mission are limited. (4) The interval break between two successive missions is assumed to be constant, and noted as $\triangle \boldsymbol{T}$. The maintenance actions in this paper are only performed at the break time.

To allocate maintenance resources more rationally and avoid wasting time, the intelligent decision-making optimization model for series-parallel MSS is customized under mutual restriction between maintenance cost and time, which can be written as follow:

$$\textbf{Minimize } C_T(\mathbf{X}) = k_1 C_M(\mathbf{X}) + k_2 T_M(\mathbf{X}) \tag{22}$$

$$\textbf{subject to}: \ X_{ij} \le Y_{ij} \le K \tag{23}$$

$$X_{ij} \ is \ an \ integer, \ i = 1,2,\dots,M; j = 1,2,\dots,N_i \tag{24}$$

The objective function minimizes the total comprehensive cost under the constraints of economic dependence and critical components. This article focuses on the maintenance of series-parallel MSS through the interaction of multilevel failure and multilevel maintenance. The following section uses the new MDU-ABC-K algorithm to automatically output the optimal dynamic maintenance strategy.

4.3.2. Intelligent decision-making solution approach

ABC algorithm is a bionic computing method which simulates the organization behavior of bees when they collect honey and search for good nectar source. However, it has the disadvantages of fast convergence in the early stage, easy to fall into local optimum, loss of diversity in the late stage, and slow convergence. To tackle these problems, an improved ABC algorithm is proposed, in which multiple dimensions updating and k-neighborhood radius are applied to enhance solution search and improve the quality of the solution respectively, thus to jump out of the local best solution. In this work, the proposed algorithm is named as MDU-ABC-K.

In the proposed MDU-ABC-K algorithm, we assumed that there are N nectar sources ($X_1, \ X_2, \ \dots, \ X_N$), which has D components per nectar source. That is, the solution space of the problem to be optimized contains N feasible solutions ($X_i = x_{i1}, \ x_{i2}, \ \dots, x_{iD}$), and each feasible solution is a D-dimensional vector. The proposed MDU-ABC-K algorithm contains five phases as follows:

A Colony initialization phase

For any solution $x_{id}(d = 1, \ 2, \ 3, \ \dots, \ D)$, each component is initialized, which can be expressed as:

$$x_{id} = x_{idmin} + rand(0,\ 1)\cdot(x_{idmax} - x_{idmin}) \tag{25}$$

where x_{idmax} and x_{idmin} represent the upper and lower limits of the D-dimensional components of the feasible solution space respectively, $rand(0,\ 1)$ is a random number between [0, 1].

A Employed bee search phase

In the traditional ABC algorithm, the bees adopt the strategy of updating the solution by dimension, which can be written as follows:

$$v_{id} = x_{id} + rand(-1, 1)\cdot\left(x_{id} - x_{jd}\right) \tag{26}$$

In which, $j \in \{1,\ 2, \ldots,\ N\}, j \neq i$ represents the selected nectar source is different from x_i among N nectar sources. Then, the fitness value of the new solution is calculated, and the greedy strategy is applied to select between x_{id} and v_{id}. Finally, employed bees will record nectar source information and fitness values. However, as the function dimension increases continuously, the convergence time will increase simultaneously, the accuracy of the final solution will be affected. The reason is that in the search process, if there are better values in some dimensions but no further mining is obtained, the solution will reach the search limit of the number of repeated mining times of honey source and be discarded. Then the scout bees will search again randomly, which will cause the algorithm to miss many opportunities to reach the global optimum.

In this paper, to address above issue, the one dimension in Eq. (26) is replaced by m different dimensions at the same time. Then the new updated formula can be expressed as:

$$\begin{cases} v_{i1} = x_{i1} + rand(-1,\ 1)\cdot\left(x_{i1} - x_{j1}\right) \\ v_{i2} = x_{i2} + rand(-1,\ 1)\cdot\left(x_{i2} - x_{j2}\right) \\ \quad\quad\quad \cdots\cdots \\ v_{iD} = x_{iD} + rand(-1,\ 1)\cdot\left(x_{iD} - x_{jD}\right) \end{cases} \tag{27}$$

Where m is determined by the number of dimensions of solutions.

A Onlooker bee search phase

After all the employed bees have completed the search, the solution information and fitness will be shared with the onlooker bees. The observer bees determine the probability of each onlooker bee being followed through choosing the probability P_i, which can be expressed as follows:

$$P_i = \frac{fit_i}{\sum_{k=1}^{N} fit_k} \tag{28}$$

where fit_k is the fitness value of the k-th solution. The onlooker bees use a roulette strategy to select the employed bees to follow. If the selection probability value of employed bees corresponding to the nectar source is larger, it will be followed by more onlooker bees, that is, the nectar source with greater fitness value has a wider search range of the neighborhood of the corresponding solution.

A Scout bee search phase

At this stage, to replace the abandoned nectar source, the standard nectar source is formulated as below:

$$x_{id} = low_d + rand(0,\ 1)\cdot(up_d - low_d) \tag{29}$$

where $d \in \{1,\ 2, \ldots,\ D\}$,and $[low_d,\ up_d]$ is the boundary constraint.

However, when a certain nectar source is not updated after being searched for repeated mining times, it is assumed to be exhausted and the corresponding solution will fall into the local optimum. To solve the above problem of falling into the local optimum, a new nectar source solution method is proposed with the k-neighborhood radius to update the abandoned solution.

Based on the concept of k-neighborhood, a novel solution selection mechanism is proposed for the onlooker bee search phase. For each solution X_i in the swarm, the best solution in the k-neighborhood of X_i is chosen as X_{ib}. Then, we use the corresponding search strategy searches around X_{ib}. The k-neighborhood of X_i has 2k+1 solutions, and X_{ib} is the best one among them. So, some better solutions in the swarm are selected for further search. Compared with most ABC algorithms, our method does not need to calculate the selection probability of each solution.

Different from the above standard scout bee search phase, U_1 and U_2 will be generated when the X_i is abandoned. Then, X_i will be replaced by the best solution between U_1 and U_2. In this study, U_1 is obtained from standard scout bee search phase according to Eq. (29). U_2 is the optimum value from the solution of X_{ib}, which can be expressed by Eq. (30).

$$u_{id} = x_{ibd} + rand(0,\ 1)\cdot(x_{r1d} - x_{r2d}) \tag{30}$$

In which, $r_1 \neq r_2 \neq ib$ represents the selected nectar source is different from x_{ib}.

