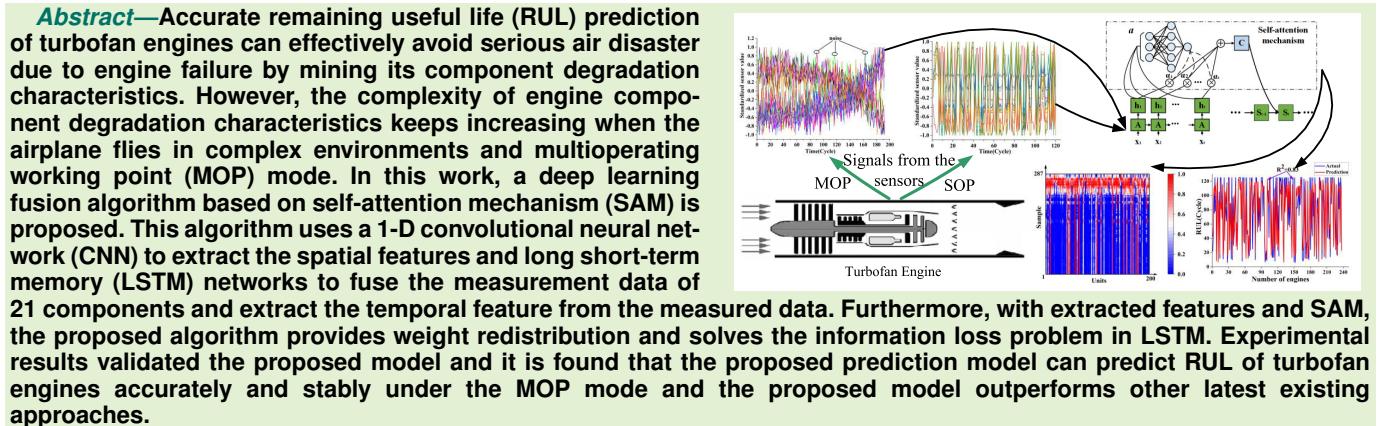


Remaining Useful Life Prediction of Turbofan Engines Using CNN-LSTM-SAM Approach

Jie Li^{ID}, Yuanjie Jia^{ID}, Mingbo Niu^{ID}, Wei Zhu^{ID}, and Fanxi Meng, *Member, IEEE*



Index Terms—Convolutional neural network (CNN), long short-term memory (LSTM), remaining useful life (RUL), self-attention mechanism (SAM), turbofan engine.

I. INTRODUCTION

THE working reliability of turbofan engines is a critical index to measure the flight safety of aircraft. A detailed and thorough overhaul is carried out before each flight taking-off to ensure such safety. This traditional maintenance method may cause unnecessary and partially repeated maintenance, which increases airlines operating cost. Nowadays, prognostics and health management (PHM) has been applied in various areas, such as finance [1], medicine [2], [3], and engine facilities [4], [5]. Remaining useful life (RUL) prediction is a key technology in PHM of the turbofan engines. Accurate RUL prediction can not only reduce the

Manuscript received 19 January 2023; revised 15 March 2023; accepted 20 March 2023. Date of publication 30 March 2023; date of current version 1 May 2023. This work was supported in part by the Science and Technology Planning Project of the State Administration of Market Supervision and Administration under Grant 2021MK104 and in part by the Key Research and Development Program of Shaanxi Province under Grant 2022GY-178. The associate editor coordinating the review of this article and approving it for publication was Dr. Pedro Oliveira Conceicao Junior. (*Corresponding author: Wei Zhu.*)

Jie Li and Mingbo Niu are with the Internet of Vehicle to Road Research Laboratory, Chang'an University, Xi'an 710064, China (e-mail: jli@chd.edu.cn; ivr.niu@chd.edu.cn).

Yuanjie Jia, Wei Zhu, and Fanxi Meng are with the School of Electronics and Control Engineering, Chang'an University, Xi'an 710064, China (e-mail: 2019232025@chd.edu.cn; wzhu@chd.edu.cn; 2020232006@chd.edu.cn).

Digital Object Identifier 10.1109/JSEN.2023.3261874

number of invalid maintenance and achieve efficient condition maintenance but also avoid flight accidents caused by engine failure.

Existing RUL prediction approaches of turbofan engine can be mainly divided into two categories: physical model-based and data-driven approaches [6], [7], [8]. The physical model-based method has obvious shortcomings. A core of this method is to establish a mathematical model, which can accurately describe the operation processes of a research object, based on full operation parameters and object preknowledge. However, the typical structure of a turbofan engine is highly complex with closely coupled components. Therefore, a physical engine RUL model can hardly be built directly. Without a full understanding on the complex operation mechanism of a turbofan engine, the data-driven method can construct the RUL model merely according to the collected historical operating data, as such the data-driven method has become a research focus.

