Attention-Based Multi-scale CNN and LSTM Model for Remaining Useful Life Estimation

DUAN Jiajun[1,2], *LU Zhong*[1*], *DU Zhiqiang*[1]

1. College of Civil Aviation, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China;
2. Sichuan Aerospace Zhongtian Power Equipment Co., Ltd., Chengdu 610100, China

(Received 16 June 2025; revised 25 September 2025; accepted 10 October 2025)

Abstract: Current aero-engine life prediction areas typically focus on single-scale degradation features, and the existing methods are not comprehensive enough to capture the relationship within time series data. To address this problem, we propose a novel remaining useful life (RUL) estimation method based on the attention mechanism. Our approach designs a two-layer multi-scale feature extraction module that integrates degradation features at different scales. These features are then processed in parallel by a self-attention module and a three-layer long short-term memory (LSTM) network, which together capture long-term dependencies and adaptively weigh important feature. The integration of degradation patterns from both components into the attention module enhances the model's ability to capture long-term dependencies. Visualizing the attention module's weight matrices further improves model interpretability. Experimental results on the C-MAPSS dataset demonstrate that our approach outperforms the existing state-of-the-art methods.

Key words: attention mechanism; convolutional neural network (CNN); long short-term memory (LSTM); multi-scale feature extraction

CLC number: V240.2 **Document code:** A **Article ID:** 1005-1120(2025)S-0064-14

0 Introduction

As a key technology for prediction and health management, the remaining useful life (RUL) prediction aims to predict the time interval between the current moment and the end of an aero-engine's life based on current and historical monitoring data. Nowadays RUL prediction methods mainly include physical model-based methods and data-driven methods. Physical model-based methods are modeled by in-depth analysis of the performance degradation process and failure mechanisms. Due to the complexity of the real world, physics-based approaches are generally only used for component-level failure and RUL prediction. Orsagh et al.[1] investigated the bearings of turbine engines and proposed the Yu-Harris model. Chiachío et al.[2] proposed the reliability-based Shear-Lag model for fatigue degradation of composite materials. For some specific failures, such as aero-engine degradation processes disturbed by complex mechanical structures and noise, physical modeling-based approaches are subject to significant limitations.

With the rapid development of sensor technology and industrial IoT, data-driven methods have received a lot of attention, and a large amount of data has been applied to engineering. Data-driven based methods are divided into statistical methods and deep learning methods. The former predicts the relevant properties of an object based on historical failure data, which usually introduces some assumptions and constraints, but these assumptions and constraints are not guaranteed to be correct in practice. Currently, deep learning methods are the mainstream of RUL prediction algorithms. Babu et al.[3] introduced a deep regression method based on convolutional neural network (CNN) for RUL prediction of aero-engines, and proved that it outperforms traditional machine learning methods on publicly

*Corresponding author, E-mail address: luzhong@nuaa.edu.cn.

How to cite this article: DUAN Jiajun, LU Zhong, DU Zhiqiang. Attention-based multi-scale CNN and LSTM model for remaining useful life estimation[J]. Transactions of Nanjing University of Aeronautics and Astronautics, 2025,42(S):64-77.
http://dx.doi.org/10.16356/j.1005-1120.2025.S.006

available datasets. Li et al.[4] proposed a two-dimensional depth CNN, which utilizes large convolution kernels for abstract data extraction, and employs sparse connections and weight-sharing strategies to effectively address the issue of vanishing gradients. Yuan et al.[5] confirmed that the long short-term memory (LSTM) network has better prediction performance in time series than other networks by using the good diagnosis and prediction performance of LSTM in complex operations. Xiang et al.[6] implemented RUL prediction for aero-engines by constructing a multicellular LSTM model. Liu et al.[7] proposed a multi-stage RUL prediction framework based on the fusion of clustering and LSTM for aero-engine RUL prediction. Hinchi et al.[8] proposed an end-to-end deep RUL estimation framework, which uses CNN to extract local features, and then introduces LSTM to capture deep degradation relationships, and finally outputs RUL predictions. Ren et al.[9] combined a deep auto-encoder and a deep neural networks (DNN) to estimate the RUL of rolling bearings. Features based on the auto-encoder can retain important information about the input data. Zhang et al.[10] proposed an integrated DBN-based model that uses a multi-objective evolutionary algorithm to optimize the parameters of deep belief networks (DBN) for RUL prediction. AL-Dulaimi et al.[11] constructed a hybrid deep neural network for the parallel prediction of RUL, which combines the ability of CNN and LSTM to extract spatial and temporal features. Aiming at the problem of large amount of data and high sample dimension, Song et al.[12] proposed a hybrid prediction model combining autoencoder and bidirectional LSTM (BiLSTM), the monitoring data is compressed by autoencoder, and the remote dependence of features is captured by BiLSTM. Chen et al.[13] proposed a deep learning method based on self-attention to predict the RUL of a machine and demonstrated that self-attention is effective in RUL prediction. Xia et al.[14] proposed a distance self-attention network parallel computing method to estimate the RUL of an aero-engine. Liu et al.[15] proposed a dual-attention data-driven system to estimate the RUL of an aero-engine. Liu et al.[16] used preprocessed features as inputs, which were separately fed into multiple parallel bidirectional gated recurrent units (Bi-GRUs) and self-attention (SA) networks. This approach enhanced the accuracy of RUL prediction by reinforcing important features. Chen et al.[17] proposed a RUL prediction method based on residual nested LSTM and SA, and experiments on degradation datasets showed that the prediction error was effectively reduced. Zhang et al.[18] proposed a Bi-GRU method based on temporal SA for aero-engine RUL prediction.

However, the above methods only consider the degradation features of aero-engines for single-scale, ignoring the degradation details that may exist at other scales, and the correlation capture problem between time series is not fully considered. To address these problems, we propose a deep network based on the fusion of multi-scale CNN and LSTM with the attention mechanism (MSCLA), applying the self-attention module to a hybrid method combining multiscale CNN and LSTM, which further extracts the features and emphasizes the long-term dependencies among the temporal data. Furthermore, the problem of capturing long-term dependencies is comprehensively addressed by assigning weights to the RUL-related time-series features using an additional attention module. The main contributions of this paper are as follows:

(1) We propose to use CNN with different sizes of convolution kernels to capture degradation features at different scales, and then fuse these degradation features at different scales to provide more complete information for RUL prediction.

(2) When LSTM studies the correlation of time series data, the attention mechanism highlights and assigns weights to deep features closely related to the degradation process, while also comprehensively considering the long-term dependencies within the time series.

(3) Experiments on the commercial modular aviation power simulation system (C-MAPSS) dataset show that the MSCLA can predict the RUL of aero-engine more accurately.

1 Aero-engine RUL Prediction Framework Based on MSCLA

The RUL prediction framework of an aero-en-

gine based on MSCLA includes data preprocessing, network structure design, model training and performance evaluation. The details are shown in Fig.1.

