

Full length article

A novel remaining useful life prediction method under multiple operating conditions based on attention mechanism and deep learning

Jie Wang^a, Zhong Lu^{a,*}, Jia Zhou^b, Kai-Uwe Schröder^c, Xihui Liang^d^a College of Civil Aviation, Nanjing University of Aeronautics and Astronautics, 29 Jiangjun Road, Nanjing 211106, China^b Department of Production & Technology, Eastern Airlines Technic Co., Ltd. Jiangsu Branch, Nanjing 211106, China^c Institute of Structural Mechanics and Lightweight Design, RWTH Aachen University, Wüllnerstrasse 7, 52062 Aachen, Germany^d Department of Mechanical Engineering, University of Manitoba, Winnipeg R3T5V6, Canada

ARTICLE INFO

ABSTRACT

Keywords:

Remaining useful life prediction
Multiple operating conditions
Efficient channel attention (ECA)
Pyramid temporal self-attention
Bidirectional temporal convolution network
Informer

Remaining useful life (RUL) prediction is a key technique for supporting predictive maintenance. Accurate RUL prediction plays an important role in maintenance decisions. However, RUL prediction has two challenges: first, it is difficult to capture long-term dependencies effectively; second, the accuracy and efficiency are not satisfied under multiple operating conditions. A novel RUL prediction model that integrates bidirectional temporal convolution and improved Informer (ABITCI) is proposed with consideration of multiple operating conditions. First, the bidirectional temporal convolution network (BiTCN) is designed with efficient channel attention (ECA). The degradation features from different channels can be extracted by weighting feature contributions. Second, the Informer with sparse pyramid temporal self-attention is designed to capture degradation information from different time steps. Finally, the effectiveness of the proposed method is verified by different datasets of aircraft engines. Compared with the present methods, the results show that the root mean square errors (RMSEs) have been reduced by 20.84 %–50.38 %, 16.29 %–41.49 %, and 36.96 %–59.53 % on the CMAPSS-FD002, CMAPSS-FD004, and NCMAPSS datasets, respectively. It demonstrates that the ABITCI model performs well for RUL prediction under multiple operating conditions.

1. Introduction

With the aim of predicting the future potential state of equipment and reducing operating risks, predictive maintenance has become a hot topic in smart manufacturing. RUL prediction is a key technique for supporting predictive maintenance. The remaining life of equipment is obtained by mining useful degradation information from monitoring data [1]. Accurate RUL results can help to make rational maintenance plans and decisions [2].

In recent years, emerging deep learning (DL) methods have received extensive attention for RUL prediction [3–5]. DL methods allow the automatic extraction of degradation features from massive amounts of monitoring data and realize the representation of low-dimensional to high-dimensional data. Typical DL methods include recurrent neural networks (RNNs) [6,7], long short-term memory (LSTM) [8,9], gate recurrent unit (GRU) [10,11], convolutional neural networks (CNNs) [12,13], temporal convolutional networks (TCNs) [14,15]. Ansari et al. [16] proposed a hybrid RUL prediction model for lithium-ion batteries

that integrates RNN and particle swarm optimization. Ribeiro et al. [17] proposed an RNN method based on statistical recurrent units for RUL prediction. Sayah et al. [18] designed a deep LSTM model to improve the accuracy of RUL prediction. Izadi et al. [19] developed a WaveNet-GRU model to predict the RUL of PEM fuel cells. Guo et al. [20] proposed a GRU-based method to predict the RUL of lithium batteries. Jou et al. [21] built an encode – decoder for RUL prediction by using LSTM and CNN, which improved the prediction performance. In engineering, varying operating conditions pose a challenge for accurate RUL prediction. Considering RUL prediction under multiple operating conditions has become a popular topic [9,13,22–25]. Peng et al. [3] adopted a graph neural network to predict the RUL of machinery under variable conditions. Pater et al. [5] designed an LSTM autoencoder to predict the RUL of engines. Xu et al. [25] proposed a temporal partial domain adaptation network for RUL prediction under multiple conditions.

However, traditional RNNs, CNNs, and their variants have the following deficiencies:

* Corresponding author.

E-mail address: luzhong@nuaa.edu.cn (Z. Lu).

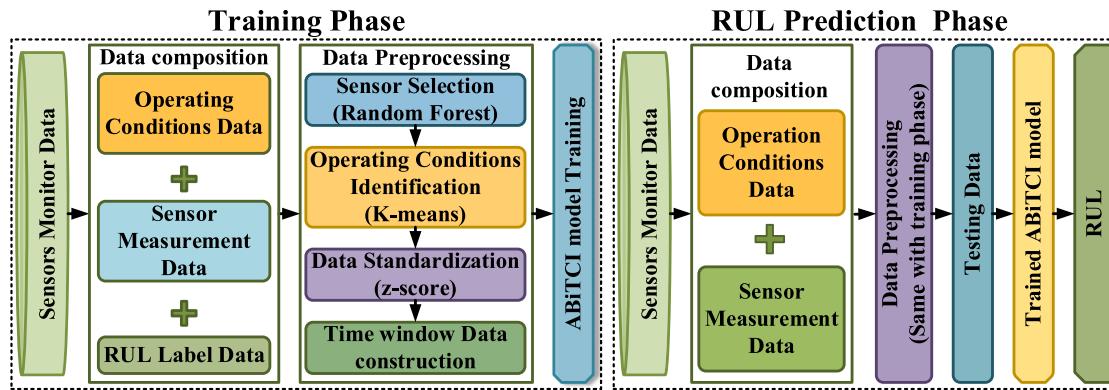


Fig. 1. RUL prediction scheme based on the ABiTCI model.

- (1) RNN-based methods are limited by their recurrent structure and easily cause long-term dependency issues. In other words, it is difficult to extract enough degradation information with the fewest time steps.
- (2) CNN-based methods are limited by the convolutional kernel and it is difficult to capture long-term dependencies.
- (3) The aforementioned methods extract degradation features, whereas the different features are not weighted, which means that the features of different channels and time steps are equally important.
- (4) Most DL-based models enhance their feature learning capability by stacking network layers, which easily results in the model being highly complex and exhibiting serious sequence dependency.

An attention mechanism is a good choice for evaluating feature importance. It is used to calculate and output the weights of different features. The attention mechanism provides highly weighted important information and suppresses useless information [15]. Thus, DL methods that fuse attention are becoming popular in RUL prediction [21,22,23]. Common methods include LSTM with attention [26,27], GRU with attention [28,29], CNNs with attention [15,30], TCNs with attention [31,32], and the Transformer-based model [33,34]. Boujamza et al. [33] embedded attention into LSTM to estimate the RUL of aircraft engines and verified the effectiveness of the attention mechanism. Deng et al. [35] constructed a GRU model that fused attention for RUL prediction. Du et al. [36] designed a CNN model with global attention to predict the RUL of bearing. Lin et al. [37] provided a dual attention framework, called the channel attention & temporal attention temporal convolutional network, to estimate aircraft engine RUL. These studies