Since Hinton et al. [9] introduced the concept of deep learning in 2006, it has emerged as a key technology in image processing, speech recognition, fault diagnosis, and many other fields due to its powerful modeling and representing abilities. Deep learning conducts forward training and reverse fine-tuning of data through constructing a multilayer neural network and deeply mining the hidden features in data to extract accurate prediction results. Recently, deep learning has been applied in RUL prediction for many systems such

as battery and bearing, which includes 1-D convolutional neural network (CNN), long short-term memory (LSTM) networks, gray neural network, deep feature disentanglement transfer learning network, transfer learning, and so on [10], [11], [12], [13], [14], [15], [16], [17], [18], [19].

RUL prediction of turbofan engines is characterized in multidimension, and CNN is an expert in feature extraction from multidimension. Therefore, CNN has been applied in RUL prediction. Li et al. [10], [11], [12] used convolution kernels in different sizes for establishing a deep CNN (DCNN) to extract data features and predict the engine RUL. RUL prediction is time-related and LSTM can extract the temporal feature. Therefore, Hu et al. [13], Siahpour et al. [14], Saxena et al. [15], and Liu et al. [16] developed an LSTM-based prediction model for engine RUL prediction and fault diagnosis. Chen et al. [17] constructed a data-mining model based on a multilayer LSTM and a standard feedforward layer and predicted the RUL for a variety of operation conditions, faults, and degradation models.

Many other deep learning methods are also employed for RUL prediction purpose. For example, Fan et al. [18] used transfer learning and a consistent self-organizing model to create transferable features to establish RUL regression models. Zhang et al. [19] proposed a restricted Boltzmann machine (RBM), which used an unsupervised self-organizing graph algorithm to predict RUL for mechanical equipment. In addition, some fusion neural network models were used for RUL or performance prediction, especially for RUL of lithium-ion batteries, such as variational mode decomposition-particle filter-Gaussian process regression (VMD-PF-GPR), LSTM-broad learning system (BLS), particle swarm optimization (PSO)-BLS, and VMD-CNN-LSTM [20], [21], [22], [23]. These fusion neural networks can integrate the advantages of each neural network and improve the prediction accuracy and stability.

The research above has made significance exploration in RUL prediction, but the accuracy and stability of RUL prediction are still to be improved. In particular, most deep learning algorithms assign equal importance to multidimensional features in the multidimensional data-driven model. However, the importance of different features in actual tasks is not exactly the same, and thus, it is desired to weigh the importance of the degradation information contained in each feature. The Google deep mind proposed an attention mechanism (AM) to improve the feature extraction ability of recurrent neural network (RNN) models in 2014 [24]. AM assigns different attention weights to different features so that the data-driven model can pay more attention on the important parts of a training process. It offers better results in the fields of image processing and machine translation [25], [26]. Furthermore, Chen et al. [27] proposed AM-RNN to predict RUL-related health indicators of the rolling bearings and then obtained the final RUL through linear regression. Liu et al. [28] applied AM to the input conditions monitoring data to weigh different input characteristics and then used bidirectional gated recurrent unit (BGRU) and CNN to encode important information. Finally, a fully connected network was used to decode the encoded information predicted by RUL. The results show that AM can improve the accuracy of RUL prediction. However, it is

noted that AM only affects the inputs. With this AM feature in mind, the inputs and outputs linkage could be weakened with the increase of neutral network inputs' dimensions. This will result in a lower weights assigning ability of AM. In the RUL prediction of turbofan engines, 24-D input features need to be considered, and thus, AM mechanism for the RUL prediction of turbofan engines remains uncertain.

RUL prediction of turbofan engines based on single-operating working point (SOP) mode has been well-studied. However, the turbofan engine often works in the multioperating working points (MOPs) mode. Under the MOP mode, the component degradation characteristics depend on both time change and their working point states, which makes it difficult to reflect the actual degradation of the turbofan engine from sensor data of the components [12], [16], [28].

Based on the aforementioned studies and taking different outputs weight into consideration, a self-AM (SAM)-based CNN-LSTM fusion algorithm with optimized weights assigning is proposed in this article to provide a comprehensive model illustrating the impacts from optimized output weights on predicting RUL of turbofan engines. Apart from AM-CNN, as a portion of neutral network, SAM directly treats the outputs from LSTM. Instead of considering input features, SAM reassigns weights to hiding layers of neutral units. This new approach assists end-to-end training of neural networks, reinforces linkage of input and outputs, and achieves considerable improvement in weights assigning as well as algorithm convergence.

In the proposed algorithm, CNN and LSTM are employed to extract the spatial and temporal features of the engine sensors, respectively, and then, SAM is utilized to adaptively assign different attention degrees to data features, accurately obtain the key information in the data, and perform accurate RUL prediction. The engine degradation dataset of NASA was used in this work to verify the proposed algorithm [15]. The main contributions of this work are summarized as follows.

- 1) SAM is introduced into fusion CNN-LSTM RUL prediction of turbofan engines. SAM can provide the enhanced adaptive ability to input features of artificial neural network through input weight reassignment during the training process.

- 2) The different mechanisms of AM and SAM in the fusion neural network are studied. The regulation ability and range of AM and SAM are investigated in detail under the level of neuron unit, which improves the interpretability of the proposed fusion neural network.