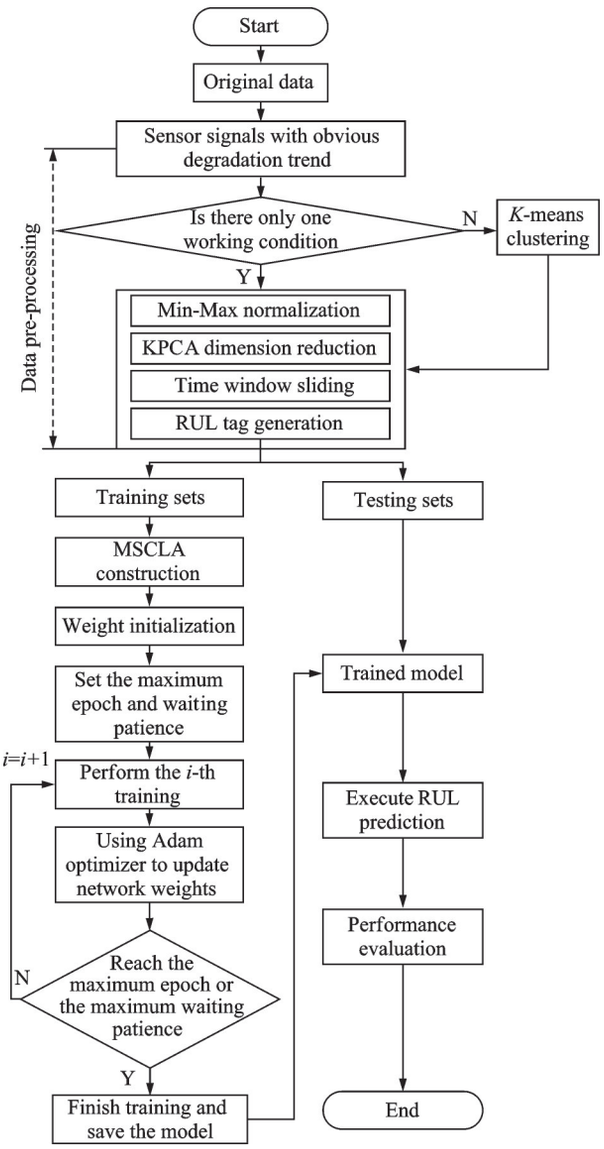

Fig.1 Flow chart of model training process

1.1 Aero-engine dataset description

We use the dataset generated by the C-MAPSS for experimental verification[19]. C-MAPSS is a simulation tool developed for large civil aviation engines. It simulates a thrust of about 90 000 pounds and can perform simulation experiments on engines flying at altitudes of 0 to 40 000 feet, Mach numbers of 0 to 0.9, and sea level temperatures of −60 to 103 °F[20]. The simulation results are given by 26-dimensional time series data, which specifically records the engine number, number of operating cycles, flight parameters (altitude, Mach number and throttle resolver angle), and the measurements of 21 sensors. The specific expressions of these sensors are shown in Table 1. In Table 1, LPC denotes the low-pressure compressor; LPT denotes the low-pressure turbine; HPC denotes the high-pressure compressor; and HPT denotes the high-pressure turbine.

Table 1 Sensor description in the CMAPSS dataset

Symbol	Description	Unit
T2	Total temperature at fan inlet	°R
T24	Total temperature at LPC outlet	°R
T30	Total temperature at HPC outlet	°R
T50	Total temperature at LPT outlet	°R
P2	Pressure at fan inlet	psia
P15	Total pressure in bypass-duct	psia
P30	Total pressure at HPC outlet	psia
Nf	Physical fan speed	r/min
Nc	Physical core speed	r/min
Epr	Engine pressure ratio	—
Ps30	static pressure at HPC outlet	psia
Phi	Ratio of fuel flow to Ps30	pps/psi
NRf	Corrected fan speed	r/min
NRc	Corrected core speed	r/min
BPR	Bypass ratio	—
FarB	Burner fuel-air ratio	—
htBleed	Bleed enthalpy	—
Nf_dmd	Demanded fan speed	r/min
PCNfR_dmd	Demanded corrected fan speed	r/min
W31	HPT coolant bleed	lbm/s
W32	LPT coolant bleed	lbm/s

The C-MAPSS dataset provides four subsets, which represent the degradation data collected by the engine under different operating conditions and fault types. Each subset contains three parts: Training set, test set and test labels. We conduct experiments on all four subsets to verify the performance of the MSCLA network.

Assuming that X is the time series data collected by the aero-engine, N is the total number of life cycles, and V is the total dimension of sensors. Then $\boldsymbol{X}$ can be expressed as

$$\boldsymbol{X}=\begin{Bmatrix} x_{11} & x_{12} & \cdots & x_{1V} \\ x_{21} & x_{22} & \cdots & x_{2V} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{NV} \end{Bmatrix} \tag{1}$$

where x_{NV} refers to the value of the V-dimensional sensor data of the engine at time N. The life prediction value at the next moment is related to the prediction value at the current moment and the sensor

data at the next moment.

This paper aims to use the deep learning network to extract sensor data features from engines, construct a regression model based on these features, and ultimately predict the RUL at the next time step.

1.2 Data pre-processing

When the engine runs under various operating conditions, its degradation characteristics are particularly complex. To eliminate the influence of operating conditions on the original data sequence[21], we used the K-means algorithm[22] to cluster different operating conditions and adopted the Min-Max method for normalizing. The normalization method is shown as

$$x_i^* = \frac{x_{ij}^n - x_{i_{\min}}^n}{x_{i_{\max}}^n - x_{i_{\min}}^n} \tag{2}$$

where x_i^* represents the normalized measurement value of the i-th sensor, and x_{ij}^n the j-th measurement value of the i-th sensor under working condition n. $x_{i_{\max}}^n$ and $x_{i_{\min}}^n$ represent the maximum and minimum values of the i-th sensor data under working condition n, respectively.

Due to the high dimensionality and nonlinearity of the aero-engine degradation dataset, kernel principal component analysis (KPCA)[23] is introduced to reduce the dimension of the normalized data. Additionally, a sparse matrix is employed to enhance the computational efficiency of the dimensionality reduction process.

Sliding window processing relies on two crucial parameters[24]: The window length T and the sliding step size S. Typically, T is set to the minimum time cycle observed in each engine, while S is set to 1. For a multi-sensor time series with a length of N, the data is segmented into $N-T+1$ time windows by sliding along the time axis with a step size of S and a window length of T.

In the field of RUL prediction, it is generally believed that the equipment operates normally in the early stage, experiences wear over time, and subsequently exhibits a decline in performance. The piecewise linear function, as shown in Eq.(3), can be used to mark the useful life of the data.

$$\mathrm{RUL} = \begin{cases} R_{\text{early}} & \mathrm{RUL} \geqslant R_{\text{early}} \\ \mathrm{RUL} & \mathrm{RUL} \leqslant R_{\text{early}} \end{cases} \tag{3}$$

where R_{early} represents the time when the engine begins to degrade, and its value is generally set between 110 and 130[25]. In this paper, 125 is adopted. The basic situation of the pre-processed samples is presented in Table 2.

Table 2 Preprocessed input sample details and labels

Dataset	FD001	FD002	FD003	FD004
Training engine	100	260	100	249
Training sample	17 731	48 819	21 820	56 518
Testing engine	100	259	100	248
Operating condition	1	6	1	6
Fault mode	HPC	HPC	HPC&FAN	HPC&FAN
Input size	30×9	20×10	30×9	20×10

After preprocessing the data, we construct the MSCLA network, which incorporates the FEB and LPB blocks. The network utilizes mean squared error (MSE) as the loss function and employs the Adam optimizer for training. To prevent overfitting, L2 regularization is incorporated. The weights are initialized using the Xavier initializer.