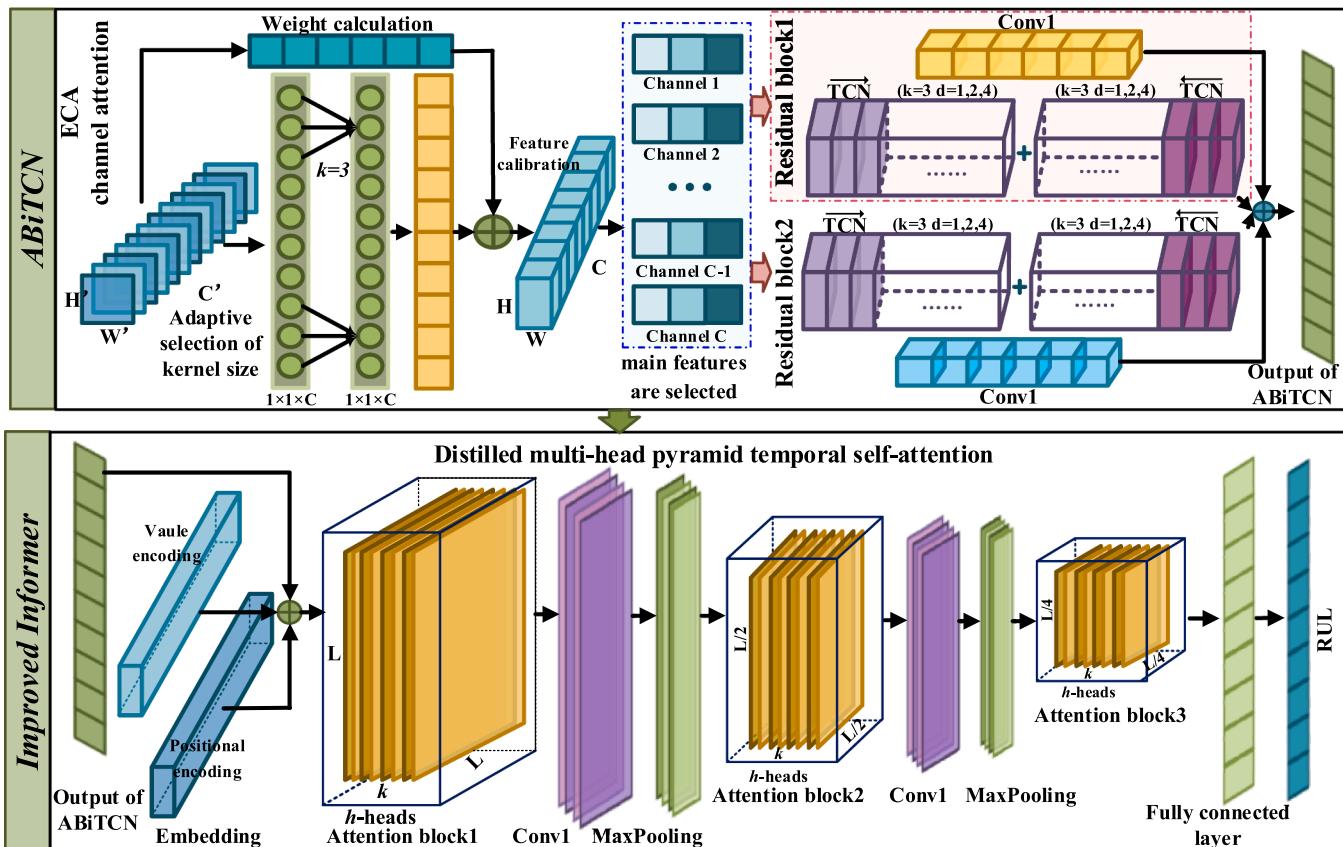


Fig. 2. Proposed RUL prediction model (ABiTCI).

demonstrate the role of attention mechanisms in RUL prediction. However, these methods do not consider attention complexity, and it is difficult to balance the relationship between performance and complexity.

To address the above problems, a novel integrated dual attention bidirectional temporal convolutional improved Informer model (ABiTCl) is proposed for RUL prediction under multiple operating conditions. This method integrates the BiTCN based on efficient channel attention and the Informer based on pyramid temporal self-attention. First, ECA is used to weigh the multi-channel information and calculate the feature weights. Second, the BiTCN is designed to extract information from different channels. Moreover, the residual connection is adopted to reduce the model depth. Third, pyramid temporal self-attention is utilized to mine feature contributions with different time steps and obtain the weighted features. Finally, the Informer encoder is designed to capture the long and short-term information, and the fully connected layer is applied to output the RUL values. The superiority of the proposed method is verified on two engine datasets with multiple operating conditions.

The main contributions of this study are as follows.

- (1) A novel RUL prediction method (ABiTCl) is developed to address RUL prediction under multiple operating conditions. Various conditions in building prediction models are considered to mitigate the impact of different conditions on RUL accuracy.
- (2) The BiTCN based on efficient channel attention, is designed to extract long-term degradation features of nonadjacent data. It adaptively assigns weights to features of different channels and enhances the attention of correlation features.
- (3) The Informer based on pyramid self-attention, is designed to extract temporal features in parallel. A masking strategy is used to highlight the important features related to degradation at time steps and reduce the complexity and time dependence of the model.

The remainder of this study is organized as follows. The proposed RUL prediction method under multiple operating conditions is detailed in [Section 2](#), which includes the prediction scheme and the model framework. The experimental studies and discussions are provided in [Section 3](#), in which the performance of the proposed method is verified on two common datasets. Finally, conclusions are drawn in [Section 4](#).

2. An RUL prediction method based on attention and the BiTCN & Informer

Our proposed RUL prediction method under multiple operating conditions is described in this section. A scheme for RUL prediction based on the ABiTCl is shown in [Fig. 1](#). The framework of the proposed model is depicted in [Fig. 2](#).

2.1. RUL prediction scheme based on the ABiTCl model

The proposed RUL prediction scheme based on the ABiTCl is shown in [Fig. 1](#). It has two phases: model training and RUL prediction.

The training phase procedure is as follows:

- i) **Operation-failure data collection.** These data include sensor measurement signals, operating condition information, and RUL labels.
- ii) **Data preprocessing.** It includes various sensor selection, operating condition identification, data standardization, and time window construction.
- iii) **ABiTCl model training.** This model includes two parts: a BiTCN with ECA, and an Informer encoder with pyramid temporal self-attention. The Adam optimizer is adopted to update the network parameters.

The prediction phase process is as follows:

- i) **Testing data collection.** It includes the monitoring information of various sensors from the test equipment.
- ii) **Test data preprocessing.** This process follows the same method used for the training data.
- iii) **RUL prediction.** The processed data fed into the trained ABiTCl model, and the predicted RUL values are output.

In our prediction scheme, two main issues, the preprocessing of data and the design of the ABiTCl model, are described in the following sections.

2.2. Data preprocessing

The preprocessing process is shown in [Fig. 1](#). It consists of four parts: sensor selection via the Random Forest (RF) algorithm, operating condition identification via the K-means algorithm, data standardization, and time window construction.

i) Sensors selection via Random Forest

Most of the sensors are irregular or do not affect degradation. Thus, to reduce the computational cost of the DL model, the RF algorithm is applied in this work to filter sensors with high contributions [38].

ii) Operating condition identification via K-means

Multiple operating condition information can contradict the premise that samples are extracted from independent identical distributions with DL methods [4]. To eliminate the condition discrepancies and improve the prediction effects, an unsupervised K-means clustering algorithm is adopted for identifying operating conditions. Specifically, the selected signals are divided into given K clusters.

iii) Data standardization

Z-score standardization is used to unify the data scales, as shown in Eq. (1)

$$x_{\text{stand}}^{i,k} = \frac{x^{i,k} - \mu^k}{\sigma^k} \quad (1)$$

where $x^{i,k}$ is the i th measurement of the k th cluster. μ^k is the mean of the k th cluster, $\sigma^k = \sqrt{\sum (x^{i,k} - \mu^k)^2 / n}$ denotes the standard deviation, and n is the size of the samples.

iv) Time window construction

The sensor of a single time point ignores contextual information, which is unfavourable for the capture of temporal features. Therefore, a fixed-size time window is utilized to construct continuous window data.

2.3. BiTCN with efficient channel attention

The conventional TCN has two shortcomings: the input features from all the channels are considered equally important for RUL prediction, and the unidirectional structure neglects the features before the sensor. The BiTCN with the ECA module (ABiTCl) is designed to collect vital channel features and suppress useless ones. The framework of the module is given in [Fig. 2](#). The details are given below.

2.3.1. Efficient channel attention

ECA is a local cross-channel interaction strategy without dimensionality reduction, which is realized via 1D convolution [39]. In this work, the main features are selected as inputs of the BiTCN on the basis

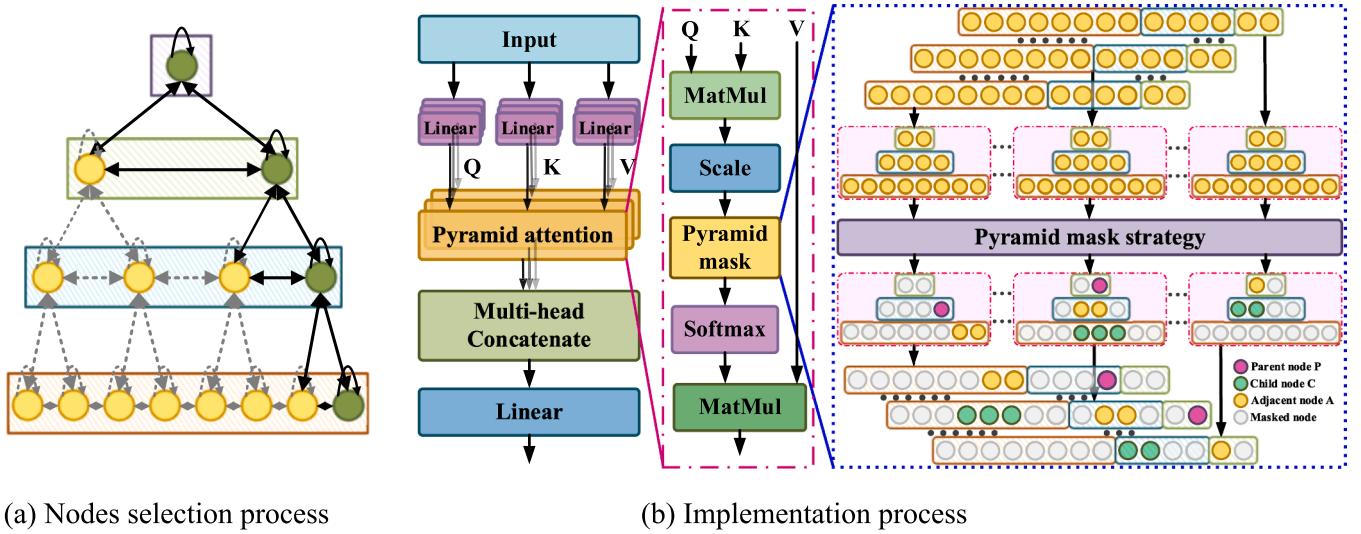


Fig. 3. Scheme of pyramid self-attention.

of the importance of multi-channel information. The channel attention is learned by the band matrix \mathbf{W}_k , which is expressed as

$$\mathbf{W}_k = \begin{bmatrix} w^{1,1} & \dots & w^{1,k} & 0 & 0 & \dots & \dots & 0 \\ 0 & w^{2,2} & \dots & w^{2,k+1} & 0 & \dots & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & 0 & \dots & w^{C,C-k+1} & \dots & w^{C,C} \end{bmatrix} \quad (2)$$

where \mathbf{W}_k involves $k \times C$ parameters.