In this work, the proposed MDU-ABC-K algorithm is conducted to solve the dynamic optimization model by minimizing the comprehensive cost under the constraint of the economic dependence and critical component, thus reducing the maintenance time and cost simultaneously. For the specific optimization problems, the optimal strategy can be outputted according to the intelligent decision-making optimization part of Fig. 1. The procedure of the proposed intelligent decision-making optimization consists of four steps:

Step 1. Enumerate a possible maintenance strategy L by considering each component's current state and each system's maintenance level, $L = (l^1_{a_1,b_1}, l^2_{a_2,b_2}, \ldots,\ l^n_{a_n,b_n})$.
Step 2. Check the desirability of strategy L by Rule 1. If the strategy L meets Rule 1, continue to execute; otherwise, go to Step 3. Rule 1 is based on the system's structure, i.e., the critical components must be repaired to an optimal state. The critical components are meant to be important and must be maintained to improve the health condition of the system, i.e., components that contribute significantly to changes in the system's state.
Step 3. Check the desirability of strategy L by Rule 2. If the strategy L meets Rule 1, continue to execute; otherwise, go to Step 4. Rule 2 is based on the probability of success of system maintenance. Through the model calculation, we obtain the condition for the successful system maintenance: at the end of the maintenance, the number of components in the system that reach state 4–6 is greater than two-thirds of the total components.
Step 4. Check if strategy L satisfies economic dependence. If strategy L satisfies economic dependence, output the optimal maintenance cost and actions by fitness function 1 or 2. The fitness functions 1 and 2 are shown in Eqs. (31) and (32) respectively.

$$\textbf{Function 1}: \ C_T(\mathbf{X}) = k_1 C_M(\mathbf{X}) + k_2 T_M(\mathbf{X}) \tag{31}$$

$$\textbf{Function 2}: \ C_T(\mathbf{X}) = k_1 C'_M(\mathbf{X}) + k_2 T'_M(\mathbf{X}) \tag{32}$$

5. Case study

5.1. Datasets and evaluation metrics

According to the experiments from Refs. [4,27,40-42], we utilized the C-MAPSS (Commercial Modular Aero-Propulsion System Simulation) dataset in our study. It is an open-access dataset and is widely used in prognostics fields. Each flight cycle data in the NASA dataset was measured through 21 sensors under 3 operation conditions. In which, single-cycle data is a 24-dimensional eigenvector. The state of engine units is healthy in the early operation stage, and the engine performance begins to degrade until the failure occurs in the training text. In the test dataset, the degradation process ends some time prior to system failure. The information of the dataset is listed in Table 3.

The dataset consists of 4 sub-datasets: FD001, FD002, FD003 and FD004. Four datasets were simulated under different combinations of operational conditions and fault modes. The subsets FD001 and FD003 are subject to a single operating condition while the subsets FD002 and FD004 present six operating conditions. In addition, there exists only one fault mode in the subsets FD001 and FD002 while the subsets FD003 and FD004 are with two fault modes.

In this study, the total information of the engines states during the whole life cycles are necessary for further maintenance activities. Thus, the FD001 training set is divided into two parts, the first 80 engines are used for training, the last 20 engines are conducted to evaluate the performance of the proposed method.

Two metrics are used for evaluating the performance of the proposed adaptive C-Transformer network, i.e. root mean square error (RMSE) and score, which can be illustrated as follows:

$$RMSE = \sqrt{\frac{1}{n_T}\sum\nolimits_{i=1}^{n_T}(Acty_{RUL\ i} - Predy_{RUL\ i})^2} \tag{33}$$

$$score = \sum_{p=1}^{p} s_p,\ s_p\begin{cases} e^{-\frac{d_p}{13}} - 1\ for\ d_p < 0 \\ e^{-\frac{d_p}{10}} - 1\ for\ d_p \geq 0 \end{cases} \tag{34}$$

where n_T is the number of test samples, $Acty_{RUL\ i}$ and $Predy_{RUL\ i}$ are the actual RUL and predicted RUL of the i th test sample respectively. In Eq. (34), where $d_p = \widehat{RUL}_p - RUL$, $\widehat{RUL}_p$ and RUL denote the predicted RUL and the label of p-th engine unit, respectively.

In addition, four metrics of the binary classification are used to evaluate the performance of the proposed model in predicting whether the component will failure in the next mission, i.e. precision, recall, F-score and accuracy. The mentioned metrics are explained in detail as follows:

$$Precision = \frac{TP}{FP + TP} \times\ 100\% \tag{35}$$

$$Recall = \frac{TP}{TP + FN} \times\ 100\% \tag{36}$$

$$F - Score = \frac{2\ \times Precision \times Recall}{Precision + Recall} \tag{37}$$

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{38}$$

where TP, TP, FP, and FN are the number of true positives, true negatives, false positives, false negatives, respectively.

5.2. Implementation details

Our experiments are run on a personal computer with Intel Core i7–11700KF (3.6 GHz) CPU, 16.0GB memory, and Microsoft Windows 10 operating system. We implement the model using Python 3.7 and with open-source software library Pytorch as a backend. For training the adaptive C-Transformer, we used the Adam optimizer with a learning rate of 5×10^{-5} and the epoch set to 200. Meanwhile, the mean squared error (MSE) is utilized as a loss function for predicting the RUL [41] and Cross-entropy is utilized as a loss function for predicting the component failure probability. The hyperparameters and implementation details of the transformer network are: $N = 2$, $d_{model} = 64$, $d_k = 32$, $d_v = 2$, $h = 8$. For the CNN network, the time window w is set to 30 as described in 4.1.2. Furthermore, the iterations setting for the MDU- ABC-K algorithm is 200.

Table 3
Characteristics of four datasets.

Dataset	Engine unit	Fault model number	Operation condition
FD001	100	1	1
FD002	259	1	6
FD003	100	2	1
FD004	248	2	6

5.3. Experimental results and performance analysis

In this section, the prognostic performance of the proposed architecture for RUL prediction is presented. The comparisons with other networks are carried out to show the effectiveness of the proposed architecture. Furthermore, the superiority of the proposed architecture is demonstrated by comparing with the latest state-of-the-art prognostic results on the same C-MAPSS dataset.

5.3.1. Prognostic performance

5.3.1.1. RUL prediction. In this section, the FD001 testing set is used for testing to better demonstrate the superiority of C-Transformer. The RUL prediction results of the testing engine units in FD001 regarding the last recorded data point are presented in Fig. 5. The testing engine units are sorted by labels from small to large for better observation and analysis. It can be observed that the predicted RUL values by the proposed adaptive C-Transformer network are closer to the actual values. Especially, the prognostic accuracy tends to be higher in the region where the RUL value is small. That is because when the engine unit is working close to failure, the fault feature is enhanced and that can be captured by the proposed architecture for better prognostics.

Fig. 6 plots the RUL prediction results of engine unit numbers 21, 31, 34, and 81 in the FD001 test set. As shown in Fig. 6, during the healthy condition of the engine unit (flatten stage of the label), it shows that precise RUL prediction where the red lines are almost overlapping with the blue lines. When the degradation begins, the prediction translates smoothly and then decreases almost linearly with time until the end of the period. All of the four engine units show precise and accurate RUL predictions when close to the end of the period. However, near the point of degradation, it appears that the evolution of the predictions does not match the given labels. This is due to the fact that not all of the engine units begin to degrade when RUL equals to 120 and degrade suddenly without transition. For the implementation of the proposed method in industry, the accuracy of the prediction is more importance when the engines work well and when they go to fault. Meanwhile, the accurate evaluation of the engine status in the late period is able to enhance operation reliability and safety, thus reduce maintenance costs and improve the whole system performance.

5.3.1.2. Failure probability prediction. To better carry out the further maintenance activities, the component failure probabilities are calculated through the proposed adaptive C-Transformer-CNN, i.e. the probability of whether a system will fail in the next mission. The confusion probability matrix is conducted to evaluate the accuracy of failure probability prediction. Fig. 7 shows the confusion matrix of the test set in the FD001.