The rest of this article is organized as follows. Section II describes the proposed RUL prediction model. In Section III, the proposed model is verified by testing the dataset of NASA. Network layers and batch size of CNN-LSTM-SAM (CLS) are analyzed, and then, the mechanism of SAM and AM is compared. The effectiveness and the superiority of CLS are also demonstrated through the comparison between the proposed algorithm and other typical deep learning-based algorithms. Section IV concludes this article.

II. NETWORK MODEL CONSTRUCTION

The process of the proposed network model construction is shown in Fig. 1, which is mainly composed of model training and RUL prediction. These two parts share data preprocessing

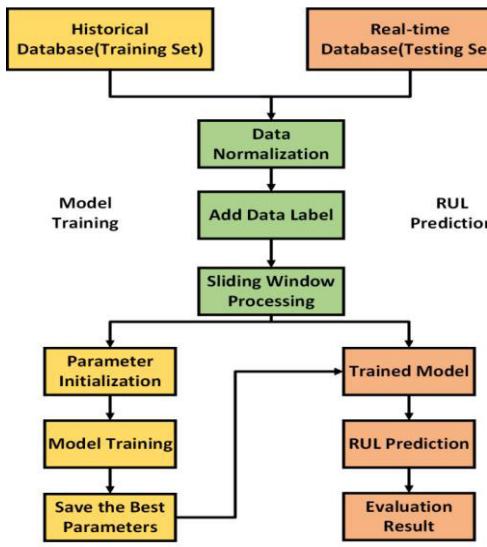


Fig. 1. Flowchart of network model construction.

results, which include data normalizing, label adding, and time windows processing. Model training is mainly to establish an optimal neural network prediction model through historical data (training set) of the turbofan engines. RUL prediction is to serve real-time data (testing set) into an established model to achieve accurate real-time RUL prediction.

A. Improved Variational Mode Decomposition

1) *Data Normalization*: The dataset study contains a variety of engine sensors' degradation data with different units and magnitude ranges [28]. The features of the data with smaller value may be easily lost without data preprocessing, and thus, the data normalization method is used to map all the input data to $[-1, 1]$ to improve the convergence speed and accuracy of model training. Based on previous studies [13], [28], the min-max normalization described in the following equation is used to scale the input data to $[-1, 1]$:

$$x_i^{j*} = 2 \left(X_i^j - X_{\min}^j \right) / \left(X_{\max}^j - X_{\min}^j \right) - 1 \quad (1)$$

where x_i^{j*} and X_i^j represent the j th sensor in the i th cycle after and before normalization, respectively. In addition, because different engine operating cycles are not the same, X_{\min}^j and X_{\max}^j are the minimum and maximum values in the j th sensor data of the same engine, respectively.

2) *Label Adding*: It is normally applied after data normalization. Generally, as turbofan engines have experienced constricted factory test before delivery, there is almost no performance degradation, which could be detected in their early working period. Therefore, the RUL prediction in the middle and late stages of engine life is more meaningful. For this reason, the RUL prediction is divided into two stages as constant value invariance of early state and linear degradation of middle and late states [6].

When RUL is greater than the set threshold value, the engine is considered healthy and its RUL label keeps constant. When RUL is in the linear degradation stage, the engine is considered likely to fail at any time. Since the minimum operating life of the engine is 128 in all the experimental data, the RUL threshold is set as 125 [13], [14], as shown in Fig. 2.

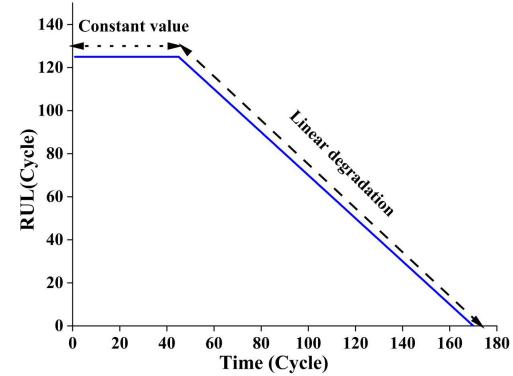


Fig. 2. Labels of the datasets, where "time" means the running time of the turbofan engine and "RUL" is the remaining useful life.

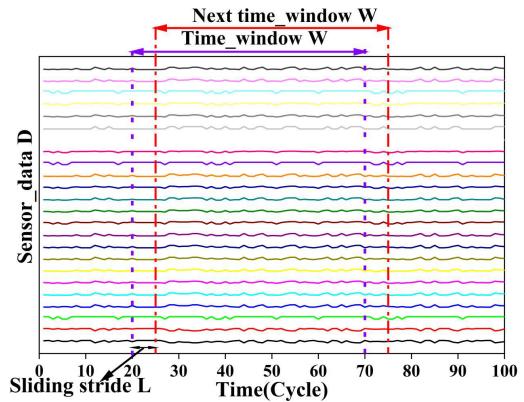


Fig. 3. Sliding window processing along time dimension, where there are training samples with D features using a time window of length W .