As for Eq. (2), the weights of x_i are obtained by sharing the learning parameters with its k adjacent channels. Namely, the ECA is expressed as

$$x_{out} = J\left(\sum_{j=1}^k w^j x_i^j\right), x_i^j \in \Omega_i^k \quad (3)$$

where $J(\cdot)$ denotes the activation function. x_i^j is the channel feature, w^j is the weight of x_i^j , and Ω_i^k denotes the set of k adjacent channels of x_i .

ECA is implemented by a 1D convolution with kernel size of k , which is expressed as

$$x_{out} = J(W_k x_{in}) = J(\text{Conv1}_k(x_{in})) \quad (4)$$

where Conv1_k denotes the 1D convolution. $x_{in} \in \mathbb{R}^C$ represents the multi-channel features matrix.

2.3.2. Bidirectional temporal convolution network

TCN is implemented by the dilation convolution. The dilation convolution kernel k' and the reception L are denoted as follows

$$k' = d \cdot (k - 1) + 1 \quad (5)$$

$$L = \left(\sum_D d \cdot (k - 1) \right) + 1 \quad (6)$$

where k is the kernel size, and d is the dilation factor. D is the dilation rate array $\{d_1, d_2, \dots, d_n\}$, n is the number of dilated convolution layers.

The BiTCN block is designed in this work to maximize the before-and-after information of the sensors. It includes information compression and feature adaptive adjustment, which are described as follows

$$X_s = \vec{\text{TCN}}(\vec{X_f}, k', L, d) + \vec{\text{TCN}}(\vec{X_f}, k', L, d) \quad (7)$$

$$X_{\text{cali}} = X_s \otimes \mathbf{W} \quad (8)$$

where TCN^\rightarrow and $\vec{\text{TCN}}$ denote the forwards and backwards feature extraction processes, respectively. $\vec{X_f} = [x_1, x_2, \dots, x_t]$ is the forwards input feature, and $\vec{X_f} = [x_t, x_{t-1}, \dots, x_1]$ is the backwards input feature. \mathbf{W} is adaptive weight matrix.

In this work, ECA is used to calculate feature weights and calibrate the fusion features of the BiTCN. ECA focuses attention on correlation features, which reduces the computational effort and improves the efficiency of the model. The feature patterns of the BiTCN are adjusted to fit the Informer input.

2.3.3. Residual connection

The residual connections are designed to prevent information loss and improve the performance of the BiTCN. The outputs of the BiTCN are expressed as follows

$$X_r = X_{\text{cali}} \oplus J\left(X_{\text{cali}} + \vec{\text{TCN}}(X) + \vec{\text{TCN}}(X)\right) \quad (9)$$

Moreover, max-pooling is used at the end of the model to reduce the dimensions of the features.

2.4. Informer with distilled pyramid temporal self-attention

Although the ABiTTCN extracts as much information as possible about sensors, it ignores the information about the time steps. The Informer encoder with distilled multi-head pyramid temporal self-attention is designed to accept features of the ABiTTCN and capture features from important time steps. The improved Informer is shown in Fig. 2, it includes embedding and distilled multi-head pyramid temporal self-attention

2.4.1. Embedding

The classical embedding is position embedding, which preserves the information via fixed positions. The combination of value and position encoding is used in this study to maximize degradation information. The formula is expressed as follows

$$\begin{aligned} \mathbf{X}_{en[i]}^t &= \alpha u_i^t + \text{PE}(L_x(t-1) + i) \\ &= \begin{cases} \alpha u_i^t + \text{PE}(t, 2n) = \alpha u_i^t + \sin[t/(2L_x)^{2n/d \text{ mod } el}] \\ \alpha u_i^t + \text{PE}(t, 2n+1) = \alpha u_i^t + \cos[t/(2L_x)^{2n/d \text{ mod } el}] \end{cases} \end{aligned} \quad (10)$$

where α is the balance factor for feature alignment, u_i^t is a m -dim vector

with 1D convolution. $\text{PE}(\cdot)$ denotes the position encoding, which is generally obtained using the sine cosine function. L_x is the length of the input feature, $i \in \{1, \dots, L_x\}$. t is the position of the feature in the sequence. $n \in \{1, \dots, d_{\text{mod el}}/2\}$ is a dimension of the feature embedding, $d_{\text{mod el}}$ is a dimension of the feature embedding projection.

Notably, the features are aligned via a zero padding strategy to ensure feature consistency. The input of the Informer is obtained by combining the outputs of the ABiTAN, value encoding, and position encoding.

2.4.2. Distilled multi-head pyramid temporal self-attention

Pyramid temporal self-attention is an effective tool for solving issues with high complexity. It is a sparse attention mechanism. Unlike full attention, pyramid self-attention selects high-correlation nodes to calculate important features, corresponding to the pyramidal graph in Fig. 3(a). For RUL prediction, the key to achieving pyramid self-attention depends on how to mask the low-correlation nodes on the time steps. Therefore, in this study, the masking strategy is designed, and the implementation process is shown in Fig. 3(b).

Pyramid self-attention is calculated as follows. All correlation matrices on the time step are calculated to obtain the full attention (i.e., calculate all the $Q - K$ pairs), which is represented as follows

$$A(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V \quad (11)$$

where Q is the query matrix, K is the key matrix, and V is the value matrix obtained via linear processing. d is the dimension of K , and T denotes the matrix transpose operation.

This study selects high-correlation nodes according to the relative position of the present node to others. The closer the relative position is, the higher the correlation. Specifically, assume that the i th node on time scale o is denoted $n_i^{(o)}$, where $o = 1, 2, \dots, O$ denotes a bottom-to-top sequential scale. Each node can focus on a set of neighbouring nodes $N_i^{(o)}$ at three scales: the adjacent A nodes that include itself at the same scale (represented as $A_i^{(o)}$), the C child nodes at the previous scale (represented as $C_i^{(o)}$), and the P parent nodes at the next scale (represented as $P_i^{(o)}$). The node selection process is represented as follows

$$\begin{cases} N_i^{(o)} = A_i^{(o)} \cup C_i^{(o)} \cup P_i^{(o)} \\ A_i^{(o)} = \left\{ n_t^{(o)} : |t - i| \leq \frac{A - 1}{2}, 1 \leq t \leq \frac{L_x}{C^{o-1}} \right\} \\ C_i^{(o)} = \left\{ n_t^{(o-1)} : (i - 1)C < t \leq iC \right\} \quad \text{if } o \geq 2 \text{ else } \emptyset \\ P_i^{(o)} = \left\{ n_t^{(o+1)} : t = \lceil \frac{i}{C} \rceil \right\} \quad \text{if } o \leq O - 1 \text{ else } \emptyset \end{cases} \quad (12)$$

The masking function is designed to mask low-correlation nodes and it uses $-1e9$ to eliminate them. The formula is expressed as follows

$$\text{Mask_func}(n_i^{(o)}) = n_i^{(o)} \text{ if } n_i^{(o)} \in N_i^{(o)} \text{ else } -1e9 \quad (13)$$

The Softmax function is used to calculate the correlation matrix of the selected nodes (i.e., $Q - K$ pairs other than masking). The formula for pyramid self-attention is expressed as follows

$$A_{\text{mask}}(Q_m, K_m, V_m) = \text{Softmax}\left(\frac{Q_m K_m^T}{\sqrt{d}}\right)V_m \quad (14)$$

where Q_m, K_m, V_m represent the high-correlation matrices after the masking process.

The attention of multiple heads is used in this study to obtain useful information and enhance computational efficiency, which is represented as follows

$$A(Q_m, K_m, V_m) = \text{Concat}(h_1, h_2, \dots, h_h)W \quad (15)$$

Table 1
CMAPSS datasets.

Datasets	Conditions	Faults modes	Training trajectories	Testing trajectories
FD001	1	1	100	100
FD002	6	1	260	259
FD003	1	2	100	100
FD004	6	2	249	248

where $\text{Concat}(\cdot)$ denotes the connection operation. h_i denotes the i th attention head and W is the weight matrix.

The distillation mechanism is added at the end of each attention block to reduce the dimensions of the features. The distillation mechanism from the j th layer to the $(j + 1)$ th layer is expressed as follows

$$X_{j+1} = \text{Maxpool}[\text{ELU}(\text{Conv}1_k([X_j]_A))] \quad (16)$$

where $\text{Maxpool}[\cdot]$ is a pooling function. $\text{ELU}(\cdot)$ denotes the activation function, $[\cdot]_A$ denotes the attention block, and X_j is the feature matrix of the j th layer.

2.4.3. Informer encoder

The encoder is designed to extract features from time steps. First, the embedding layer is utilized to obtain the inputs of the encoder. Second, pyramid attention blocks are applied to capture the features with vital time steps and compress the feature dimensions. Finally, the fully connected layer is used to map the hidden features and output the RUL values. The fully connected layer in this study is essentially a linear mapping function, which is expressed as follows

$$RUL_t = J(W_p X^h + b_p) \quad (17)$$

where $X^h \in \mathbb{R}^{(L_x/2) \times d_{\text{mod el}}}$ denotes the output vector of encoder. W_p and b_p are the training parameters.