1.3 Model training and performance evaluation

During model training, early stopping is employed to monitor the performance of the validation set, which helps to prevent overfitting and enhance convergence efficiency. The test sets corresponding to the four subsets are input into the model, and the model's performance is evaluated using evaluation indicators.

We use the root mean square error (RMSE) and the score function as the evaluation indicator of the model performance, the formulas are as follows

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{r}_i - r_i)^2} \tag{4}$$

$$\mathrm{Score} = \begin{cases} \displaystyle\sum_{i=1}^{N}\left(\mathrm{e}^{-\frac{\hat{r}_i - r_i}{13}} - 1\right) & \hat{r}_i < r_i \\ \displaystyle\sum_{i=1}^{N}\left(\mathrm{e}^{\frac{\hat{r}_i - r_i}{10}} - 1\right) & \hat{r}_i \geqslant r_i \end{cases} \tag{5}$$

where $\hat{r}_i$ represents the predicted value of RUL, and r_i the actual RUL value. The Score function is adopted due to its asymmetric penalty on the predicted

results. Specifically, when the predicted value of RUL exceeds the true value, a heavier penalty is imposed. This means that the penalty for aggressive prediction results is more severe than that for conservative prediction results, as aggressive predictions often lead to catastrophic accidents.

2 MSCLA Network Structure Design

This section will elaborate on the specific framework of the MSCLA network for aero-engine RUL prediction. The network consists of two components. The feature extraction block and the life prediction block. The network structure of MSCLA is shown in Fig.2, and two components are represented by dotted lines of different colors. Conv represents the convolutional layer, "1×1", "5×5", and "7×7" represent the size of the convolution kernel, the Add layer represents additive fusion, and the Flatten layer reshapes the features from the previous layer into one-dimensional vectors.

After the input layer, we set an initial convolution layer to increase the number of channels, avoid the slow convergence caused by excessive calculation in the deep network, and enhance the integrity of the input information as well as the ability of cross-channel information exchange.

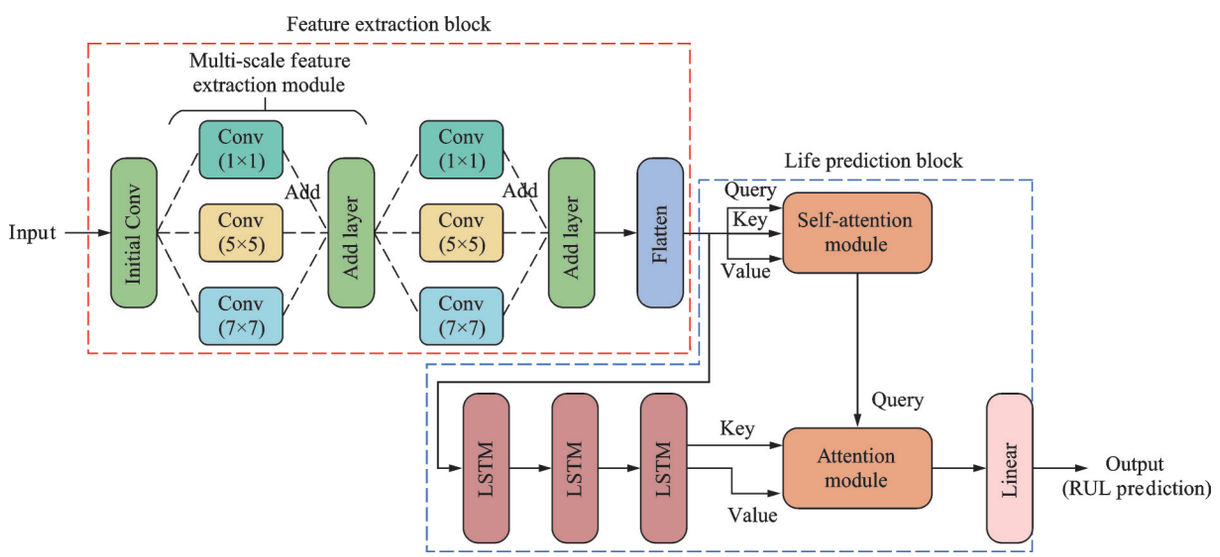

Fig.2 Network structure of MSCLA framework

2.1 Feature extraction block design

The feature extraction block (FEB) is depicted by the red dotted line in Fig.2. Following the initial convolution layer, two layers of the same multi-scale feature extraction module are superimposed to iteratively extract and reconstruct the intrinsic features of the preceding output. The specific details of this module are presented in Fig.3. Each multi-scale feature extraction module comprises three convolutional networks arranged in parallel with different sizes[26], to extract degradation features at various scales and enhance feature extraction efficiency through parallel convolution operations. The process of extracting input features by CNN is described in Eq.(6), and we employ padding to maintain the feature dimension unchanged after the convolution operation.

$$\boldsymbol{y}=\mathrm{ReLU}(\boldsymbol{x}^{l-1}\times\boldsymbol{w}^{l}+\boldsymbol{b}^{l}) \tag{6}$$

where $\boldsymbol{y}$ represents the output feature map of the current convolutional layer, $\boldsymbol{x}^{l-1}$ the feature map of the previous layer, $\boldsymbol{w}^{l}$ the weight matrix of the current layer, $\boldsymbol{b}^{l}$ the bias vector, and ReLU(•) represents the activation function of the convolutional layer.

In order to reduce the sensitivity to network initialization and accelerate the operation of the convolutional network, a batch normalization layer is added after the convolutional layer. The calculation formula for this layer is shown as

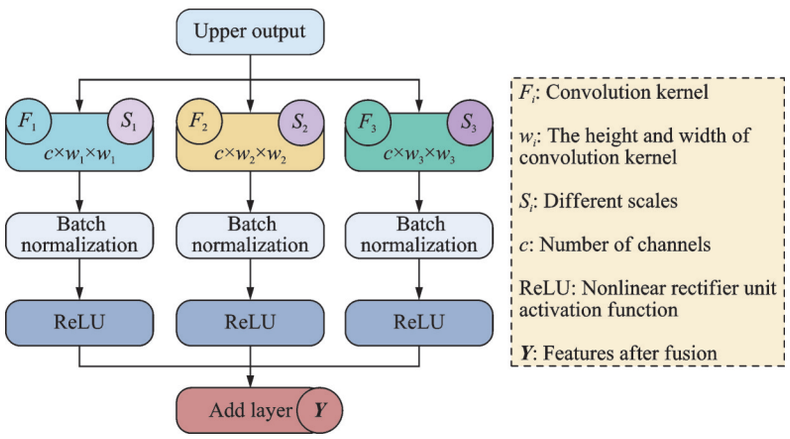

Fig.3 Multi-scale feature extraction module

$$x_B^* = \frac{x_i^j - \mu_B}{\sqrt{{\sigma_B}^2 + \varepsilon}} \tag{7}$$

where x_i^j represents the i-th value in the j-th batch, x_B^* the normalized value, μ_B the mean value of the current batch, σ_B the standard deviation of the current batch, and ε the minimum positive number set to prevent the denominator from being zero.