3) *Time Window Selection*: Due to the high dependence between data and time, it is necessary to select an appropriate time window to capture this dependent relation, which is the key to deal with time series. A sliding time window is applied to generate network inputs for the dataset, and a sample sequence of size $D*W$ is generated for model training, where W represents the window width and D represents the data dimension, as shown in Fig. 3 [13].

The first input sample sequence is expressed as $I_1 = [X_1, X_2, \dots, X_W]$, and after the sliding operation with a step length of L , the next input sample sequence can be expressed as $I_2 = [X_{1+L}, X_{2+L}, \dots, X_{W+L}]$ so that the N th engine circulates a total of C cycles; a total of $(C-W*L+1)$ sample sequences are generated [13], [28]. Each sample sequence is trained by a neural network, and the final output corresponds to the RUL label value of the last cycle of the time window. An appropriate length of time window can improve the extraction efficiency of the time series features.

B. Model Building

CLS is composed of CNN, LSTM, and SAM, as shown in Fig. 4. In this model, CNN is employed to extract spatial features and effectively suppress noise, and LSTM is applied to extract temporal features. Furthermore, SAM is added to capture important information in the extracted features for

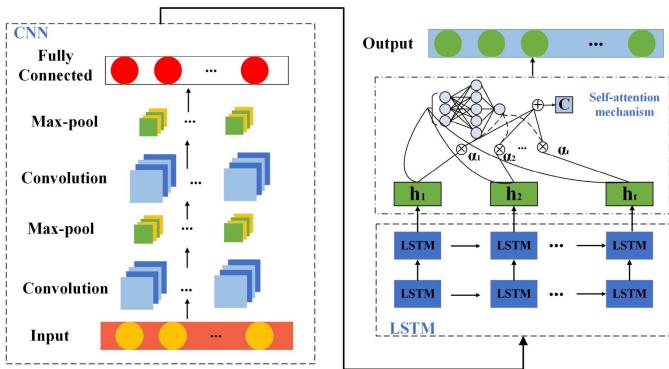


Fig. 4. Schematic of CNN-LSTM-SAM.

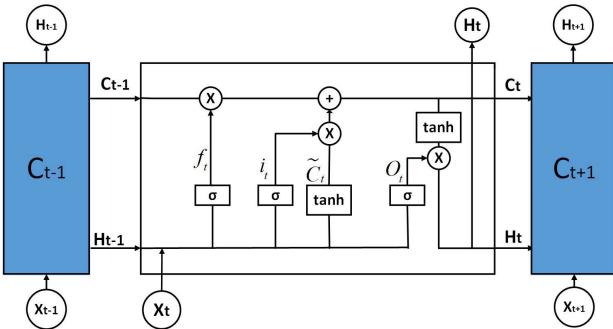


Fig. 5. Schematic of LSTM.

prediction. Each part of the model is discussed in detail as follows.

1) Convolutional Neural Network: Each layer of CNN consists of a convolutional layer and a pooling layer. A 1×3 convolution kernel is designed for the convolutional layer to extract the spatial characteristics of the information. Then, the output of the convolutional layer is downsampled through the max-pooling layer to compress and extract the features in the convolutional layer. Finally, the extracted features are merged through the fully connected layer. Thus, the correlation between multidimensional data is mined, where its noise and unstable components are extracted by CNN [29]. In addition, the padding of the convolutional layer is set to be “same” to keep the data size unchanged.

2) Long Short-Term Memory: It is employed to solve the problem of gradient disappearance and gradient explosion during RNN training. As shown in Fig. 5, information is transmitted from cells in LSTM and it is easy to capture the long-term dependence relationship between the preceding moments and the following moments in the time sequence. In Fig. 5, C_{t-1} and C_{t+1} represent the information state of the cell at the previous time and the next time, respectively. A unit cell in the middle represents the information state at current time. The unit cell is designed to control the transmission of information, which is composed of forget gate, input gate, and output gate. This structure ensures that LSTM neurons have the ability of mining and storing the long-term dependent relation. The forget gate determines what kind of cell information can be forgotten, the input gate determines which information should be added to the cell state according to the choice of the forget gate, and the output gate determines

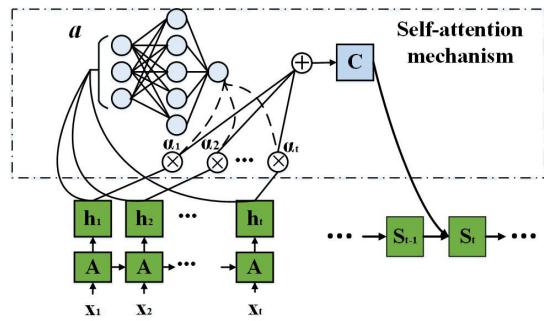


Fig. 6. Schematic of SAM.

which information will be output as the current state. They are expressed as follows [15], [30], [31]:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2)$$

$$O_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (3)$$

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (4)$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (5)$$

$$C_t = f_t \otimes C_{t-1} + i_t \otimes \tilde{C}_t \quad (6)$$

$$h_t = O_t \otimes \tanh(C_t) \quad (7)$$

where W_f , W_i , W_c , and W_o are the weight of the data; b_f , b_i , b_c , and b_o denote the input node, forget gate, input gate, and output gate deviation, respectively; i_t , f_t , O_t , and \tilde{C}_t represent the input gate, forget gate, output gate, and new candidate state values, respectively; C_{t-1} and C_t are the LSTM neuron states at $t - 1$ and t , respectively; \otimes denotes the elementwise multiplication of two vectors; and σ and \tanh are the activation functions.