In this study, the last 20 engines in FD001 are conducted to evaluate the performance of the proposed method. In which, the sequence length of adaptive C-Transformer-CNN is set as 30 [31]. Fig. 8 plots the failure probability prediction results of case engine units 82 and 83 in the FD001. As illustrated in Fig. 8, the system may not have much loss at the beginning of operation, thus the decision maker does not need to perform maintenance activities in this scenario. It can also be noted that the failure probability in some inspection periods of the life cycle increases sharply. Therefore, the dynamic multilevel predictive

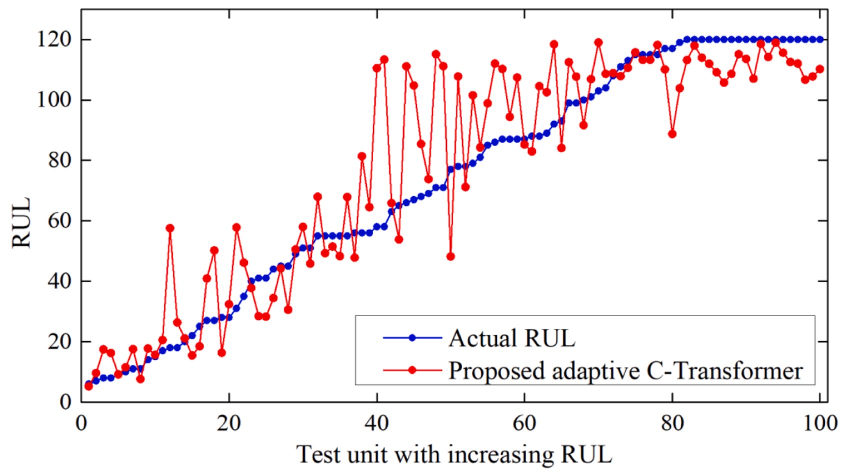

Fig. 5. Sorted prediction for the 100 testing engine units in FD001.

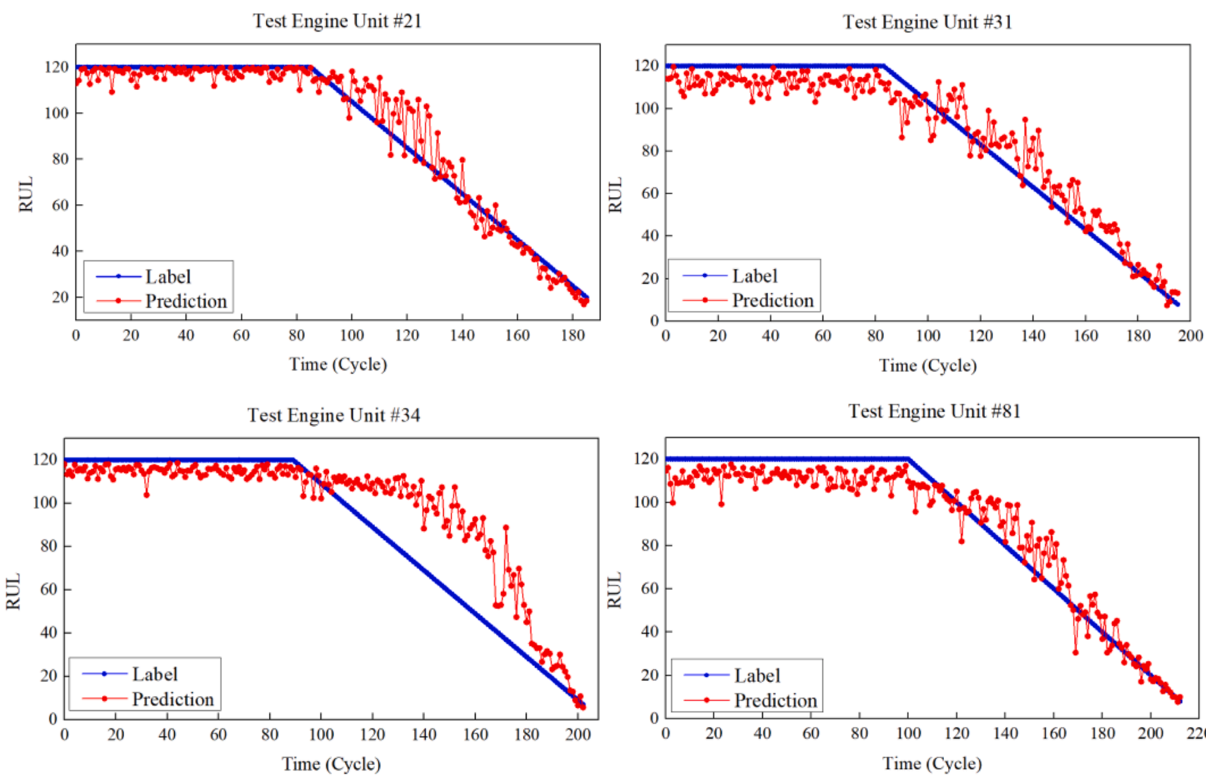

Fig. 6. RUL prediction results of engine unit 21, 31, 34, and 81 in FD001 test set.

maintenance must be performed during these inspection periods.

5.3.2. *Comparison with other methods*

5.3.2.1. RUL prediction. The proposed adaptive C-Transformer network is able to provide a systematic and accurate prognostic. In order to show the superiority of the proposed architecture to existing networks, its testing performance is compared with those of other networks in this paper.

In this study, several existing methods including RNN, GRU and LSTM are conducted to predict the RUL results on the same dataset. The comprehensive comparison results of the RUL prognostic performance are illustrated in Table 4. It can be concluded that the proposed adaptive C-Transformer network shows the better performance with its RMSE and Score metrics are both having smaller values for all four sub-sets in C-MAPSS dataset. In which, the comparison curves between the proposed adaptive C-Transformer and LSTM are drawn and illustrated in Fig. 9. From Fig. 9, it can be found that the predicted curve by proposed C-Transformer network is closer to the actual RUL, which proves the superiority of the proposed architecture.

5.3.2.2. Failure probability prediction. In this study, the proposed adaptive C-Transformer-CNN is used to calculate the component failure probability for further maintenance activities. The superiority of the proposed method is demonstrated by comparing it to other methods. Table 5 lists each method's calculated accuracy, precision, recall, and F1-score. The comparison results indicate the proposed method is more precise and superior to other conventional methods, proving that the

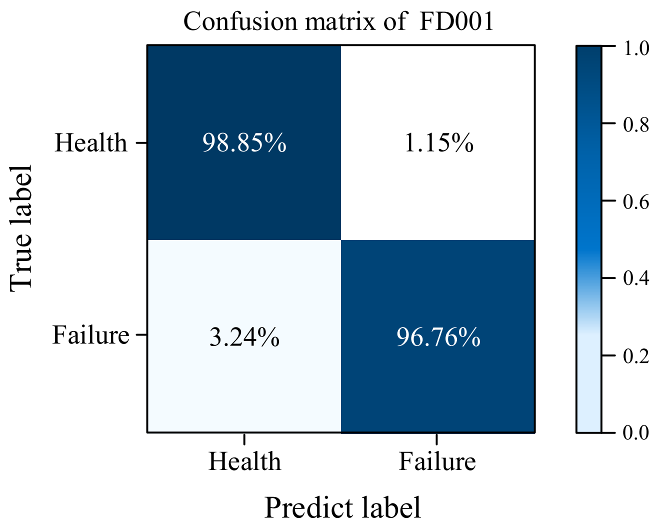

Fig. 7. Confusion matrix of the test data in the FD001.

proposed method can effectively predict the component failure probability and has excellent generalization under various datasets.