3. Experiment

The FD002 and FD004 subsets of CMAPSS (CMAPSS-FD002 and CMAPSS-FD004) and the NCMAPSS datasets are used in our experiments to evaluate the performance of the ABiTAN model. The datasets are obtained from the simulation of a commercial modular aerospace propulsion system provided by NASA. All the datasets contain the degradation information of several engines under multiple operating conditions. The RMSE and score are adopted as evaluation metrics of the model. All the experiments were performed in the same experimental environment, the specific environment was Python 3.7 with PyTorch 1.12.0 + CU113, and the Wins system with an Intel(R) Xeon(R) Gold 6226R CPU @ 2.90 GHz and 256 GB of RAM was used.

3.1. Case 1

The CMAPSS dataset consists of turbofan engine simulation model data provided by NASA, which contains four subsets, as detailed in Table 1. CMAPSS-FD002 and CMAPSS-FD004 which contain six operating conditions are adopted in this work to perform experiments for RUL prediction.

3.1.1. Data processing

i) Sensor selection

Although 21 sensors recorded degradation signals such as temperature, pressure, etc., not all the sensors affected engine degradation. Invalid sensors may reduce the accuracy of RUL prediction. The RF algorithm is used to calculate the contributions of sensors to RULs [38,40,41], as shown in Fig. 4. The 14 sensors with contributions greater

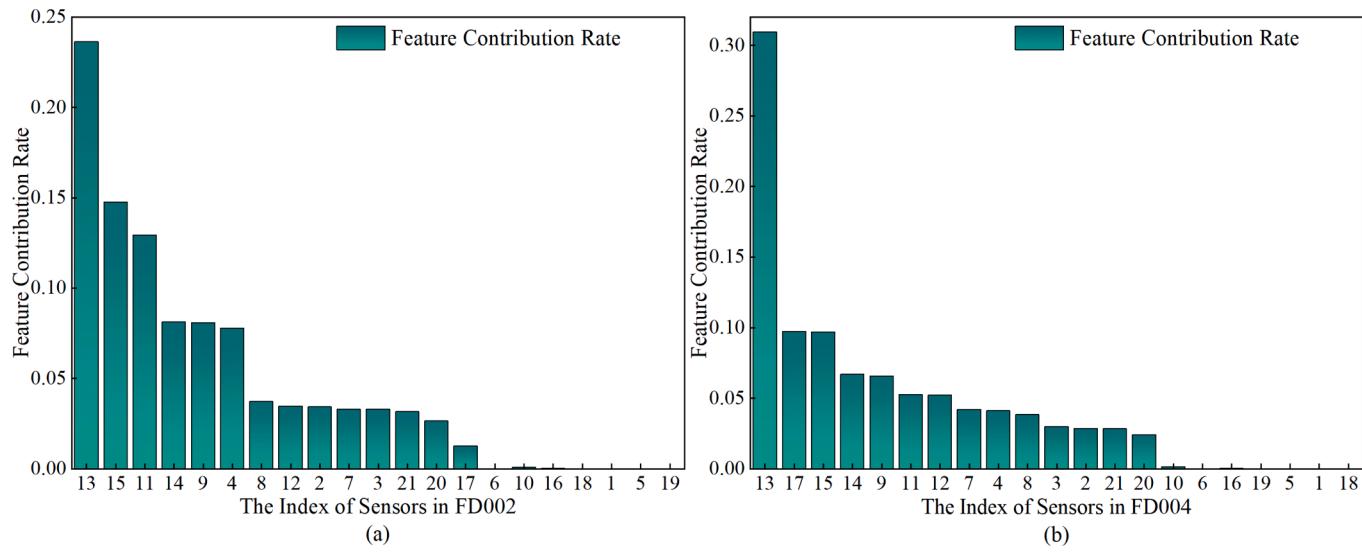


Fig. 4. Feature contributions of CMAPSS-FD002 and CMAPSS-FD004.

Table 2
Hyperparameters of the ABiTCI.

Hyperparameters	Description	Selected parameters
batch_size	The number of samples for each back propagation	256
head numbers	The number of heads for attention	4
kernel size	The convolution kernel size of the TCN	3,4
ABITCN blocks	The number of stacked ABITCN blocks	2
Encoder blocks	The number of stacked encoder blocks	2
d_model	The dimension of the attention mechanism	128

than zero are selected as inputs of the ABiTCI.

ii) Operating condition identification & Data standardization

Based on flight height, mach number, and throttle resolver angle condition parameters, the K-means algorithm is used to divide the 14 sensors into six clusters. This way reduces the disadvantageous influence of complex conditions for RUL prediction. The sensors are mapped to [0,1] by the z-score to unify the data scales. The appropriate time

window is used to encapsulate the sensors and establish the input features of the model. In addition, the mapping of window data to RUL labels is established.

iii) RUL labels

Considering that the engines do not degrade in the early stage, the labels of the RUL are established via a piecewise function [37]. Following previous studies, the maximum remaining useful life is set to 125 [4,15].

3.1.2. Hyperparameter analysis

The hyperparameters of the ABiTCI have a significant effect on the learning ability and prediction accuracy of the model. According to this study [4], the candidate values of hyperparameters are established. Based on the training data, the optimal parameters are determined via the grid search method, as shown in Table 2.

3.1.2.1. Impact of the time window size. Sliding window processing is essential for time-series data, where window size, as a key parameter directly affects RUL prediction. Hence, we investigated the effect of window size on model performance and determined the optimal

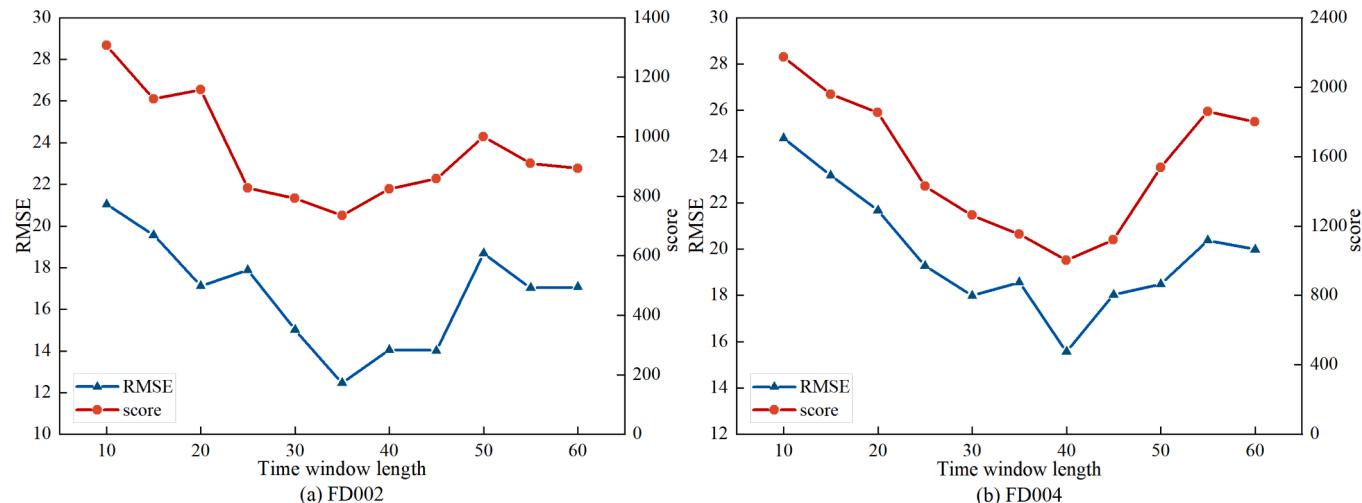


Fig. 5. Performance of the ABiTCI model with different time window sizes.

Table 3
Hyperparameter value settings.

Hyperparameters	C1	C2	C3	C4	C5	C6	Co
batch_size	16	32	64	128	256	512	256
head numbers	2	4	8	16	32	64	4
kernel size	1	2	3	4	5	6	3,4
ABiTCA blocks	1	2	3	4	5	6	2
Encoder blocks	1	2	3	4	5	6	2
d_model	32	64	128	256	512	1,024	128

configuration of the model on different datasets. The window size range is set from 10 to 60 for testing with other identical parameters. The evaluation metrics for model performance are the RMSE and score.

As shown in Fig. 5, the prediction performance of the model improves as the time window size increases. This shows that a larger window size contains more feature information. However, in CMAPSS-FD002 and CMAPSS-FD004, the prediction performance degrades as the window size exceeds 35 and 40, respectively. This may be due to model overfitting and learning much useless information. On the basis of the experimental results, the windows for CMAPSS-FD002 and CMAPSS-FD004 are 35 and 40, respectively. Moreover, a similar experiment is applied to the NCMAPSS DS02 dataset to determine the appropriate time windows.