3) Self-Attention Mechanism: The AM is proposed based on the research of human vision and its main idea is to generate a weight matrix based on the input information. It is known that the data-driven model necessitates a large number of parameters to extract features from the data, and the ability of feature extraction increases with the number of model parameters. However, too many parameters may lead to incomplete extraction of required features due to information redundancy. Therefore, a weight matrix can be generated by AM, and the key information can be assigned with higher weights. Thus, the model can accurately capture critical information and reduce the attention of useless information, then effectively solves the problem of information redundancy, and improves the effectiveness and accuracy of the neural network [28], [29], [31].

Regardless of the length of input data sequence, LSTM encodes the sequence into a fixed-length vector expression and thus limits the model learning ability of long sequences when LSTM is applied to extract data features. Therefore, SAM is introduced to extract the time series of data, as shown in Fig. 6. S_t is an LSTM hidden state for time t in Fig. 6. When the information passes between LSTM cells, vectors in the hidden state sequence h_t are fed into the learnable function “ a ” to produce a weight vector α_t , and then, the vector C is obtained by calculating the weighted average of the hidden state vector h_t and weight α_t [32], [33].

In order to search for the best matched element, traditional AM assigns input sequence weights based on outside search. However, this mechanism ignores the self-feature of neuron sequence and units' interconnection. In this article, SAM is proposed and it focuses on the units after LSTM. The search target is independent of outside information and depends on units' interconnection. The working process of SAM is described as follows.

CNN abstracted sensor information spatially, which lowered the information dimension effectively. Then, LSTM is applied to catch the dependence of pre- and post-information relationship. SAM will compensate the lost information from LSTM. The caught key information will be assigned with higher weights to reduce the interference from redundant knowledge. In this way, SAM directly affects the neutral units instead of input features. Therefore, it is named SAM in this work.

SAM can be briefly described through the following three formulas:

$$C = \sum_{i=1}^t \alpha_i h_i \quad (8)$$

$$\alpha_i = \exp(e_i) / \sum_{k=1}^t \exp(e_k) \quad (9)$$

$$e_i = a(h_i) = \sigma(Wh_i + b) \quad (10)$$

where t is the sum of time steps of the input LSTM network, h is the eigenvector output by the LSTM network, α_i is the weight of the vector h , σ is the activation function, W is the weight matrix from the input layer to the hidden layer, and b is the bias matrix. Notice that C has the different meaning in (6) and (8), C of (6) is the state of LSTM at a certain time, and C in (8) is obtained by calculating the weighted average of the hidden state vector and weight in LSTM based on the weight generated by SAM.

Neural network model is trained by the back propagation (BP) algorithm in this work and the Adam algorithm is used to optimize the model parameters to minimize the training error. Thus, the weight matrix can be obtained and the key features in the information can be captured by the weight matrix after model training [31].

C. Summary of the Proposed Method

First, the proposed CLS employs a convolutional layer and a pooling layer in CNN to extract spatial features from data, where CNN can also effectively suppress the noise in data. Second, the characteristics of LSTM cells are applied to extract time features in data, which effectively captures the dependency relationship between the preceding moments and the following moments in the current state. Finally, SAM is introduced to obtain the key information from the data by assigning different weights to the initially extracted features and, then, RUL prediction is established based on the key information with different weights. Table I summarizes the algorithm details of this proposed method.

III. EXPERIMENT ANALYSIS

A. Description of Dataset

A turbofan engine benchmark dataset is chosen to test and verify the effectiveness of the proposed method. This

TABLE I
ALGORITHM OF CLS

ALGORITHM1: Proposed CLS Approach	
Step 1	Input: Hyper-parameters of the trained model (epoch, dropout rate, batch size, etc.), training set Output: Trained model for RUL prediction
Step 2	Data Preprocessing: Feature selection, data normalization, add data label, sliding window processing
Step 3	Build: CNN-LSTM-SAM model
Step 4	Initialize: The parameters of CNN-LSTM-SAM model initialization. Model Training: for $i = 1, 2, \dots, \text{epoch}$ do: Model building by CNN , obtaining data spatial characteristics Model building by LSTM , obtaining data time characteristics Model building by SAM , obtaining data key characteristics The model is trained by BP algorithm, and Step 4.1–4.3 represent the forward process of model training Step 4.4 Backward process of model training. Loss function is MSE. The Adam algorithm is used for parameter optimization, which reduces the training error. End for.