5.4. Dynamic multilevel predictive maintenance

To further study the maintenance of series-parallel MSS, we take the C-MAPSS dataset as the example. In this work, refer to the exiting PdSM framework in [29], turbofans engines are defined as components. The data in FD001 is randomly selected before test, which is correspond to the components in the four cases to predict the RUL and failure probability of each component, and the dynamic maintenance strategy is output according to the proposed MPM framework. In this study, five typical cases in response to different maintenance levels are investigated. In which, Level 0 represents that no maintenance action is carried out, thus the corresponding case is not covered in detail.

5.4.1. Case 1

In previous research [13], the system's current state of multi-state components is generally determined by the operator's experience. To address this issue, multilevel failure is introduced to reflect the current state of components that closely combine the failure probability with the series-parallel MSS.

The series-parallel MSS with $K=6$, $M=3$, N_1=3, N_2=2, and N_3=3, is considered in case 1. In which, each subsystem consists of several identical components in parallel. In addition, the critical component is defined as the first component of each subsystem as shown in Fig. 10. The mission time (time window w) is set as 30 cycles [4] and the break time between two successive missions is set as 45 units, the maintenance time must be within this period. The maintenance cost and time matrices for each maintenance action of the component in subsystem i are:

$$\mathbf{C}_1=\begin{bmatrix}0&3&4&7&8&12&26\\0&0&6&9&13&14&16\\0&0&0&7&10&13&15\\0&0&0&0&14&18&22\\0&0&0&0&0&12&14\\0&0&0&0&0&0&10\\0&0&0&0&0&0&0\end{bmatrix},\ \mathbf{C}_2=\begin{bmatrix}0&5&8&11&13&17&26\\0&0&7&9&11&13&16\\0&0&0&7&9&10&15\\0&0&0&0&10&11&22\\0&0&0&0&0&16&14\\0&0&0&0&0&0&10\\0&0&0&0&0&0&0\end{bmatrix}$$

$$\mathbf{C}_3=\begin{bmatrix}0&5&8&11&13&16&23\\0&0&5&8&10&12&15\\0&0&0&6&10&14&18\\0&0&0&0&10&15&19\\0&0&0&0&0&19&20\\0&0&0&0&0&0&18\\0&0&0&0&0&0&0\end{bmatrix},\ \mathbf{T}_1=\begin{bmatrix}0&2&4&6&7&9&12\\0&0&4&6&7&8&10\\0&0&0&7&8&9&11\\0&0&0&0&9&10&12\\0&0&0&0&0&10&11\\0&0&0&0&0&0&10\\0&0&0&0&0&0&0\end{bmatrix}$$

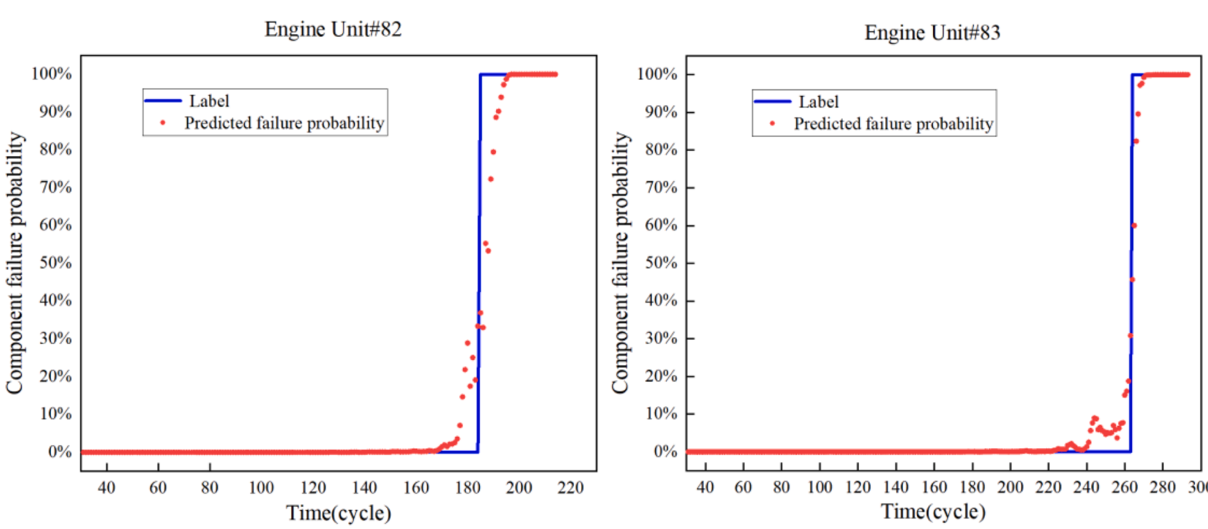

Fig. 8. Failure probability results of engine units 82 and 83 in the FD001.

Table 4
Comparison between different methods.

Methods	FD001		FD002		FD003		FD004	
	RMSE	Score	RMSE	Score	RMSE	Score	RMSE	Score
RNN	17.54	1680.26	18.8	3018	19.81	2279.66	23.95	9769.99
GRU	16.79	791	18.41	2466.85	18.75	1754.44	24.56	9432.57
LSTM	17.1	1005.88	18.16	2466.93	20.27	2347.17	23.07	5209.24
C-Transformer	13.79	475.46	16.11	2214.59	17.1	939.1	19.77	3237.37

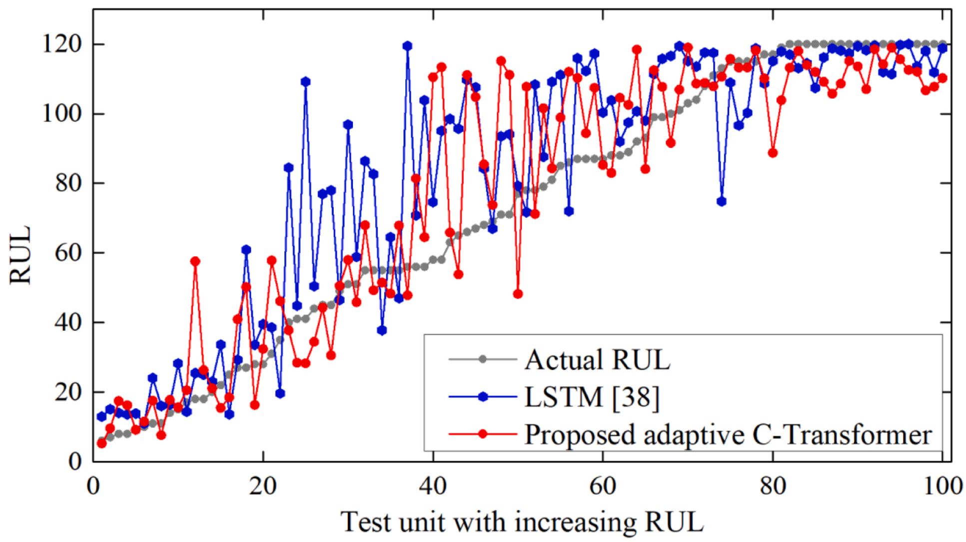

Fig. 9. Comparison of the RUL prediction performance between the proposed C-Transformer and the LSTM model.