Finally, we utilize the Add layer to perform the additive fusion of features extracted at different scales. This approach not only retains the unique features extracted by each convolutional network at their respective scales but also enhances the common features after the additive fusion. The features fused by the Add layer contain more comprehensive engine degradation information, which ultimately aids in improving the prediction accuracy. The fusion formula for the Add layer is shown as

$$\boldsymbol{Y}_{\text{add}} = \sum_{i=1}^{c} (\boldsymbol{y}_1^i + \boldsymbol{y}_2^i + \boldsymbol{y}_3^i) \tag{8}$$

where $\boldsymbol{Y}_{\text{add}}$ is the output of the add layer; and $\boldsymbol{y}_1^i$, $\boldsymbol{y}_2^i$, $\boldsymbol{y}_3^i$ represent the feature maps of the ith channel from three different convolutional networks in MFE, respectively.

2.2 Life prediction block design

The life prediction block (LPB) comprises LSTM, attention mechanism[27] and linear regression layer. As illustrated by the blue dotted line in Fig. 2, the fused features obtained after MFE are transformed into one-dimensional vectors through the Flatten layer, and subsequently fed into both the self-attention module and LSTM network.

We use a three-layer LSTM to capture long-term dependencies from input data and learn the correlation between previous samples and the subsequent lifetime labels. Nevertheless, LSTM is susceptible to losing important information when handling very long sequences[28]. To mitigate this issue, we introduce the self-attention mechanism. By swiftly calculating the dependencies between any two positions in the sequence, self-attention enables the model to capture relationships within ultra-long sequences, ultimately achieving more accurate RUL predictions.

2.2.1 Long short-term memory network

The reshaped one-dimensional vectors of the Flatten layer are fed into the three-layer stacked LSTM to capture the long-term dependencies between time series. The LSTM network can effectively alleviate the gradient vanishing and exploding problems of the RNN. The information is filtered through three gating mechanisms, to eliminate redundant information while retaining useful information, thus, effectively combining new information and old memories. The LSTM structure is shown in Fig. 4. The following are the calculation formulas of LSTM.

Forget gate is given as

$$\boldsymbol{f}_t = \sigma(\boldsymbol{W}_{\text{f}} \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_{\text{f}}) \tag{9}$$

Input gate is given as

$$\boldsymbol{i}_t = \sigma(\boldsymbol{W}_{\text{i}} \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_{\text{i}}) \tag{10}$$

Candidate cell state is given as

$$\tilde{\boldsymbol{C}}_t = \tanh(\boldsymbol{W}_{\text{c}} \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_{\text{c}}) \tag{11}$$

New cell state is given as

$$\boldsymbol{C}_t = \boldsymbol{i}_t \otimes \boldsymbol{C}_{t-1} + \boldsymbol{f}_t \otimes \tilde{\boldsymbol{C}}_t \tag{12}$$

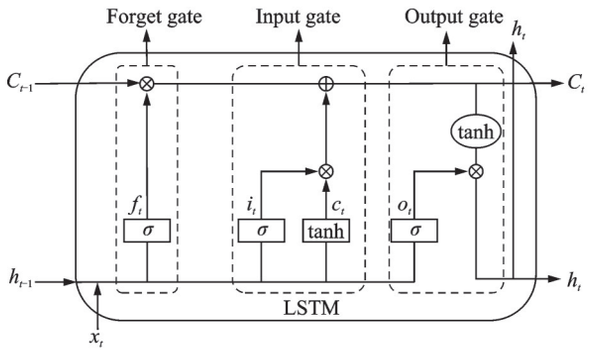

Fig.4 LSTM structure

Output gate is given as

$$\boldsymbol{O}_t = \sigma(\boldsymbol{W}_o \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_o) \tag{13}$$

New hidden value is given as

$$\boldsymbol{h}_t = \boldsymbol{O}_t \otimes \tanh(\boldsymbol{C}_t) \tag{14}$$

where $\boldsymbol{W}$ and $\boldsymbol{b}$ are weight and bias, respectively, which gradually change with the training process.

2.2.2 Attention mechanism

Another parallel route of the reshaped one-dimensional vector is fed to the self-attention module, which automatically gives greater weight to key features through the unique structure of the self-attention itself, allowing the network to focus on parts of the data that are closely related to the degradation process. Meanwhile, the computational efficiency can be efficiently processed through multiple parallel heads.

The input of the attention module comprises query vector ($\boldsymbol{Q}$), key vector ($\boldsymbol{K}$), and value vector ($\boldsymbol{V}$), as vividly depicted in Fig.5. The attention matrix is derived by scaling the dot product of $\boldsymbol{Q}$ and $\boldsymbol{K}$, which is subsequently multiplied by $\boldsymbol{V}$ to yield a weighted data sample. The calculation of the dot product is outlined in Eq.(12).

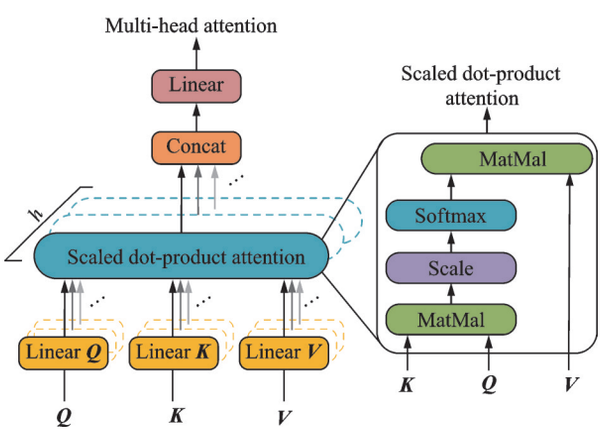

Fig.5 Multi-head attention mechanism and scaled dot product

$$\text{Attention}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}}{\sqrt{d_k}}\right)\boldsymbol{V} \tag{12}$$

where d_k denotes the dimension of $\boldsymbol{K}$.

The calculation method of the multi-head attention is shown as

$$\begin{aligned} y &= \text{MultiHead}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = \\ &\quad \text{concat}(\text{head}_1,\text{head}_2,\cdots,\text{head}_h)\boldsymbol{W}^O \\ \text{head}_i &= \text{Attention}(\boldsymbol{Q}\boldsymbol{W}_i^Q,\boldsymbol{K}\boldsymbol{W}_i^K,\boldsymbol{V}\boldsymbol{W}_i^V) \end{aligned} \tag{13}$$

where h is the number of heads in the multi-head self-attention, and $\boldsymbol{W}^O$, $\boldsymbol{W}_i^Q$, $\boldsymbol{W}_i^K$, $\boldsymbol{W}_i^V$ are all parameters that can be learned during the training process. Each head represents a scaled dot product. Due to the varying correlations of head emphases, the weights obtained in training are also different.

The degradation patterns learned by the three-layer LSTM network are taken as $\boldsymbol{K}$ and $\boldsymbol{V}$, while the deep features closely related to the degradation process learned by the self-attention module serve as $\boldsymbol{Q}$. These are combined as inputs to the attention module. This approach allows the LPB to further emphasize features pertinent to RUL by taking dependencies into account. Consequently, the LPB provides a more comprehensive approach to capturing long-term dependencies in time series prediction problems.