TABLE II
CMAPSS DATASET DESCRIPTION

Data set	FD001	FD002	FD003	FD004
Training engines	100	260	100	249
Testing engines	100	259	100	248
Fault modes	1	1	2	2
Operation conditions	1	6	1	6

dataset was created by NASA and it simulates the actual degradation of turbofan engine, which is called commercial modular aero-propulsion system simulation (C-MAPSS) dataset [15], [34]. Therefore, sensors signal used in this study is from this simulation software, and the C-MAPSS dataset is divided into four subsets according to different failure modes and operation conditions [28], [35], namely, FD001–FD004. The specific characteristics of each subset are shown in Table II. For detailed parameter selection, measurement, and simulation calculation process of C-MAPSS, please see [15].

The failure modes in Table II include two types of failures: fan degradation (type 1) and high-pressure compressor degradation (type 2). Type 1 indicates that only one engine failure is monitored, and type 2 represents that two failure modes are monitored at the same time. Operation conditions represent the number of operating points of the turbofan engines. FD001 and FD003 are the testing data of engine sensors when the operation conditions are “1,” while FD002 and FD004 are the testing data collected when the operation conditions is “6.”

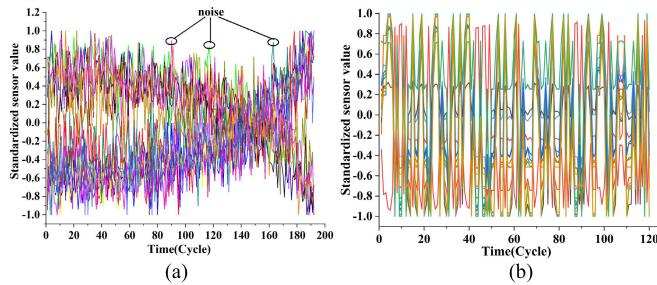


Fig. 7. Sensors data of turbofan engine components. Data of (a) FD001 and (b) FD002.

TABLE III
CLASSIFICATION TABLE FOR DIFFERENT FLIGHT CONDITIONS

Operating points	Alt/Kft	M	TRA /°
1	0	0	100
2	10	0.25	100
3	20	0.7	100
4	25	0.62	60
5	35	0.84	100
6	42	0.84	100

Operation conditions greatly influence the sensors signals, which is found in FD001 (in SOP mode) and FD002 (in MOP mode). The component degradation process of SOP mode and MOP mode is shown in Fig. 7(a) and (b), respectively. It can be concluded from Fig. 7 that the trend of component degradation is simple and clear in the SOP mode, while this trend is extremely complicated in the MOP mode.

FD002 and FD004 both contain six operation points and they are shown in Table III. The six operating points are distinguished by different altitudes (Alt), Mach number (M), and throttle resolver angles (TRAs). Fig. 8(a) shows that the operating points of a group of engines are changed with working time in FD002 and it can be seen that an engine switches frequently between the six operating points as the operating time changes. Thus, the signals from the sensors of the components do not show a regular degradation trend. It is characterized by strong nonlinearity, as shown in Fig. 8(b).

B. Performance Metrics

R^2 , score, and root-mean-square error (RMSE) are used to evaluate the performance of the proposed RUL prediction algorithm. The definitions of R^2 , score, and RMSE are given as follows [11], [12], [13], [14], [16], [17]:

$$R^2 = 1 - \sum_{i=1}^n (\hat{y}_i - y_i)^2 / \sum_{i=1}^n (\bar{y}_i - y_i)^2 \quad (11)$$

$$\text{Score} = \begin{cases} \sum_{i=1}^n \left(e^{-\frac{\hat{y}_i - y_i}{13}} - 1 \right), & \hat{y}_i - y_i < 0 \\ \sum_{i=1}^n \left(e^{\frac{\hat{y}_i - y_i}{10}} - 1 \right), & \hat{y}_i - y_i > 0 \end{cases} \quad (12)$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2} \quad (13)$$

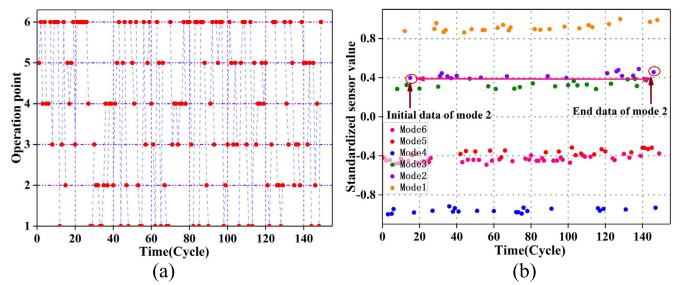


Fig. 8. Conversion process of engine operating mode. (a) Operating points of a group of engines. (b) Nonlinearity analysis.