Table 5
Comparison between the proposed method and other methods using FD001 dataset.

Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
CNN	94.32	90.25	89.72	91.32
LSTM	96.77	94.12	96.96	95.52
Adaptive C-Transformer-CNN	**97.63**	**96.88**	**97.20**	**96.82**

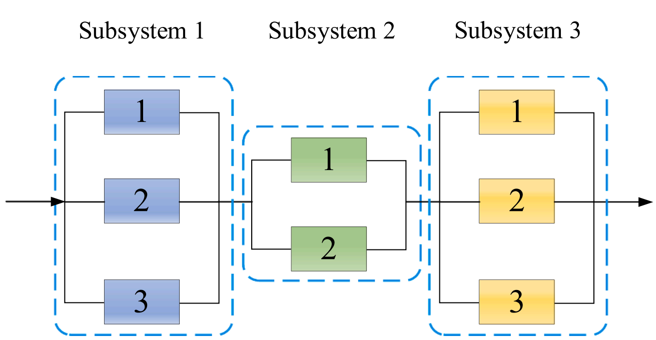

Fig. 10. Series-parallel system example.

$$\mathbf{T}_2 = \begin{bmatrix} 0 & 2 & 4 & 6 & 7 & 9 & 12 \\ 0 & 0 & 5 & 7 & 8 & 9 & 10 \\ 0 & 0 & 0 & 5 & 7 & 9 & 10 \\ 0 & 0 & 0 & 0 & 8 & 9 & 10 \\ 0 & 0 & 0 & 0 & 0 & 9 & 11 \\ 0 & 0 & 0 & 0 & 0 & 0 & 10 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \mathbf{T}_3 = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ 0 & 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 0 & 0 & 2 & 3 & 4 & 5 \\ 0 & 0 & 0 & 0 & 1 & 2 & 3 \\ 0 & 0 & 0 & 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

We randomly picked out the data from the FD001, and applied them to Case 1. The component failure probability prediction of case 1 is listed in Table 6. Based on the evaluation criterion of state in Table 1, the current states of the components before maintenance can be expressed as:

$$X_{ij} = \begin{bmatrix} 5 & 4 & 4 \\ 5 & 4 & \\ 5 & 4 & 4 \end{bmatrix} \text{ or } X = [5\ 4\ 4\ 5\ 4\ 5\ 4\ 4]$$

As mentioned in Section 4, the subsystem and system failure probabilities can be calculated by Eqs. (20) and (21). The failure probabilities of subsystems 1, 2, and 3 are 0.5702, 0.4942, and 0.5807, respectively, and the system failure probability is 0.5807. Based on the evaluation criterion of maintenance level in Table 2, the maintenance level of case 1 is observed as Level 1.

Table 6
Component failure probability before maintenance for the sub-system *i* of case 1.

Sub-system i	Component j		
	1	2	3
1	0.2017	0.2553	0.2771
2	0.2154	0.3553	—
3	0.1003	0.2772	0.3552

By analyzing the current state of the component, that components (1,2) and (1,3), and components (3,2) and (3,3) are economically dependent, the cost and time saving coefficients of components are both assumed to be 0.6, the "set-up cost" Δc_s and "set-up time" Δt_s are both set as 0. Then, as described in Section 4.3, the optimal maintenance strategy can be obtained through the proposed intelligent decision-making optimization model. For an illustration, Table 7 shows the dynamic maintenance strategies according to the different comprehensive cost coefficients. Take the coefficients $k_1 = 0.7$and $k_2 = 0.3$ as an example, there are various maintenance activities for the components. Component (1,3) is repaired to state 6 from state 4, and the other components remain the state before maintenance.

5.4.2. Case 2

Consider the system in case 1 again with the data selected from

Table 7
The dynamic maintenance strategies of case 1.

Coefficients	Sub-system i	Component j			C_M	T_M	C_T
		1	2	3			
k_1=0.7, k_2=0.3	1	5→5	4→4	4→6	14	11	13.1
	2	5→5	4→4	—			
	3	5→5	4→4	4→4			
k_1=0.3, k_2=0.7	1	5→5	4→4	4→4	22.8	1.2	7.68
	2	5→5	4→4	—			
	3	5→5	4→5	4→5			
k_1=0.5, k_2=0.5	1	5→5	4→4	4→4	20	3	11.5
	2	5→5	4→4	—			
	3	5→5	4→6	4→4			

FD001. The component failure probability prediction of case 2 is listed in Table 8. Based on the evaluation criterion of state in Table 1, the current states of the components before the maintenance can be expressed as:

$$X_{ij} = \begin{bmatrix} 4 & 3 & 3 \\ 3 & 2 & \\ 4 & 3 & 3 \end{bmatrix} \text{ or } X = [4\ 3\ 3\ 3\ 2\ 4\ 3\ 3]$$

The failure probabilities of subsystems 1, 2, and 3 are 0.7469, 0.7401, and 0.7452, respectively, and the system failure probability is 0.7469. Based on the evaluation criterion of maintenance level in Table 2, the maintenance level of case 1 is Level 2.

In this case, components (1,2) and (1,3), and (3,2) and (3,3) are economically dependent, the cost and time saving coefficients of components are both assumed to be 0.6, the "set-up cost" Δc_s and "set-up time" Δt_s are both set as 0. Table 9 shows three sets of dynamic maintenance strategies with different coefficients. Take the coefficients $k_1 = 0.7$ and $k_2 = 0.3$ as an example, component (1,1) is repaired from state 4 to 5, components (1,2) and (1,3) remain the origin state before maintenance; component (2,1) is repaired from state 4 to 5, component (2,2) is repaired from state 2 to 5; component(3,1) is repaired from state 4 to 6, components (3,2) and (3,3) are repaired from state 3 to 4 simultaneously.

5.4.3. Case 3

Consider the system in case 1 and the data selected from FD001 again. The component failure probability prediction of case 3 is listed in Table 10. Based on the evaluation criterion of state in Table 1, the current states of the components before maintenance can be expressed as:

$$X_{ij} = \begin{bmatrix} 3 & 2 & 2 \\ 1 & 3 & \\ 4 & 2 & 2 \end{bmatrix} \textbf{ or } X = [3\ 2\ 2\ 1\ 3\ 4\ 2\ 2]$$

The failure probabilities of subsystems 1, 2, and 3 are 0.8962, 0.8330, and 0.8442, respectively, and the system failure probability is 0.8962. Based on the evaluation criterion of maintenance level in Table 2, the maintenance level of case 2 is observed as Level 3.

In this case, components (1,2) and (1,3), and (3,2) and (3,3) are economically dependent, the cost and time saving coefficients of components are both assumed to be 0.6, the "set-up cost" Δc_s and "set-up time" Δt_s are both set as 0. Table 11 shows three sets of dynamic maintenance strategies with different coefficients, which can be obtained through the proposed intelligent decision-making optimization model. Take the coefficients $k_1 = 0.7$and $k_2 = 0.3$ as an example, component (1,1) is repaired to state 3 to 5, components (1,2) and (1,3) remain the origin state before maintenance; component (2,1) is repaired from state 1 to 5, component (2,2) is repaired from state 3 to 5; component (3,1) is repaired from state 1 to 5, components (3,2) and (3,3) are repaired from state 2 to 4 simultaneously.