2.2.3 Regression layer

Finally, the knowledge learned by LPB is mapped to the RUL prediction by a linear regression layer containing one neuron. The calculation method of the linear layer is shown as

$$\boldsymbol{y}_{\text{pred}} = \sigma(\boldsymbol{x}_e\boldsymbol{w} + \boldsymbol{b}) \tag{14}$$

where $\boldsymbol{x}_e$ is the output of the attention module, $\boldsymbol{w}$ the weight matrix, $\boldsymbol{b}$ the bias, σ the sigmoid activation function, and $\boldsymbol{y}_{\text{pred}}$ the RUL prediction.

3 Analysis and Discussion of Experimental Results

In order to have more samples for training, the first 80% of each training dataset is selected for model training, and the remaining 20% is used as the validation set. The model is conducted on a

workstation equipped with Intel Xeon Gold 6 226R (2.90 GHz) CPU, 256 GB memory, and Microsoft Windows 10 operating system, and the experimental results are compared with other state-of-the-art (SOTA) results based on C-MAPSS datasets.

3.1 Hyperparameter tuning experiment

3.1.1 Batch size test

Since the model built in this paper adopts a mini-batch training mode, it is necessary to conduct relevant experiments to determine the optimal batch size. With the epoch number set to 100, we utilize early-stopping and vary the batch sizes to assess the model's performance. The relationship between the batch size and the corresponding test results is illustrated in Fig.6.

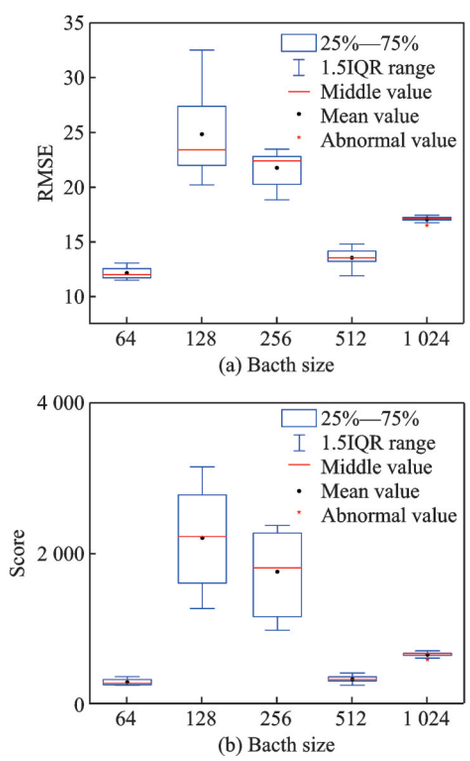

Fig.6 Results from batch size test

Fig.6(a) shows the test results of RMSE, and Fig.6(b) displays the test results of Score. Lower values in both metrics indicate better model performance. As illustrated in Fig.6, model performance does not consistently improve with larger batch sizes. The RMSE and Score are lowest when the batch size is 64 or 512. However, during training, we observed that with a batch size of 64, the validation loss function curve began to increase, indicating overfitting. Therefore, we have set the training batch size to 256 for optimal performance.

3.1.2 Different kernel sizes test

Our proposed network utilizes three different convolution kernels to extract information at different scales, necessitating experiments with different kernel sizes to determine the optimal combination. Due to the dimensions of the four subsets being reduced to 9, 10, 9 and 10, respectively, the maximum size for the convolution kernel is constrained to '9×9'. Initially, we set the convolution kernel sizes based on Ref.[29] to '[1×1, 3×3, 5×5]' (abbreviated as [1, 3, 5], the same below). Larger convolution kernels capture broader degradation trends by extracting large-scale features, while smaller kernels focus on finer details of the degradation process[30]. For our experiments, we consider a range of convolution kernel sizes: [1, 3, 5, 7, 9]. These five kernel sizes were tested in all possible combinations of three kernels per group. The results of these experiments are presented in Fig.7.

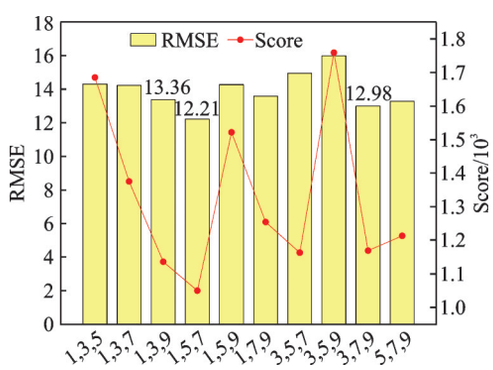

Fig.7 Results of different convolution kernel size combinations

Fig.7 demonstrates that the convolution kernel combination [1, 5, 7] yields the smallest RMSE and Score, followed by [3, 7, 9], while the combination [1, 3, 5] performs the worst. This indicates that the choice of kernel size significantly impacts model performance. Additionally, using either larger or smaller kernels does not consistently improve performance, likely due to the varying sensitivity of different-sized filters in capturing degradation features.

Other hyperparameters are determined through empirical methods and cross-validation to identify the optimal values. Finally, the model hyperparameters were set as detailed in Table 3. The data from four subsets, FD001—FD004, were then used to test and evaluate the model's performance.

Table 3 Optimal parameters of the proposed framework

Parameter	Value
Max_epochs	100
Batch_size	512
Kernel size	[1, 5, 7]
L2 coefficient	0.006
R_{early}	125
Embed_size	512
Heads number	4
Learning rate	0.000 1
Waiting epochs	10

3.2 Learned attention weight distribution

To illustrate the attention mechanism in RUL prediction, Fig.8 provides an example using the life prediction of the 80th engine from the FD001 test set. The attention weights learned by the MSCLA model are visualized here. With a sliding window size of 30 for the FD001 subset, the resulting weight matrix is '30×30', representing the contribution of features at each time step to the final RUL prediction. Darker colors in the matrix indicate a greater contribution to the prediction.

In the traditional LSTM, only the features from the last time step are utilized for classification or regression tasks. In contrast, this work incorporates an attention mechanism following the LSTM layer to adaptively assign weights based on the relevance of deep features to the degradation process. This approach addresses the challenge of capturing long-term dependencies in time series data more effectively, leading to more accurate RUL predictions.