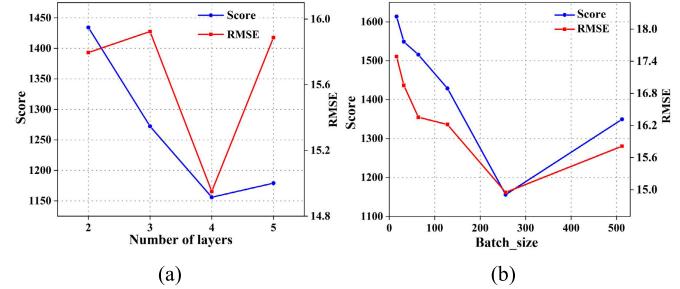


Fig. 9. Influence of model parameters. (a) Results of different layers. (b) Results of different batch sizes.

where n represents the total number of predicted samples, \hat{y}_i is the predicted value of the engine life, y_i is the true value of the engine life, and \bar{y}_i is the average value of the engine life. The same penalty is set for RMSE in the early prediction (i.e., the predicted RUL is shorter than the actual RUL) and delayed prediction (i.e., the predicted RUL is greater than the actual RUL). The penalty is increased for score in delayed prediction. This is because the delayed prediction of RUL prediction often brings greater serious consequences than early prediction in practical applications.

C. Selection of Network Parameters

All algorithms in this article are conducted on PyCharm 2019 and Python 3.7. The computing device is an Intel core i7-9750H (2.7 GHz) CPU and 16-GB RAM computer, and a large number of experiments are conducted by using FD002 to optimize the parameters of the prediction models.

The number of network layers greatly affects the accuracy of a model. The increase of network layer will improve the accuracy of the model, but it will also cause the calculation surge and overfitting of the model. According to the data size and the calculation complexity in this study, two-layer CNN is set to extract the spatial features of the data and suppress the noise.

Furthermore, in order to investigate the characteristics of LSTM in this study in detail, the RUL prediction results of different layers of LSTM are compared in Fig. 9(a). By comparison, it is found that the four-layer LSTM achieves the best result, and an overfitting problem occurs when the network layers continue increasing. RUL prediction results of different batch sizes are compared in Fig. 9(b) and 256 is the relatively good result. The detailed neural network parameters of the CLS are shown in Table IV. Ten experiments were carried

TABLE IV
MODEL PARAMETERS OF CLS

Parameter	Value	Parameter	Value
hidden neurons of Conv1D	8/16	optimizer	Adam
convolution kernel size	1*3	dropout	0.1
hidden neurons of LSTM	50/100/150/200	padding	same
hidden neurons of SAM	200	dense	1
data dimension	16/24/17/24	epoch	150
batch size	256	time window	30

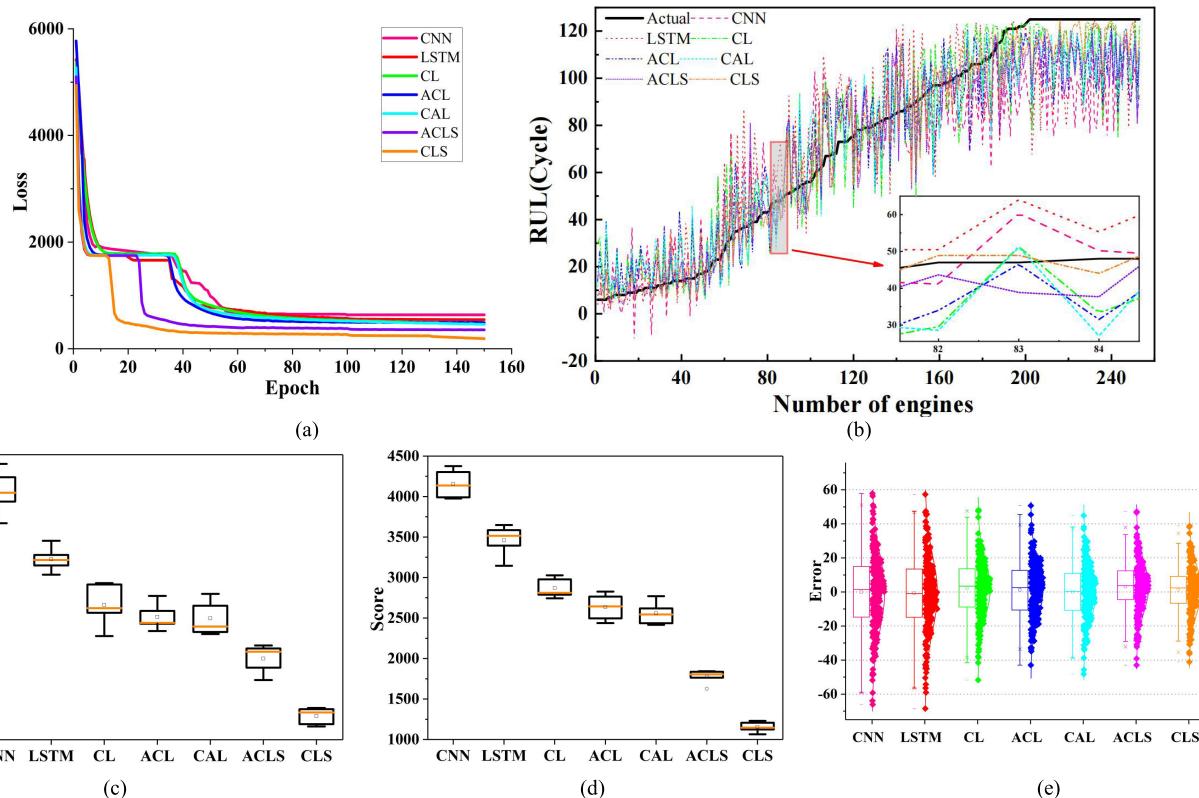


Fig. 10. Comparison of different algorithms. (a) Trend of loss in the training process. (b) Prediction results of different algorithms. (c) RMSE of different algorithms. (d) Score of different algorithms. (e) Boxplots of prediction error.