5.4.4. Case 4

Consider the system in case 1 and the data selected from FD001 again. The component failure probability prediction of case 4 is listed in Table 12. Based on the evaluation criterion of state in Table 1, the current states of the components before the maintenance can be expressed as:

$$X_{ij} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & \\ 2 & 1 & 1 \end{bmatrix} \textbf{ or } X = [1\ 0\ 0\ 2\ 1\ 2\ 1\ 1]$$

The failure probability of subsystems 1, 2, and 3 are 0.7584, 0.8895, and 0.9012, respectively, and the system failure probability is 0.9974. Based on the evaluation criterion of maintenance level in Table 2, the maintenance level of case 1 is observed as Level 4.

In this case, components (1,2) and (1,3), and (3,2) and (3,3) are economically dependent, and the cost and time saving coefficients of components, the "set-up cost" and "set-up time" are all the same as case 1. Three optimal maintenance strategies are obtained through the proposed intelligent decision-making optimization model, as listed in Table 13. Take the coefficients $k_1 = 0.7$and $k_2 = 0.3$ as an example, component (1,1) is repaired from state 1 to 4, components (1,2) and (1,3) are repaired from state 0 to 2 simultaneously; components (2,1) is

Table 8
Component failure probability before maintenance for the sub-system *i* of case 2.

Sub-system i	Component j		
	1	2	3
1	0.2552	0.4117	0.4223
2	0.3996	0.5671	—
3	0.2501	0.4331	0.4007

Table 9
The dynamic maintenance strategies of case 2.

Coefficients	Sub-system i	Component j			C_M	T_M	C_T
		1	2	3			
k_1=0.7, k_2=0.3	1	4→5	3→3	3→3	65	32.2	55.16
	2	4→5	2→5	—			
	3	4→6	3→4	3→4			
k_1=0.3, k_2=0.7	1	4→6	3→3	3→3	67	30.2	41.24
	2	4→6	2→4	—			
	3	4→5	3→4	3→4			
k_1=0.5, k_2=0.5	1	5→5	4→3	4→3	66	31.2	48.6
	2	5→6	4→4	—			
	3	5→6	4→4	4→4			

Table 10
Component failure probability before maintenance for the sub-system *i* of case 3.

Sub-system i	Component j		
	1	2	3
1	0.4062	0.5591	0.6035
2	0.7113	0.4215	—
3	0.3011	0.5502	0.5663

Table 11
The dynamic maintenance strategies of case 3.

Coefficients	Sub-system i	Component j			C_M	T_M	C_T
		1	2	3			
k_1=0.7, k_2=0.3	1	3→5	2→2	2→2	74	34.6	62.18
	2	1→5	3→5	—			
	3	4→6	2→4	2→4			
k_1=0.3, k_2=0.7	1	3→5	2→2	2→2	76.8	32.8	45.99
	2	1→5	3→4	—			
	3	4→5	2→5	2→5			
k_1=0.5, k_2=0.5	1	3→5	2→2	2→2	75	33.6	54.3
	2	1→5	3→6	—			
	3	4→5	2→4	2→4			

Table 12
Component failure probability before maintenance for the sub-system *i* of case 4.

Sub-system i	Component j		
	1	2	3
1	0.7584	0.8895	0.9012
2	0.5665	0.7108	—
3	0.6337	0.8004	0.7324

repaired from state 2 to 5, component (2,2) is repaired from state 1 to 4; component (3,1) is repaired from state 3 to 5; components (3,2) and (3,3) are repaired from state 1 to 5 simultaneously.

5.4.5. Flexibility of mpm decisions

The advantage of the proposed MPM framework is that the optimum maintenance decisions are made based on the prognostics information and comprehensive cost, which can evaluate the comprehensive cost of maintenance strategy at the decision time, thus bring more flexibility for decisions to well adapt to the maintenance cost and time change. During decision-making, if the manager pays more attention to saving maintenance cost, the value of k_1 will be greater than k_2; On the other hand, if the manager prefers to save more maintenance time, k_2 is greater than k_1, as described in Section 4.3. In this study, by considering the comprehensive cost, the proportion of maintenance cost k_1 and the proportion of maintenance time k_2 are set as three groups (0.7 and 0.3, 0.3 and 0.7, 0.5 and 0.5) to make the maintenance decision.

Fig. 11 illustrates the maintenance decisions for all cases. For ideal case, the optimal decision is to do nothing. For other four cases, the failure probability level not only determine the maintenance level but also present an increasing trend as failure probability increases. Moreover, more flexibility decisions of each maintenance level can be made based on manager prefers. Such as the listed in Table 6 of case 1, the total maintenance cost C_M is the smallest of three strategies when $k_1 > k_2$, the total maintenance time T_M is the smallest of three strategies when $k_2 > k_1$, and the same is true in other cases.

5.5. Comparison with other models

This study conducted the optimal dynamic maintenance strategy considering the comprehensive cost under economic dependence and critical components constraints, rather than the minimum maintenance cost only under a certain threshold value of intermission break time [14]. Through the proposed intelligent dynamic decision-making optimization, as described in Section 4.3, the optimal dynamic maintenance strategies are the output in response to four cases, as shown in Fig. 12. The proposed dynamic optimization model is compared with the following models:

1 Hadis's model [4]: In this model, the optimal maintenance strategy is output with the object of the minimum maintenance cost only under a certain threshold value of intermission break time. Thus, the cost and time saving coefficients of components are both assumed to be 1, the "set-up cost" Δc_s and "set-up time" Δt_s are both set as 0. In addition, the proportion of maintenance cost k_1 and the proportion of maintenance time k_2 are set as 0.5 and 0.5.
2 Cuong's model [11]: Economic dependence is considered in this model. In which, the cost and time saving coefficients of components are both assumed to be 0.6, the "set-up cost" Δc_s and "set-up time" Δt_s are both set as 0. Comprehensive cost is not considered, thus the proportion of maintenance cost k_1 and the proportion of maintenance time k_2 are set as 0.5 and 0.5.

Table 13
The dynamic maintenance strategies of case 4.