Figs.8(a—d) illustrates the attention weights extracted by the four attention 'heads'. Although the numerical ranges for each 'head' are normalized and appear similar, a closer inspection reveals that most regions are lighter while only a few regions are darker. This variation indicates that each attention 'head' focuses on different aspects of the data, allowing the model to capture diverse features from various segments of the input. By adaptively assigning weights, the model effectively prioritizes fea-

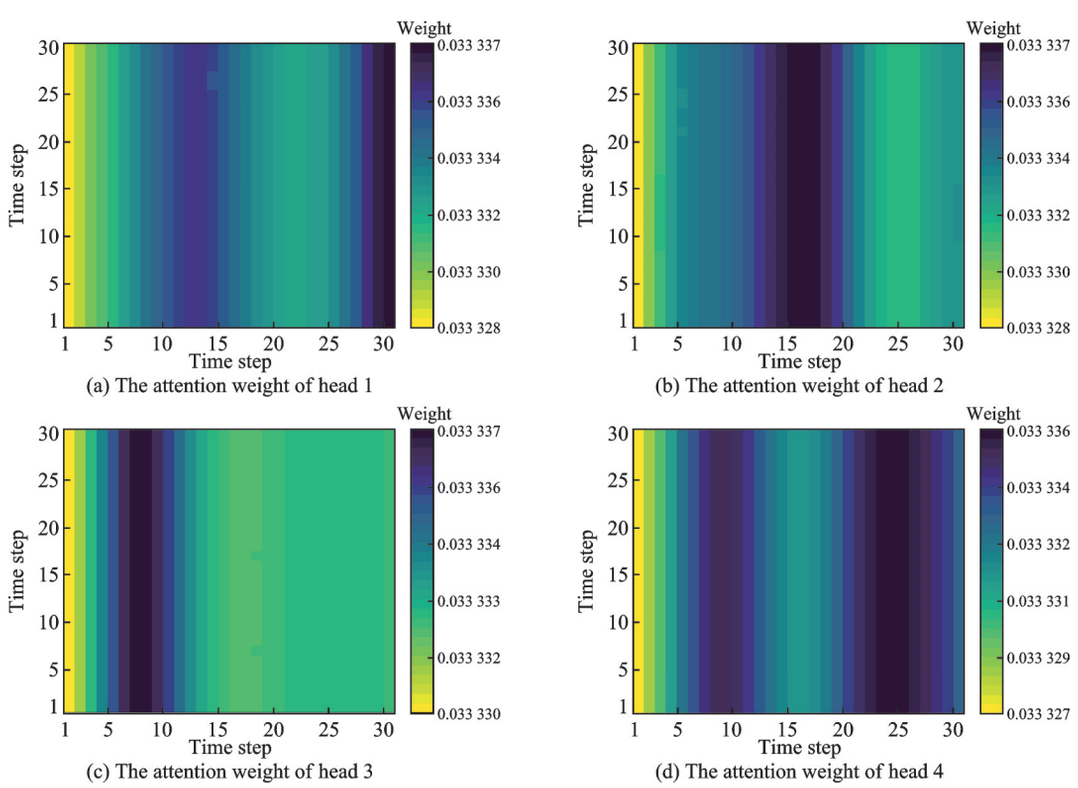

Fig.8 Learned attention weight of MSCLA

tures that are more relevant to the degradation process. For instance, Fig.8(d) shows that the weights are significantly higher near time steps 7—11 and 22—27, highlighting that these specific features have a greater impact on the degradation of the aero-engines.

3.3 Comprehensive prediction performance analysis

To evaluate the model's performance in aero-engine prediction tasks, we compared it with other SOTA methods using the C-MAPSS dataset. The results, presented in Table 4 and Table 5, show the RMSE and Score, respectively, where bold indicates optimal results, while underlined indicates sub-optimal results.

Table 4 RMSE comparison of different methods

Method	FD001	FD002	FD003	FD003
CNN[3]	18.45	30.29	19.82	29.16
DCNN[4]	12.61	22.36	12.64	23.31
HDNN[11]	13.02	15.24	12.22	18.16
MHNN[16]	—	18.88	—	21.32
NLSTM-Attn[17]	12.53	20.51	12.15	22.36
BiGRU-TSAM[18]	12.56	18.94	12.45	20.47
MSTformer[21]	—	**14.48**	—	15.03
MS-DCNN[26]	<u>11.44</u>	19.35	<u>11.67</u>	22.22
MSFMTP[29]	13.24	14.83	**11.17**	<u>14.09</u>
MSCLA (This work)	**10.82**	<u>14.75</u>	12.71	**14.07**

Table 5 Score comparison of different methods

Method	FD001	FD002	FD003	FD003
CNN[3]	1 287	13 570	1 596	7 886
DCNN[4]	273	10 412	284	12 466
HDNN[11]	245	1 282	288	1 527
MHNN[16]	—	1 308	—	2 225
NLSTM-Attn[17]	265	<u>1 195</u>	284	2 692
BiGRU-TSAM[18]	213	2 264	**233**	3 610
MSTformer[21]	—	**1 099**	—	<u>1 012</u>
MS-DCNN[26]	<u>196</u>	3 747	<u>241</u>	4 844
MSFMTP[29]	34 051	131 190	32 817	254 290
MSCLA (This work)	**162**	1 322	896	**918**

As shown in Tables 4 and 5, our model achieves the best results on the FD001 and FD004 subsets and is nearly optimal on the FD002 and FD003 subsets. Compared to the methods in Refs.[3-4], which used only CNN, our model reduces RMSE by 14.19%, 34.03%, and 39.64%, on the FD001, FD002, and FD004 subsets, respectively, and decreases the Score by 40.66%, 87.3%, and 92.64%. When compared to Ref.[11], which combined CNN and LSTM, our model reduces RMSE by 16.9% and 22.52% on the FD001 and FD004 subsets and lowers the Score by 33.88% and 39.88%. Compared to Refs.[16-18, 21], which used LSTM/GRU and attention mechanism, our model decreases RMSE by 13.65% and 6.39% on the FD001 and FD004 subsets and reduces the Score by 23.94% and 9.29%. Against Refs.[26, 29], which employed CNN for multi-scale feature extraction, our model lowers RMSE by 5.42%, 0.54%, and 0.14% on the FD001, FD002, and FD004 subsets, and reduces the Score by 17.08%, 64.72%, and 81.05%, respectively. These results demonstrate the successful integration of the advantages of CNN, LSTM, and attention mechanisms in our study, validating the effectiveness of MSCLA and showing that combining multiple networks achieves superior performance compared to using a single network alone.

To further observe and analyze the results produced by the model, the engines have been sorted in ascending order based on the actual RUL. The predicted life spans for these engines are then compared. The results are illustrated in Figs.9(a—d) showing the RUL prediction results for the 100, 259, 100, and 248 engines from the FD001, FD002, FD003, and FD004 test sets, respectively. The results demonstrate that the predicted degradation trends closely align with the actual degradation patterns, further validating the effectiveness of our model in accurately predicting overall degradation trends.

Finally, an engine was randomly selected from the FD001—FD004 test set, and its predicted RUL was compared with the actual RUL as well as with predictions from DCNN[4], HDNN[11], NLSTM-Attn[17], MS-DCNN[26], and MSFMTP[29] methods, as shown in Fig.10. The predicted RUL closely follows the actual RUL trajectory, reflecting superior performance in capturing local degrada-

tion trends. Notably, the MSCLA model provides precise predictions even for the more complex degradation trends observed in the FD002 and FD004 subsets, showcasing the effective integration of multi-scale feature extraction and attention mechanisms.