3.1.2.2. Sensitivity of the hyperparameters. To evaluate the sensitivity of the ABiTCA to different hyperparameter value variations, we performed sensitivity testing on the CMAPSS-FD002 dataset. As shown in Table 3, 6 different values are set for 6 important hyperparameters, and the C1 to C6 marks are used to indicate the ascending order of the hyperparameters. Co denotes the optimal hyperparameter combination. The RMSE and score are adopted as evaluation metrics. A heatmap is applied to illustrate the effects of hyperparameters on model performance. As shown in Fig. 6, the lighter areas in the figure indicate better performance and the darker ones indicate poorer performance. Obviously, Co contains more light areas, which illustrates the good performance of our proposed model. The horizontal axis indicates the values of the hyperparameters, and the vertical axis indicates the hyperparameters.

Fig. 6 shows that ABiTCA blocks and encoder blocks have more shallow areas, which indicates that the ABiTCA is insensitive to these two hyperparameters. All the hyperparameters used for testing have fewer variations in the RMSE, which suggests that the results are quite stable. For the score, there are more shallow areas on the graph and obvious discrepancies between the highest and lowest scores, which shows that the model easily selects the optimal hyperparameter value. Overall, the

ABiTCA demonstrates stability and few fluctuations for the RMSE, and the optimal value of hyperparameters can be obtained easily in terms of score.

3.1.3. Results and discussions

3.1.3.1. Ablation study on the ABiTCA model. Two ablation experiments are performed as described in this section, which verify the validity of the attention mechanism, model structure, and operating conditions for RUL prediction.

i) Assessment of the role of the ABiTCA

To evaluate the validity of the attention mechanism and model structure, the ablation experiments include five ways as follows: BiTCN without ECA plus Transformer (NAT), BiTCN without ECA plus Informer (NAI), BiTCN with ECA plus Transformer (AT), BiTCN with ECA plus Informer (AI), and the ABiTCA model proposed in this study. Table 4 shows the results of the ablation studies.

The results show that the RMSEs of the models with ECA are lower than those without ECA, and the RMSEs of the models adopting pyramid self-attention are lower than those of other models. The above results indicate that the attention mechanism can capture important features on different channels and time steps to obtain accurate RULs. The models with the Informer have lower RMSEs than those with the Transformer do, which suggests that the Informer reduces the complexity of the model and enhances the model performance.

ii) Assessment of the effect of condition information

To evaluate the effects of operating conditions on model performance, we set up four scenarios. In case (i), only initial sensors are used. In case (ii), only standardized sensors are used. In case (iii), standardized

Table 4
Results of the ablation study (ABiTCA).

Methods	CMAPSS-FD002		CMAPSS-FD004	
	RMSE	score	RMSE	score
No ECA + Transformer (NAT)	19.42	2513	21.86	3885
No ECA + Informer (NAI)	18.11	1857	20.87	3179
ABiTCA + Transformer (AT)	16.38	1490	19.97	2491
ABiTCA + Informer (AI)	15.74	1101	18.60	2144
ABiTCA (This study)	12.46	736	15.57	1003

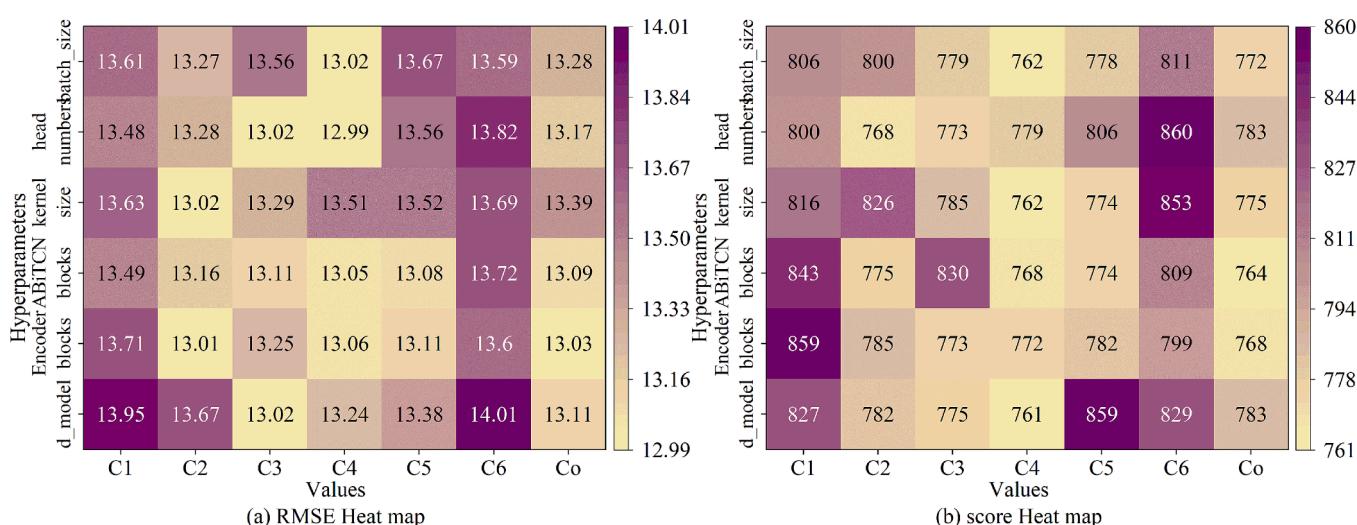


Fig. 6. Heatmap of hyperparameter influences.

Table 5
Results of ablation study (multiple conditions messages).

Cases	RMSE	
	FD002	FD004
Case i	20.59	22.64
Case ii	18.76	20.47
Case iii	15.29	18.27
Case iv (This study)	12.46	15.57

condition information and sensors are used without condition identification. In case (iv), the information after condition identification and sensor selection are used in this study. The experimental results are shown in Table 5. The RMSEs of the four scenarios (case i to case iv) decrease gradually, indicating that condition identification is necessary. The fusion of conditions is beneficial for improving model performance.

3.1.3.2. Experimental results. The RUL prediction results of the ABiTCI model are given in Fig. 7. Obviously, the prediction values and the true values fit closely with each other. This proves that the model can rationally exploit the operating condition data and automatically extract the full degradation features to obtain satisfactory prediction results.

The predicted results of test engine #185 on CMAPSS-FD002 and test engine #111 on CMAPSS-FD004 are displayed in Fig. 8. The RUL is close to 125 in the early stage, which means that the engines are in a healthy state. The RUL decreases linearly after degradation begins. As the number of cycles decreases, the predicted values gradually converge to the true values. This is because, as the service time increases, the degradation of the equipment becomes clear and the model easily captures the information related to the RUL.

3.1.3.3. Comparisons with other methods. Some state-of-the-art methods are compared to verify the superiority of the proposed ABiTCI method. The comparison results are shown in Table 6. The prediction results of our model rank first among the 11 models listed in Table 6. Compared with those of the other models, the RMSEs of our model are reduced by 20.84 %–50.38 % and 16.29 %–41.49 % on CMAPSS-FD002 and CMAPSS-FD004, respectively. The scores of our model are reduced by 33.15–84.64 % and 53.22 %–79.82 % on both datasets. These results clearly illustrate the superiority of the ABiTCI model.

3.1.3.4. Model efficiency analysis. To verify the applicability of the proposed model in practical applications, model efficiency is verified. Verification includes model training and test time calculations, as well

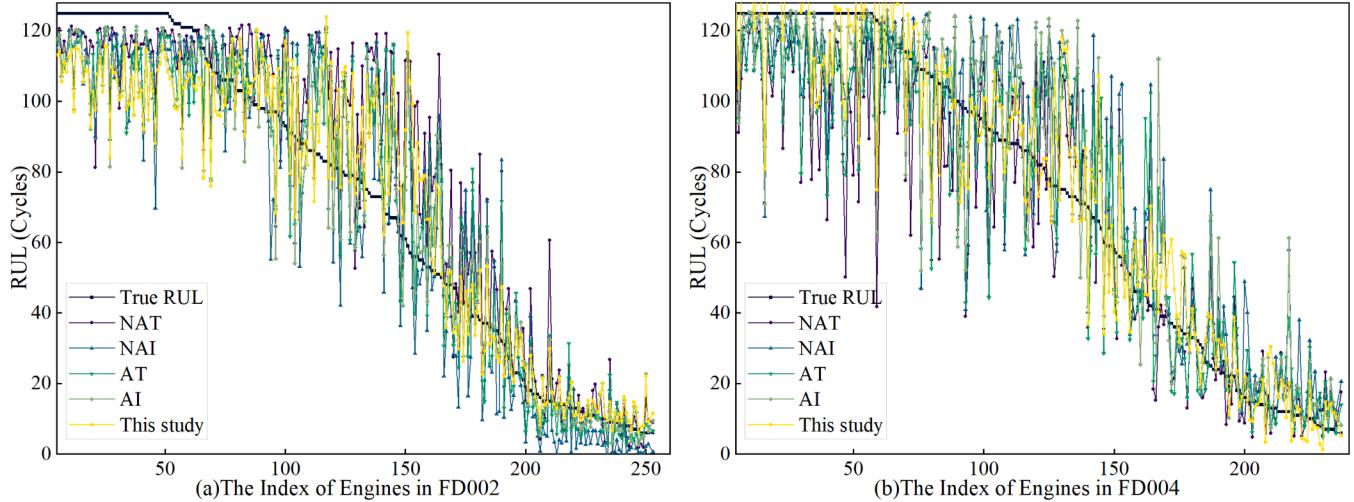


Fig. 7. RUL prediction results for all the engines on the CMAPSS-FD002 and CMAPSS-FD004.