Coefficients	Sub-system i	Component j			C_M	T_M	C_T
		1	2	3			
k_1=0.7,	1	1→5	0→2	0→2	68.2	38.6	59.32
k_2=0.3	2	2→5	1→4	—			
	3	2→5	1→5	1→5			
k_1=0.3,	1	3→5	2→2	2→2	68.2	38.6	47.48
k_2=0.7	2	1→5	3→4	—			
	3	4→5	2→5	2→5			
k_1=0.5,	1	1→5	0→2	0→2	68.2	38.6	53.4
k_2=0.5	2	2→5	1→4	—			
	3	2→5	1→5	1→5			

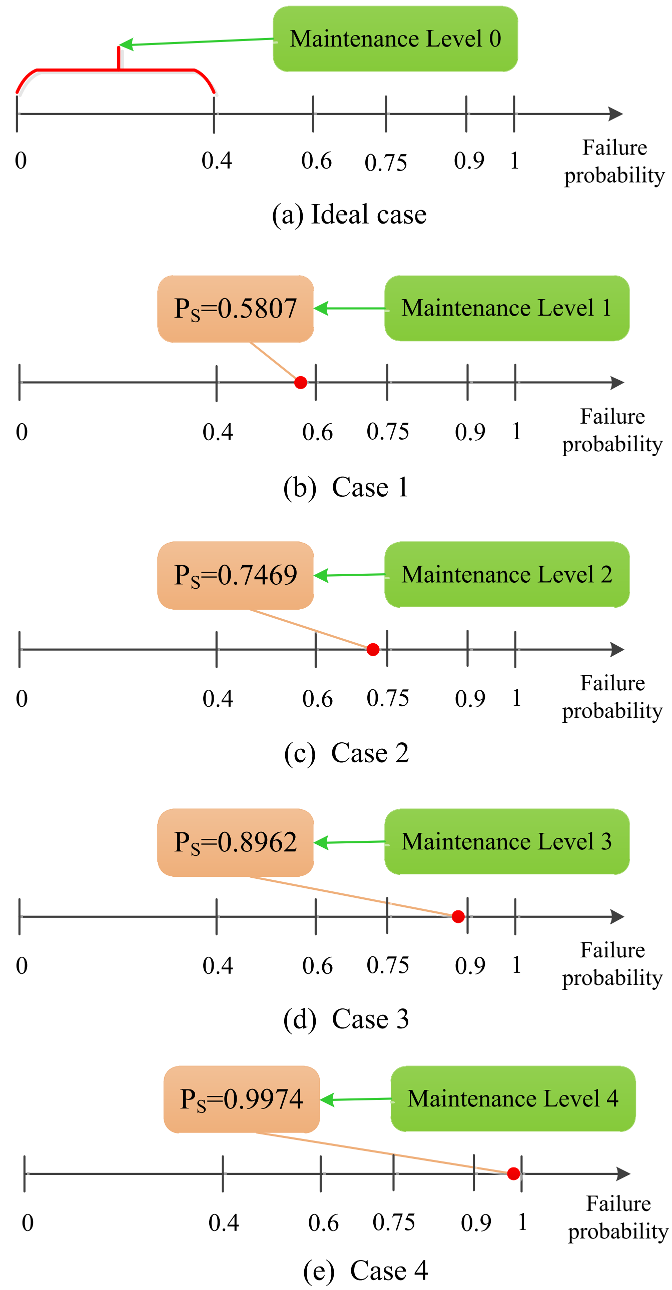

Fig. 11. Illustration of the maintenance decisions for all cases.

Fig. 12 plots the comparison results of the optimal maintenance cost calculated by different models. The comparison results show that the optimal maintenance strategy obtained by our method is flexibility, which produces multi strategies with different comprehensive cost coefficients. For these four cases, the k_1=0.7, k_2=0.3 and k_1=0.3, k_2=0.7 are upper and lower bounds of the comprehensive maintenance cost coefficients in this study. As shown in Fig. 12, in Cases 3 and 4, when considering only the minimum maintenance cost under the threshold time limit, the total maintenance resources consumed are far greater than when the comprehensive cost is considered. The reason is that the step spanning the state before and after maintenance is large, which requires more maintenance resources; thus, the difference will be more evident. However, in Cases 1 and 2, the difference between them is small because low maintenance levels require fewer maintenance resources; thus, extreme situations will rarely occur. The comparison results show that optimal dynamic maintenance strategy considering the comprehensive cost can indeed achieve the purpose of saving resources.

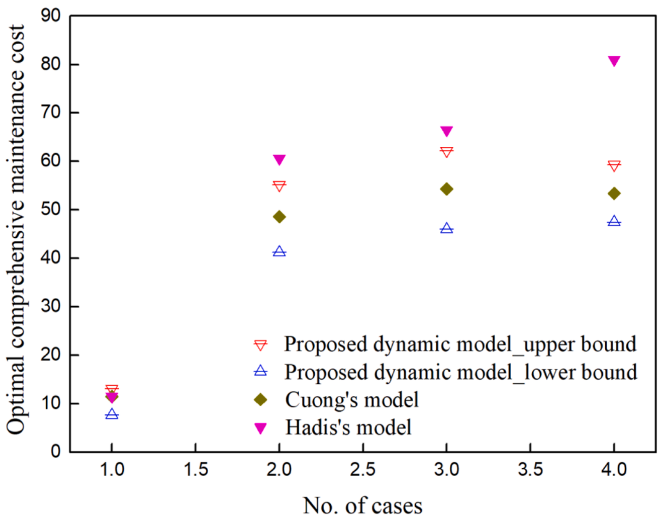

Fig. 12. Comparisons of the optimal comprehensive cost calculated by different models.

6. Conclusion

This study proposes a novel deep learning-based intelligent MPM framework for series-parallel MSS considering multilevel maintenance and comprehensive cost. Its advantages are reflected in two aspects: prognostic accuracy and intelligent decision-making. For prognostic accuracy, a new adaptive C-Transformer network is constructed to predict component RUL accurately through extracting features adaptively. Based on this, the component failure probability is obtained through CNN network. In addition, using the trial-and-error learning method, the series-parallel MSSs maintenance level is established based on multilevel failure through the predicted results, which reduces the computational complexity and avoids the wrong maintenance decisions at decision-making stage.

Regarding the intelligent decision-making aspect, the proposed framework includes a dynamic optimization model capable of outputting the optimal maintenance strategy, which consists of comprehensive maintenance cost and maintenance actions. Meanwhile, the new MDU-ABC-K algorithm is conducted to solve the proposed decision-making optimization model by minimizing the comprehensive cost under the constraint of the economic dependence and critical component, thus balancing the maintenance time and cost simultaneously.

The proposed approach compares with other models through turbofan engine data set by NASA. The comparison results of component failure probability show that the proposed adaptive C-Transformer network is superior to the other networks. In addition, the comparison results of maintenance decision show that the proposed approach considering the multilevel maintenance and comprehensive cost, can save more maintenance resources, which not only prove the superiority and generality of the proposed approach but also can help decision-makers select the dynamic maintenance actions that should be performed in response to different maintenance levels at each break.

In terms of maintenance resources, only comprehensive maintenance cost under the maintenance cost and time restriction is considered in the proposed framework. Further work should integrate spare parts and labor allocation to reduce ineffective maintenance.

CRediT authorship contribution statement

Kai-Li Zhou: Conceptualization, Methodology, Writing – original draft. **De-Jun Cheng:** Supervision, Conceptualization, Writing – review & editing. **Han-Bing Zhang:** Writing – review & editing. **Zhong-tai Hu:** Writing – review & editing. **Chun-Yan Zhang:** Writing – review & editing.

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

No data was used for the research described in the article.

Acknowledgments

The authors would like to thank the comments and suggestions from reviewers and the Editor are very much appreciated.