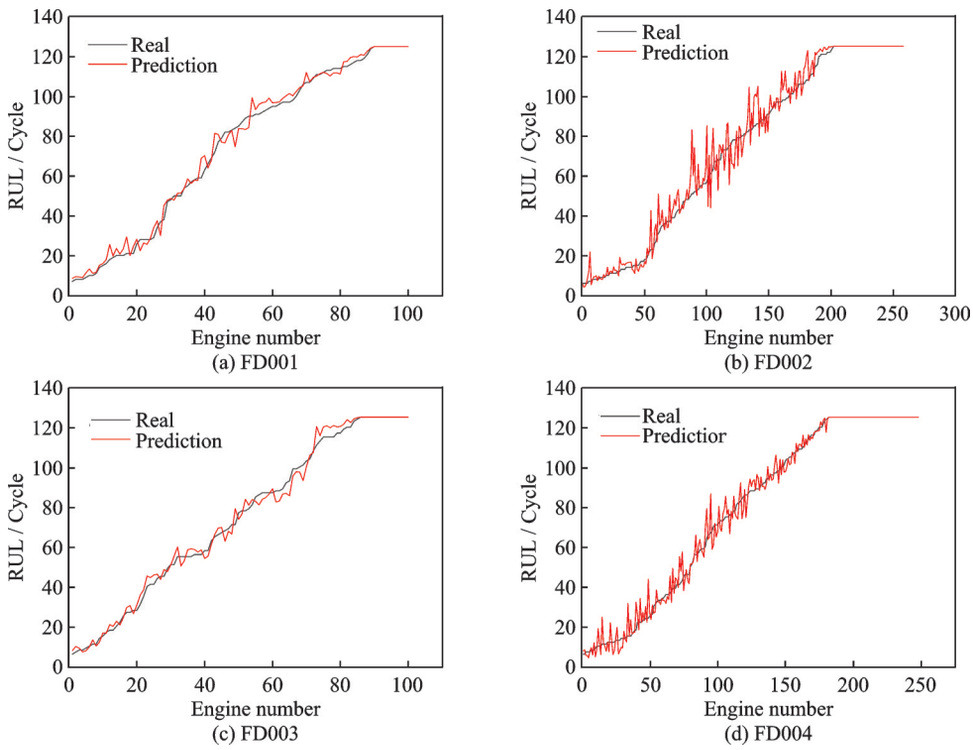

Fig.9 RUL prediction results for all engines (FD001—FD004)

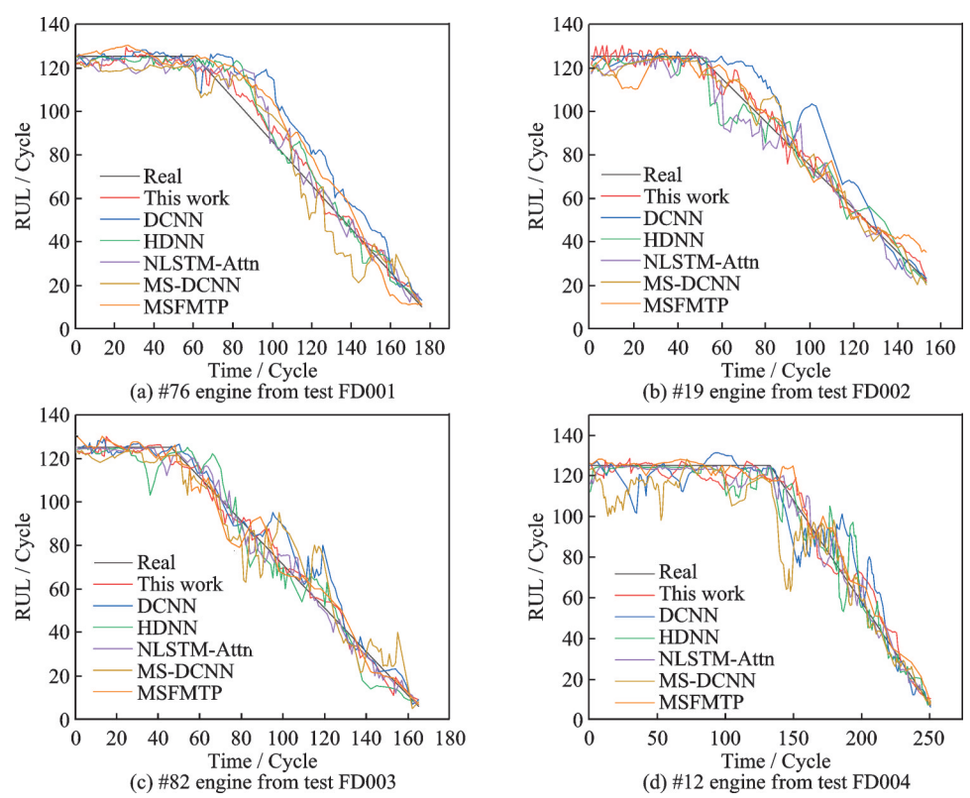

Fig.10 RUL prediction for a single engine (FD001—FD004)

4 Conclusions

In this paper, we propose a deep learning model combining multi-scale CNN and LSTM frameworks with the attention mechanism for RUL prediction of aero-engines. The degradation information at different scales is first extracted and integrated using a two-layer multi-scale feature extraction module. These integrated degradation features are then processed separately by the self-attention module and the three-layer LSTM network. The self-attention model focuses on identifying and emphasizing the most relevant features within the data, while the LSTM learns long-term dependencies in the temporal data during the degradation process. Finally, the degradation patterns learned by the self-attention module and the LSTM are combined and processed through the attention module. This approach enhances the capture of comprehensive time-dependent features, enabling more accurate RUL prediction.

Since batch size significantly influences model performance, we investigated how varying batch sizes affect our mode. We then identified the optimal convolution kernel combination by evaluating different kernel sizes. By visualizing the attention weights for each head, we can assess the impact of each time step on the final RUL prediction. When comparing our model to SOTA methods, we found that the MSCLA model demonstrates superior prediction accuracy on the FD001 and FD004 test sets, and its performance remains competitive with current mainstream algorithms.

In the future, dynamic values could be assigned to features at different scales to enhance the network's feature extraction capability. Additionally, leveraging the attention mechanism opens the possibility of further integrating Transformers with LSTM to improve long-term dependency learning.

References

[1] ORSAGH R F, SHELDON J, KLENKE C J. Prognostics/diagnostics for gas turbine engine bearings [M]. Atlanta, USA: American Society of Mechanical Engineers, 2003.

[2] CHIACHIO J, CHIACHIO M, SANKARARAMAN S, et al. Condition-based prediction of time-dependent reliability in composites[J]. Reliability Engineering & System Safety, 2015, 142: 134-147.

[3] SATEESH B G, ZHAO Peilin, LI Xiaoli. Deep convolutional neural network based regression approach for estimation of remaining useful life[C]//Proceedings of 21st International Conference. Database Systems for Advanced Applications. Dallas, USA: Springer International Publishing, 2016: 214-228.

[4] LI X, DING Q, SUN J Q. Remaining useful life estimation in prognostics using deep convolution neural networks[J]. Reliability Engineering & System Safety, 2018, 172: 1-11.

[5] YUAN M, WU Y T, LIN L. Fault diagnosis and remaining useful life estimation of aero engine using LSTM neural network [C]//Proceedings of International Conference on Aircraft Utility Systems. Beijing: IEEE, 2016, 135-140.

[6] XIANG S, QIN Y, LUO J. Multi-cellular LSTM-based deep learning model for aero-engine remaining useful life prediction[J]. Reliability Engineering & System Safety, 2021, 216: 107927.

[7] LIU J, LEI F, PAN C. Prediction of remaining useful life of multi-stage aero-engine based on clustering and LSTM fusion[J]. Reliability Engineering & System Safety, 2021, 214: 107807.