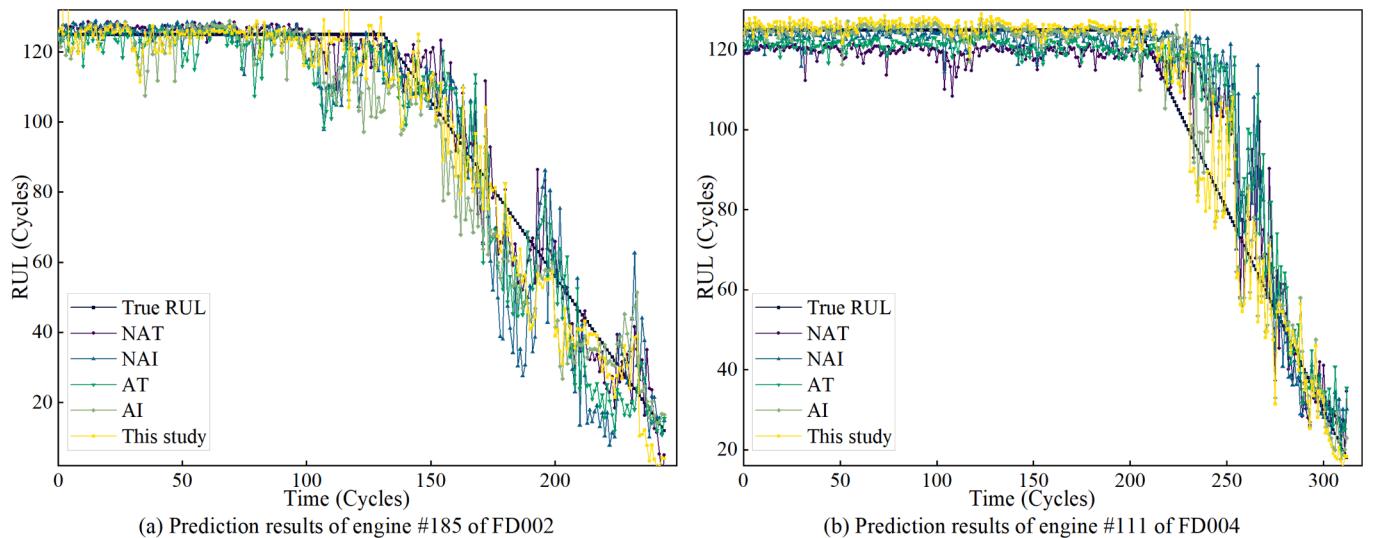


Fig. 8. RUL prediction results for random engines on CMAPSS-FD002 and CMAPSS-FD004.

Table 6

Comparison of performance with those of other methods.

Methods	CMAPSS-FD002		CMAPSS-FD004	
	RMSE	score	RMSE	score
Stacked BLSTM [24]	25.11	4793	26.61	4971
DCNN [12]	22.36	10,412	23.31	12,466
BLSTM-AE [7]	22.07	3099	23.49	3202
AGCNN [10]	19.43	1493	21.50	3393
BiGRU-TSAM [11]	18.94	2264	20.47	3610
MSDCNN-LSTM [42]	18.70	1874	21.57	2699
ABGRU [28]	17.97	2072	21.55	3626
CATA-TCN [37]	17.61	1361	24.04	2303
SCTA-LSTM [26]	16.90	1267	21.93	3310
TCN [15]	16.26	1402	19.78	4092
SE + TCN [15]	16.17	1354	18.86	3975
ABITCN + Informer	15.74	1101	18.60	2144
ABITCI (This study)	12.46	736	15.57	1003

as memory consumption studies. Taking FD002 as an example, the ABITCI is compared with the NAT, NAI, AT, and AI methods in terms of consumption time and memory in each module. Table 7 shows the training time for 50 iterations and the testing time for 259 engines. Table 8 shows the memory consumption for 50 iterations.

Table 7 clearly shows that the total training time and testing time of the proposed ABITCI are the lowest compared with those of the other models. Although the training time of the BiTCN with ECA increases, the range of increase is less than 2 %. The training time of the encoder is reduced by 61.42 %, and the testing time is reduced by 50.46 %. This is due to the sparsity of pyramid self-attention, which covers low correlation values via the shortest propagation path. Table 8 shows that the ABITCI has the lowest memory consumption, which is reduced by 67.64 % compared with those of NAT. In summary, the ABITCI model has high efficiency, with low time and memory consumption, and is suitable for RUL prediction tasks.

3.2. Case 2

The NCMAPSS dataset records the complete run-to-failure degradation trajectories of several engines, and contains eight datasets (DS01 to DS08). Among them, there are two failure modes and nine engines in the DS02 dataset. Since DS02 has been extensively studied, it is selected in this study to further verify the performance of the proposed model.

3.2.1. Data processing

RF is used for data selection and the K-means algorithm is used for condition identification. The 31 sensors are selected and divided into 16 clusters. The z-score is utilized to standardize each cluster. Finally, a time window size of 30 is applied to build the inputs of the model.

Table 7

Comparison of training and testing times.

Methods	Training time (s)					Testing time (s)
	BiTCN	BiTCN with ECA	Encoder	Other Steps	Total	
No ECA + Transformer (NAT)	210.724	N/A	368.450	259.642	838.816	0.053
No ECA + Informer (NAI)	211.291	N/A	279.283	225.296	715.870	0.029
ABITCN + Transformer (AT)	N/A	215.769	364.130	304.374	884.273	0.064
ABITCN + Informer (AI)	N/A	216.585	286.899	207.461	710.945	0.038
ABITCI (This study)	N/A	215.011	142.131	196.972	554.114	0.022

Table 8

Comparison of memory consumption.

Methods	No ECA + Transformer (NAT)	No ECA + Informer (NAI)	ABITCN + Transformer (AT)	ABITCN + Informer (AI)	ABITCI (This study)
Memory consumption (GB)	3.409	2.391	3.538	2.527	1.103

3.2.2. Hyperparameter analysis

The NCMAPSS DS02 dataset exhibits diverse operating conditions and fault modes, and the time window for the same model may vary. Therefore, it is necessary to explore the effect of different window sizes on model performance to reveal the optimal parameter configuration. As in Case I, RMSE and score are used as evaluation metrics. With other constant parameters, the model performance is tested using window sizes ranging from 10 to 60. The experimental results are shown in Fig. 9, which shows the optimal value of window size is 30.

3.2.3. Results and discussions

The first eight engines are used for training and the last engine is used for testing. According to Case I, the hyperparameters of the model are determined. The predicted results are given in Fig. 10. The prediction RULs are close to the true values, and the errors fluctuate within a minor range.

The prediction results of our method compared with those of other methods are shown in Table 9. The RMSE of our method is reduced by 36.96 %–59.53 % on the DS02 dataset, which demonstrates the superior performance of the ABITCI model.

3.2.3.1. Model error analysis. To address the randomness of the neural network, we conducted ten trials and took the means as the final results. The means and variabilities of the trial results are given in Fig. 11.

Fig. 11 shows the RMSEs of the ten trials for the datasets CMAPSS-

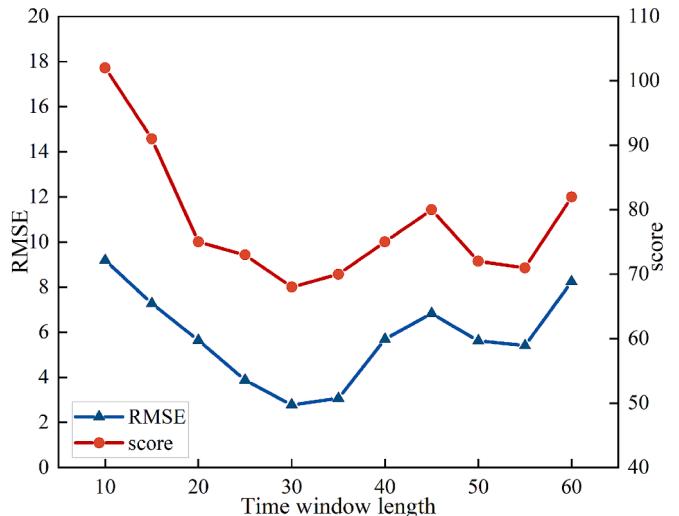


Fig. 9. Performance of the ABITCI model with different time window sizes (NCMAPSS DS02).

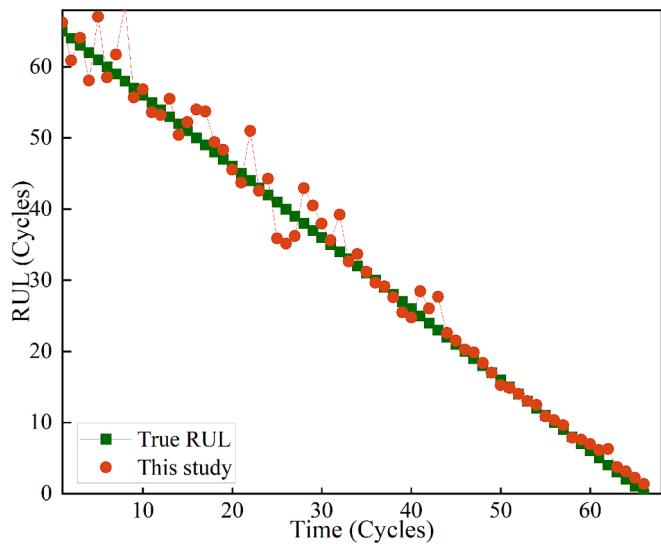


Fig. 10. RUL prediction results on NCMAPSS DS02.