References

[1] Alsyouf I. The role of maintenance in improving companies' productivity and profitability. Int J Prod Econ 2007;105(1):70–8. 70-8.
[2] Schneider K, Cassady CRichard. Fleet Performance under Selective Maintenance. IEEE Trans Instrum Meas 2004;4:571–6. 571-6.
[3] Maillart LM, Cassady CR, Rainwater C, Schneider K. Selective maintenance decision-making over extended planning horizons. IEEE Trans Reliab 2009;58(3): 462–9. 462-9.
[4] Hesabi H, Nourelfath M, Hajji A. A deep learning predictive model for selective maintenance optimization. Reliab Eng Syst Saf 2022;219:108191. 108191.
[5] Jiang T, Liu Y. Parameter inference for non-repairable multi-state system reliability models by multi-level observation sequences. Reliab Eng Syst Saf 2017;166:3–15. 3-15.
[6] P D. A few useful things to know about machine learning. Commun ACM 2012;55 (10):78–87. 78-87.
[7] Najafabadi MM, Villanustre F, Khoshgoftaar TM, Seliya N, Wald R, Muharemagic E. Deep learning applications and challenges in big data analytics. J Big Data 2015;2 (1).
[8] Pan T, Chen J, Ye Z, Li A. A multi-head attention network with adaptive meta-transfer learning for RUL prediction of rocket engines. Reliab Eng Syst Saf 2022: 225.
[9] Xu D, Xiao X, Liu J, Sui S. Spatio-temporal degradation modeling and remaining useful life prediction under multiple operating conditions based on attention mechanism and deep learning. Reliab Eng Syst Saf 2023:229.
[10] Ma J, Cai L, Liao G, Yin H, Si X, Zhang P. A multi-phase Wiener process-based degradation model with imperfect maintenance activities. Reliab Eng Syst Saf 2023:109075. 109075.
[11] Pham H, Wang H. Imperfect maintenance. Eur J Oper Res 1996;94(3):425–38. 425-38.
[12] Maaroufi G, Chelbi A, Rezg N. Optimal selective renewal policy for systems subject to propagated failures with global effect and failure isolation phenomena. Reliab Eng Syst Saf 2013;114:61–70. 61-70.
[13] Dao CD, Zuo MJ, Pandey M. Selective maintenance for multi-state series–parallel systems under economic dependence. Reliab Eng Syst Saf 2014;121:240–9. 240-9.
[14] Chaabane K, Khatab A, Diallo C, Aghezzaf EH, Venkatadri U. Integrated imperfect multimission selective maintenance and repairpersons assignment problem. Reliab Eng Syst Saf 2020:199.
[15] Rice W, Cassady C, Nachlas J. Optimal maintenance plans under limited maintenance time. In: Proceedings of the seventh industrial engineering research conference; 1998. p. 1–3.
[16] Cassady CR. Selective maintenance modeling for industrial systems. J Qual Maintenance Eng 2001;7(1):104–17. 104-17.
[17] de Pater I, Mitici M. Predictive maintenance for multi-component systems of repairables with Remaining-Useful-Life prognostics and a limited stock of spare components. Reliab Eng Syst Saf 2021:214.
[18] Jiang T, Liu Y. Selective maintenance strategy for systems executing multiple consecutive missions with uncertainty. Reliab Eng Syst Saf 2020:193.
[19] Pandey M, Zuo Ming J. Selective maintenance considering two types of failure modes. Int J Strat Eng Asset Manage 2014:37–62. 37-62.
[20] Rajagopalan R, Cassady CR. An improved selective maintenance solution approach. J Qual Maint Eng 2006;12(2):172–85. 172-85.
[21] Pandey M, Zuo MJ, Moghaddass R, Tiwari MK. Selective maintenance for binary systems under imperfect repair. Reliab Eng Syst Saf 2013;113:42–51. 42-51.
[22] Mo Y, Liu Y, Cui L. Performability analysis of multi-state series-parallel systems with heterogeneous components. Reliab Eng Syst Saf 2018;171:48–56. 48-56.
[23] Mo Y, Xing L, Cui L, Si S. MDD-based performability analysis of multi-state linear consecutive-k-out-of-n: f systems. Reliab Eng Syst Saf 2017;166:124–31. 124-31.
[24] Zhao X, Wu C, Wang S, Wang X. Reliability analysis of multi-state k-out-of-n: g system with common bus performance sharing. Comput Ind Eng 2018;124:359–69. 359-69.

[25] Cao W, Jia X, Liu Y, Hu Q. Selective maintenance optimization for fuzzy multi-state systems. J Intell Fuzzy Syst 2018;34(1):105–21. 105-21.

[26] Li W, Zuo MJ. Reliability evaluation of multi-state weighted -out-of- systems. Reliab Eng Syst Saf 2008;93(1):160–7. 160-7.

[27] Namuduri S, Narayanan BN, Davuluru VSP, Burton L, Bhansali S. Review—deep learning methods for sensor based predictive maintenance and future perspectives for electrochemical sensors. J Electrochem Soc 2020;167(3).

[28] Zhang Z, Song W, Li Q. Dual-aspect self-attention based on transformer for remaining useful life prediction. IEEE Trans Instrum Meas 2022;71:1–11. 1-11.

[29] Wu Y, Yuan M, Dong S, Lin L, Liu Y. Remaining useful life estimation of engineered systems using vanilla LSTM neural networks. Neurocomputing 2018;275:167–79. 167-79.

[30] Liu J, Li Q, Chen W, Yan Y, Qiu Y, Cao T. Remaining useful life prediction of PEMFC based on long short-term memory recurrent neural networks. Int J Hydrogen Energy 2019;44(11):5470–80. 5470-80.

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

[32] Lee J, Mitici M. Deep reinforcement learning for predictive aircraft maintenance using probabilistic Remaining-Useful-Life prognostics. Reliab Eng Syst Saf 2023: 230.

[33] Chen C, Liu Y, Sun X, Cairano-Gilfedder CD, Titmus S. An integrated deep learning-based approach for automobile maintenance prediction with GIS data. Reliab Eng Syst Saf 2021:216.

[34] He X, Wang Z, Li Y, Khazhina S, Du W, Wang J, et al. Joint decision-making of parallel machine scheduling restricted in job-machine release time and preventive maintenance with remaining useful life constraints. Reliab Eng Syst Saf 2022:222.

[35] Nourelfath M, Châtelet E. Integrating production, inventory and maintenance planning for a parallel system with dependent components. Reliab Eng Syst Saf 2012;101:59–66. 59-66.

[36] Dao CD, Zuo MJ. Selective maintenance of multi-state systems with structural dependence. Reliab Eng Syst Saf 2017;159:184–95. 184-95.

[37] Dao CD, Zuo MJ. Optimal selective maintenance for multi-state systems in variable loading conditions. Reliab Eng Syst Saf 2017;166:171–80. 171-80.

[38] 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. 1-11.

[39] Zhao S, Zhang Y, Wang S, Zhou B, Cheng C. A recurrent neural network approach for remaining useful life prediction utilizing a novel trend features construction method. Measurement 2019;146:279–88. 279-88.

[40] Saxena A., Geodel K. Turbofan engine degradation simulation data set. NASA Ames Prognostics Data Repository; 2008. p. 878–87. CA.

[41] Rathore MS, Harsha SP. An attention-based stacked BiLSTM framework for predicting remaining useful life of rolling bearings. Appl Soft Comput 2022:131.

[42] Asif O, Haider SA, Naqvi SR, et al. A deep learning model for remaining useful life prediction of aircraft turbofan engine on C-MAPSS dataset. IEEE Access 2022;10: 95425–40.