[8] HINCHI A, TKIOUAT M. Rolling element bearing remaining useful life estimation based on a convolutional long-short-term memory network[J]. Procedia Computer Science, 2018, 127: 123-132.

[9] REN L, SUN Y Q, CUI J, et al. Bearing remaining useful life prediction based on deep autoencoder and deep neural networks [J]. Journal of Manufacturing Systems, 2018, 48: 71-77.

[10] ZHANG C, LIM P, QIN A K. Multiobjective deep belief networks ensemble for remaining useful life estimation in prognostics[J]. IEEE Transactions on Neural Networks and Learning Systems, 2016, 28(10): 2306-2318.

[11] AL-DULAIMI A, ZABIHI S, ASIF A. A multimodal and hybrid deep neural network model for remaining useful life estimation[J]. Computers in Industry, 2019, 108: 186-196.

[12] SONG Y, SHI G, CHEN L Y, et al. Remaining useful life prediction of turbofan engine using hybrid model based on autoencoder and bidirectional long short-term memory [J]. Journal of Shanghai Jiaotong University, 2018, 23: 85-94.

[13] CHEN Z H, WU M, ZHAO R, et al. Machine remaining useful life prediction via an attention-based deep learning approach[J]. IEEE Transactions on Industrial Electronics, 2020, 68(3): 2521-2531.

[14] XIA J, FENG Y W, TENG D, et al. Distance self-attention network method for remaining useful life estimation of aeroengine with parallel computing[J]. Reliability Engineering & System Safety, 2022, 225: 108636.

[15] LIU L, SONG X, ZHOU Z T. Aircraft engine remaining useful life estimation via a double attention-based data-driven architecture[J]. Reliability Engineering & System Safety, 2022, 221: 108330.

[16] LIU Z Y, LIU H, JIA W Q, et al. A multi-head neural network with unsymmetrical constraints for remaining useful life prediction[K]. Advanced Engineering Informatics, 2021, 50: 101396.

[17] CHEN B, GUO K, CHEN F. Residual service life prediction of aero engine based on residual NLSTM network and attention mechanism[J]. Journal of Aerodynamics, 2023, 38(5): 1176-1184.

[18] ZHANG J S, JIANG Y C, WU S M, et al. Prediction of remaining useful life based on bidirectional gated recurrent unit with temporal self-attention mechanism[J]. Reliability Engineering & System Safety, 2022, 221: 108297.

[19] SAXENA A, GOEBEL K, SIMON D. Damage propagation modeling for aircraft engine run-to-failure simulation[C]//Proceedings of 2008 International Conference on Prognostics and Health Management. Denver: IEEE, 2008: 1-9.

[20] FREDERICK D K, DECASTRO J A, LITT J S. User's guide for the commercial modular aero-propulsion system simulation (C-MAPSS)[M]. Cleveland: Army Research and Technology Labs, 2007.

[21] XU D, XIAO X Q, LIU J, et al. Spatio-temporal degradation modeling and remaining useful life prediction under multiple operating conditions based on attention mechanism and deep learning[J]. Reliability Engineering & System Safety, 2023, 229: 108886.

[22] ADNAN R M, KHOSRAVINIA P, KARIMI B. Prediction of hydraulics performance in drain envelopes using K-means based multi-variate adaptive regression spline[J]. Applied Soft Computing, 2021, 100: 107008.

[23] CHEN J L, JING H J, CHANG Y H, et al. Gated recurrent unit based recurrent neural network for remaining useful life prediction of nonlinear deterioration process[J]. Reliability Engineering & System Safety, 2019, 185: 372-382.

[24] CHE C C, WANG H W, NI X M, et al. Aeroengine residual life prediction based on 1D-CNN and Bi-LSTM[J]. Journal of Mechanical Engineering, 2021, 57: 304-312.

[25] AL-DULAIMI A, ZABIHI S, ASIF A. NBLSTM: Noisy and hybrid convolutional neural network and BLSTM-based deep architecture for remaining useful life estimation[J]. Journal of Computing and Information Science in Engineering, 2020, 20: 021012.

[26] LI H, ZHAO W, ZHANG Y X, et al. Remaining useful life prediction using multi-scale deep convolutional neural network[J]. Applied Soft Computing, 2020, 89: 106113.

[27] VASWANI A, SHAZIER N, PARMAR N. Attention is all you need[J]. Advances in Neural Information Processing Systems, 2017, 30: 6000-6010.

[28] CHO K, MERRIËNBOER B, BAHDANAU D, et al. On the properties of neural machine translation: Encoder-decoder approaches[C]//Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation. Doha, Qatar: Association for Computational Linguistics,2014: 103-111.

[29] ZHOU L, WANG H W, XU S S. Aero-engine prognosis strategy based on multi-scale feature fusion and multi-task parallel learning[J]. Reliability Engineering & System Safety, 2023, 234: 109182.

[30] LI H, ZHAO W, Zhang Y X, et al. Remaining useful life prediction using multi-scale deep convolutional neural network[J]. Applied Soft Computing, 2020, 89: 106113.

Acknowledgements This work was supported by the National Key Research and Development Program of China (2023YFB4302403), and the Research and Practical Innovation Program of NUAA (xcxjh20230735).

Authors

The first author Mr. **DUAN Jiajun** received his Master's degree in traffic transportation from Nanjing University of Aeronautics and Astronautics. His research interests focus on the remaining useful life prediction of aeroengines.

The corresponding author Prof. **LU Zhong** received the B.E., M.S., and Ph.D. degrees from Nanjing University of Aeronautics and Astronautics. His research direction is aircraft system safety assessment and system reliability engineering.

Author contribution Mr. **DUAN Jiajun** carried out the literature review, performed the technique analysis and drafted the initial manuscript. Prof. **LU Zhong** designed the study, contributed to the methodology development. Mr. **DU Zhiqiang** contributed to revising images and tables, as well as optimizing hyperparameters. All authors commented on the manuscript draft and approved the submission.

Competing interests The authors declare no competing interests.

(Production Editor: LIU Yandong)

基于注意力机制的多尺度CNN与LSTM网络及在剩余使用寿命预测中的应用

段佳俊[1,2]，陆 中[1]，杜志强[1]

（1.南京航空航天大学民航学院，南京 211106，中国；
2.四川航天中天动力装备有限责任公司，成都 610100，中国）

摘要：当前航空发动机寿命预测领域通常聚焦于单尺度退化特征，现有方法未能充分捕捉时间序列数据中的内在联系。为解决此问题，本文提出一种基于注意力机制的新型剩余使用寿命（Remaining useful life，RUL）预测方法。本文设计了两层多尺度特征提取模块，以整合不同尺度的退化特征，这些特征随后通过并行的自注意力模块与3层长短期记忆（Long short-term memory，LSTM）网络进行处理，共同捕捉长期依赖关系并自适应关键权重特征。最后将两部分的退化模式集成至注意力模块，显著增强了模型捕捉长期依赖关系的能力。通过可视化注意力模块的权重矩阵，进一步提升了模型可解释性。在C-MAPSS数据集上的实验结果表明，本方法优于现有最先进的方法。

关键词：注意力机制；卷积神经网络；长短期记忆网络；多尺度特征提取