Table 9
Prediction performance comparison with other methods (NCMAPSS DS02).

Methods	RMSE	score
Bi-level LSTM[8]	6.87	2.188×10^{-6}
G-Transformer[43]	5.33	N/A
CNN[44]	4.14	N/A
ABiTCL (This study)	2.78	68

FD002, CMAPSS-FD004, and NCMAPSS DS02. The means are 12.46, 15.57, and 2.78, respectively. And the corresponding standard deviations are 1.11, 1.00, and 0.67. The standard deviations are very low, which demonstrates the stability of our results.

4. Conclusion

In this study, an RUL prediction method based on attention mechanism, BiTCN, and improved Informer for multiple operating conditions is developed. The main innovations are as follows.

- (1) The operating conditions data are merged into the prediction model, which alleviates the influence of differences in conditions on the prediction results.
- (2) The BiTCN based on efficient channel attention, is developed to capture long short-term dependency features. Highly correlated channel information is highlighted by lightweight efficient channel attention. The model calculation is reduced, and the efficiency is improved.
- (3) The Informer based on pyramid self-attention, is developed to extract temporal degradation features in parallel. The masking strategy is used to mask useless information and the distillation mechanism is used to reduce the feature dimension. This reduces the complexity and improves the computational efficiency of the model.

Two aircraft engine datasets with multiple operating conditions are utilized to verify the superiority of the proposed method. The results indicate that the proposed method can perform the RUL prediction task under multiple operating conditions and outperforms most existing methods.

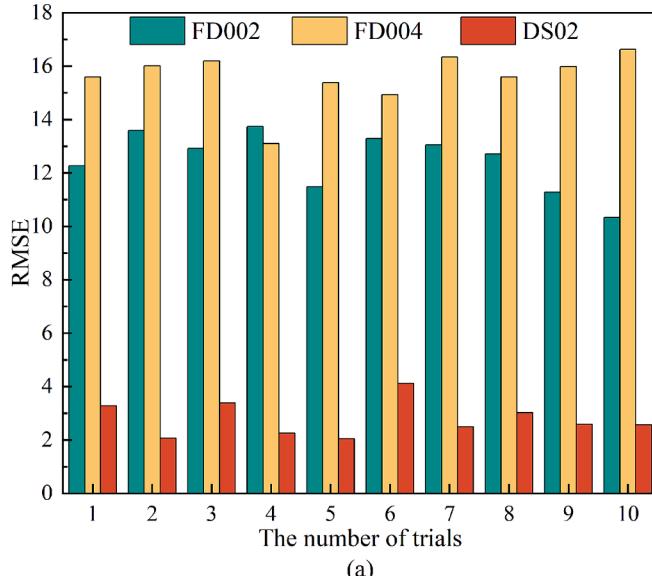
It should be noted that the RUL prediction method in this study is a supervised learning issue. However, labeling samples is a cumbersome work in practical engineering. In future work, we will design a transfer-learning-based approach to perform cross-domain RUL prediction. In this way, the cumbersome work of labeling samples can be avoided. In addition, maintenance decisions are not studied with consideration of RUL prediction in our work. A complete framework for predictive maintenance based on RUL prediction will be designed to support rational and economical maintenance in the future.

CRediT authorship contribution statement

Jie Wang: Writing – original draft, Software, Methodology, Conceptualization. **Zhong Lu:** Writing – review & editing, Supervision, Funding acquisition. **Jia Zhou:** Validation, Supervision. **Kai-Uwe Schröder:** Writing – review & editing, Supervision. **Xihui Liang:** 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



(a)

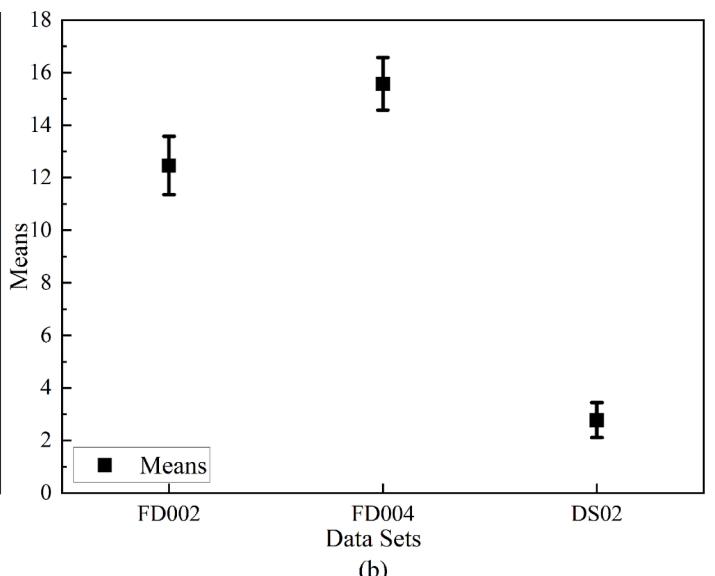


Fig. 11. The results of ten trials.

the work reported in this paper.

Acknowledgments

This work was supported by the Fundamental Research Funds for the Central Universities of China under Grant NG2023003 and the National Key Research and Development Program of China under Grant 2023YFB4302403.

Data availability

Data will be made available on request.

References

- [1] R. Zhu, Y. Chen, W. Peng, Z.-S. Ye, Bayesian deep-learning for RUL prediction: An active learning perspective, Reliab. Eng. Syst. Saf. 228 (2022) 108758, <https://doi.org/10.1016/j.ress.2022.108758>.
- [2] L. Jiang, T. Zhang, W. Lei, K. Zhuang, Y. Li, A new convolutional dual-channel Transformer network with time window concatenation for remaining useful life prediction of rolling bearings, Adv. Eng. Inform. 56 (2023) 101966, <https://doi.org/10.1016/j.aei.2023.101966>.
- [3] P. Ding, J. Xia, X. Zhao, M. Jia, Graph structure few-shot prognostics for machinery remaining useful life prediction under variable operating conditions, Adv. Eng. Inform. 60 (2024) 102360, <https://doi.org/10.1016/j.aei.2024.102360>.
- [4] D. Xu, X. Xiao, J. Liu, S. Sui, Spatio-temporal degradation modeling and remaining useful life prediction under multiple operating conditions based on attention mechanism and deep learning, Reliab. Eng. Syst. Saf. 229 (2023) 108886, <https://doi.org/10.1016/j.ress.2022.108886>.
- [5] I. de Pater, M. Mitici, Developing health indicators and RUL prognostics for systems with few failure instances and varying operating conditions using a LSTM autoencoder, Eng. Appl. Artif. Intell. 117 (2023) 105582, <https://doi.org/10.1016/j.jengappai.2022.105582>.
- [6] J.C. Chen, T.-L. Chen, W.-J. Liu, C.C. Cheng, M.-G. Li, Combining empirical mode decomposition and deep recurrent neural networks for predictive maintenance of lithium-ion battery, Adv. Eng. Inform. 50 (2021) 101405, <https://doi.org/10.1016/j.aei.2021.101405>.
- [7] W. Yu II, Y. Kim, C. Mecheske, Remaining useful life estimation using a bidirectional recurrent neural network based autoencoder scheme, Mech.syst. Signal Process. 129 (2019) 764–780, <https://doi.org/10.1016/j.ymssp.2019.05.005>.
- [8] T. Song, C. Liu, R. Wu, Y. Jin, D. Jiang, A hierarchical scheme for remaining useful life prediction with long short-term memory networks, Neurocomputing 487 (2022) 22–33, <https://doi.org/10.1016/j.neucom.2022.02.032>.
- [9] A. Arunan, Y. Qin, X. Li, C. Yuen, A change point detection integrated remaining useful life estimation model under variable operating conditions, Control Eng. Practice 144 (2024) 105840, <https://doi.org/10.1016/j.conengprac.2023.105840>.
- [10] H. Liu, Z. Liu, W. Jia, X. Lin, Remaining useful life prediction using a novel feature-attention-based end-to-end approach, IEEE Trans. Ind. Inform. 17 (2021) 1197–1207, <https://doi.org/10.1109/TII.2020.2983760>.
- [11] J. Zhang, Y. Jiang, S. Wu, X. Li, H. Luo, S. Yin, Prediction of remaining useful life based on bidirectional gated recurrent unit with temporal self-attention mechanism, Reliab. Eng. Syst. Saf. 221 (2022) 108297, <https://doi.org/10.1016/j.ress.2021.108297>.
- [12] X. Li, Q. Ding, J.-Q. Sun, Remaining useful life estimation in prognostics using deep convolution neural networks, Reliab. Eng. Syst. Saf. 172 (2018) 1–11, <https://doi.org/10.1016/j.ress.2017.11.021>.
- [13] B. Zhang, C. Hu, H. Zhang, J. Zheng, J. Zhang, H. Pei, Remaining useful life prediction method for rolling bearings based on hybrid dilated convolution transfer, Qual. Reliab. Eng. Int. (2024), <https://doi.org/10.1002/qre.3563>.
- [14] W. Ji, J. Cheng, Y. Chen, Remaining useful life prediction for mechanical equipment based on Temporal convolutional network, IEEE, New York, 2019, pp. 1192–1199, <https://doi.org/10.1109/icemi46757.2019.9101706>.
- [15] Q. Zhang, Q. Liu, Q. Ye, An attention-based temporal convolutional network method for predicting remaining useful life of aero-engine, Eng. Appl. Artif. Intell. 127 (2024) 107241, <https://doi.org/10.1016/j.jengappai.2023.107241>.
- [16] S. Ansari, A. Ayob, M.S.H. Lipu, A. Hussain, M.H.M. Saad, Particle swarm optimized data-driven model for remaining useful life prediction of lithium-ion batteries by systematic sampling, J. Energy Storage 56 (2022) 106050, <https://doi.org/10.1016/j.est.2022.106050>.
- [17] A. Ribeiro de Miranda, T.M.G. de Andrade Barbosa, A.G. Scolari Conceicao, S. G. Soares Alcalá, Recurrent Neural Network Based on Statistical Recurrent Unit for Remaining Useful Life Estimation, IEEE, 2019, pp. 425–430, <https://doi.org/10.1109/BRACIS.2019.00081>.
- [18] M. Sayah, D. Guebli, Z. Al Masry, N. Zerhouni, Robustness testing framework for RUL prediction Deep LSTM networks, ISA Trans. 113 (2021) 28–38, <https://doi.org/10.1016/j.isatra.2020.07.003>.
- [19] M.J. Izadi, P. Hassani, M. Raeesi, P. Ahmadi, A novel WaveNet-GRU deep learning model for PEM fuel cells degradation prediction based on transfer learning, Energy 293 (2024) 130602, <https://doi.org/10.1016/j.energy.2024.130602>.
- [20] F. Guo, X. Wu, L. Liu, J. Ye, T. Wang, L. Fu, Y. Wu, Prediction of remaining useful life and state of health of lithium batteries based on time series feature and Savitzky-Golay filter combined with gated recurrent unit neural network, Energy 270 (2023) 126880, <https://doi.org/10.1016/j.energy.2023.126880>.
- [21] J.W. Song, Y.I. Park, J.-J. Hong, S.-G. Kim, S.-J. Kang, Attention-based Bidirectional LSTM-CNN Model for Remaining Useful Life Estimation, IEEE, New York, 2021, pp. 1–5, <https://doi.org/10.1109/ISCASS51556.2021.9401572>.
- [22] Y. Shi, Y. Cui, H. Cheng, L. Li, X. Li, X. Kong, Multi-representation transferable attention network for remaining useful life prediction of rolling bearings under multiple working conditions, Meas. Sci. Technol. 35 (2024) 025037, <https://doi.org/10.1088/1361-6501/ad093a>.
- [23] C. Zhao, H. Shi, X. Huang, Y. Zhang, A multiple conditions dual inputs attention network remaining useful life prediction method, Eng. Appl. Artif. Intell. 133 (2024) 108160, <https://doi.org/10.1016/j.engappai.2024.108160>.
- [24] C.-G. Huang, H.-Z. Huang, Y.-F. Li, A Bidirectional LSTM prognostics method under multiple operational conditions, IEEE Trans. Ind. Electron. 66 (2019) 8792–8802, <https://doi.org/10.1109/TIE.2019.2891463>.
- [25] Y. Xu, T. Xia, Y. Jiang, Y. Wang, D. Wang, E. Pan, L. Xi, A temporal partial domain adaptation network for transferable prognostics across working conditions with insufficient data, Reliab. Eng. Syst. Saf. 250 (2024) 110273, <https://doi.org/10.1016/j.ress.2024.110273>.
- [26] H. Tian, L. Yang, B. Ju, Spatial correlation and temporal attention-based LSTM for remaining useful life prediction of turbofan engine, Measurement 214 (2023) 112816, <https://doi.org/10.1016/j.measurement.2023.112816>.
- [27] H. Jia, Z. Zhang, Y. Gao, F. Shi, A Dual-Stage Attention-Based LSTM Neural Network for Tool Remaining Useful Life Prediction, IEEE, Changzhou China, 2021, pp. 273–277, <https://doi.org/10.1109/ISRMIT53730.2021.9597048>.
- [28] R. Lin, H. Wang, M. Xiong, Z. Hou, C. Che, Attention-based Gate Recurrent Unit for remaining useful life prediction in prognostics, Appl. Soft. Comput. 143 (2023) 110419, <https://doi.org/10.1016/j.asoc.2023.110419>.
- [29] J. Chen, H. Jing, Y. Chang, Q. Liu, Gated recurrent unit based recurrent neural network for remaining useful life prediction of nonlinear deterioration process, Reliab. Eng. Syst. Saf. 185 (2019) 372–382, <https://doi.org/10.1016/j.ress.2019.01.006>.
- [30] P. Ma, G. Li, H. Zhang, C. Wang, X. Li, Prediction of remaining useful life of rolling bearings based on multiscale efficient channel attention CNN and bidirectional GRU, IEEE Trans. Instrum. Meas. 73 (2024) 1–13, <https://doi.org/10.1109/TIM.2023.3347787>.
- [31] G. Jiang, Z. Duan, Q. Zhao, D. Li, Y. Luan, Remaining useful life prediction of rolling bearings based on TCN-MSA, Meas. Sci. Technol. 35 (2024) 025125, <https://doi.org/10.1088/1361-6501/ad07b6>.
- [32] Z. Xu, Y. Zhang, J. Miao, Q. Miao, Global attention mechanism based deep learning for remaining useful life prediction of aero-engine, Measurement 217 (2023) 113098, <https://doi.org/10.1016/j.measurement.2023.113098>.
- [33] J. Guo, S. Lei, B. Du, MHT: A multiscale hourglass-transformer for remaining useful life prediction of aircraft engine, Eng. Appl. Artif. Intell. 128 (2024) 107519, <https://doi.org/10.1016/j.jengappai.2023.107519>.
- [34] F. Xiang, Y. Zhang, S. Zhang, Z. Wang, L. Qiu, J.-H. Choi, Bayesian gated-transformer model for risk-aware prediction of aero-engine remaining useful life, Expert Syst. Appl. 238 (2024) 121859, <https://doi.org/10.1016/j.eswa.2023.121859>.
- [35] Y. Deng, C. Guo, Z. Zhang, L. Zou, X. Liu, S. Lin, An attention-based method for remaining useful life prediction of rotating machinery, Appl. Sci.-Basel 13 (2023) 2622, <https://doi.org/10.3390/app13042622>.
- [36] X. Du, W. Jia, P. Yu, Y. Shi, B. Gong, RUL prediction based on GAM-CNN for rotating machinery, J. Braz. Soc. Mech. Sci. Eng. 45 (2023) 142, <https://doi.org/10.1007/s40430-023-04062-8>.
- [37] L. Lin, J. Wu, S. Fu, S. Zhang, C. Tong, L. Zu, Channel attention & temporal attention based temporal convolutional network: a dual attention framework for remaining useful life prediction of the aircraft engines, Adv. Eng. Inform. 60 (2024) 102372, <https://doi.org/10.1016/j.aei.2024.102372>.
- [38] R. Gong, J. Li, C. Wang, Remaining useful life prediction based on multisensor fusion and attention TCN-BIGRU Model, IEEE Sens. J. 22 (2022) 21101–21110, <https://doi.org/10.1109/JSEN.2022.3208753>.
- [39] Q.-Q. Wang, B.-G. Wu, P.-F. Zhu, P.-H. Li, W.-M. Zuo, Q.-H. Hu, ECA-Net: Efficient channel attention for deep convolutional neural networks, arXiv (2020) 11.
- [40] H. Wang, D. Li, D. Li, C. Liu, X. Yang, G. Zhu, Remaining useful life prediction of aircraft turbofan engine based on random forest feature selection and multi-layer perceptron, Appl. Sci.-Basel 13 (2023) 7186, <https://doi.org/10.3390/app13127186>.
- [41] J. Tang, L. Xiao, The Improvement of Remaining Useful Life Prediction for Aero-engines by Classification and Deep Learning, IEEE, Jinan, China, 2020, pp. 130–136, <https://doi.org/10.1109/PHM-Jinan48558.2020.00030>.
- [42] W. Chen, C. Liu, Q. Chen, P. Wu, Multi-scale memory-enhanced method for predicting the remaining useful life of aircraft engines, Neural Comput. Appl. 35 (2023) 2225–2241, <https://doi.org/10.1007/s00521-022-07378-z>.
- [43] Q. Ma, M. Zhang, Y. Xu, J. Song, T. Zhang, Remaining Useful Life Estimation for Turbofan Engine with Transformer-based Deep Architecture, IEEE, Portsmouth, United Kingdom, 2021, pp. 1–6, <https://doi.org/10.23919/ICAC50006.2021.9594150>.
- [44] M.A. Chao, C. Kulkarni, K. Goebel, O. Fink, Fusing physics-based and deep learning models for prognostics, Reliab. Eng. Syst. Saf. 217 (2022) 107961, <https://doi.org/10.1016/j.ress.2021.107961>.