out and the average values of RMSE and score were taken to indicate the RUL prediction results, as shown in Table V.

D. Results Discussion

Seven RUL prediction architecture models are discussed to find out how the SAM mechanism affects RUL prediction. They are CNN, LSTM, CL (CNN-LSTM), ACL (AM-CNN-LSTM), CAL (CNN-AM-LSTM), ACLS (AM-CNN-LSTM-SAM), and CLS.

The obtained results are shown in Fig. 10. In Fig. 10(a), loss function changes are presented for the seven models using database FD002. In Fig. 10(b), RUL prediction accuracy is given for different models. RMSE, score, and prediction error are given in Fig. 10(c)–(e), respectively. It can be found that CLA offers the best convergence speed. CLA also beats other models in accuracy and stability. Compared to classical prediction models, ACL improved prediction performance with the use of AM at the input. However, CAL with SAM is found to outperform ACL. Therefore, the proposed SAM

TABLE V
EXPERIMENTAL RESULTS OF EACH DATASET

	FD001	FD002	FD003	FD004
RMSE	12.56	15.25	13.77	18.63
Score	261.03	1155.97	252.92	2425.86

is good at catching important information in the MOP mode, compared to that of AM.

To further investigate the SAM mechanism, AM in ACL, and SAM in CAL, weights assigning processes are analyzed and the results in Fig. 11 are obtained. Fig. 11(a) reveals the adjustment process of neurons by SAM, where 200 units were shown along the x -coordinate. Fig. 11(b) shows 24 input features (on x -coordinate) adjustment by AM. Fig. 11(c) shows the neurons' adjustment process in the ACLS model by SAM. Fig. 11(d) compares the neuron adjustment processes by SAM in CLA and in ACLS with solid line and dashed line,

respectively. (In this figure, 7, 35, and 16 are three neurons randomly selected.)

From Fig. 11(a) and (b), it can be found that input features weights assignment by AM reveals dramatic variations. Weights at different times differed from each other significantly. One reasonable explanation of weights repeatedly variation is that, in a given state, a large change in AM assigned weight does not effectively improve the model performance, and thus, its weight value goes back or is adjusted in another direction. The randomness and uncertainty will be increased for input features assignment if such a large variation keeps showing up. Clearly, the input features extracted by AM are insufficient with lower convergency. In contrast, SAM assigns neuron weights in a hiding layer directly, while there is much less variation in its assigning process. Moreover, SAM assignment offers better flexibility because the adjustable number of hidden units is higher than that of input features. Wider and deeper extraction of important information by SAM in CAL results in consistent and stable assigning process. To sum up, the different regulating objects and mechanisms of SAM and AM are the key reasons for the efficiency and convergence of feature extraction of the two models.

It is worth mentioning that although ACLS is not the best in the aforementioned comparison, ACLS outperforms ACL and CAL. This is because SAM in ACLS can compensate the shortcomings of LSTM despite the incompletely input information extracted from AM. However, compared to CLS, AM in ACLS revised input features and may cause unnecessary weight changes, so the subsequent process of SAM becomes more complicated. This could be a potential reason that CLS still outperforms ACLS. Fig. 11(d) reveals the adjustment processes of ACLS and CLS on the same neuron. Clearly, the latter one is more stable.

In a nutshell, the proposed SAM in the CLS model offers a superior performance in weight assignment, deeper information extraction, and better stability.

The RUL prediction details of CLA model are shown in Fig. 12; one engine is randomly selected from F001–F004 for real-time RUL prediction and the different comparison results are shown in Fig. 12(a)–(d).

As shown in these figures, the prediction errors of CLA in the early operation of engines are larger than those in the late operation. This is due to the fact that there is more degradation information as failure approaches, which leads to high prediction accuracy in the late stage, especially when the RUL is close to 0. Furthermore, all the data of engine sensors are used for RUL prediction in batches and the results are shown in Fig. 13.

In addition, the predicted results of each subset are compared with the true RUL value to obtain the number of errors within a certain range, as shown in Fig. 14. It can be seen from Fig. 14 that more than 90% of the errors are concentrated in $[-20, 20]$.

Under the same computing hardware environment, the computation time of the main neural network models is shown in Table VI. Setting the computation time of LSTM as standard time (100%), it can be seen from Table VI that, compared with CL and LSTM, the computation time of CLS is slightly more

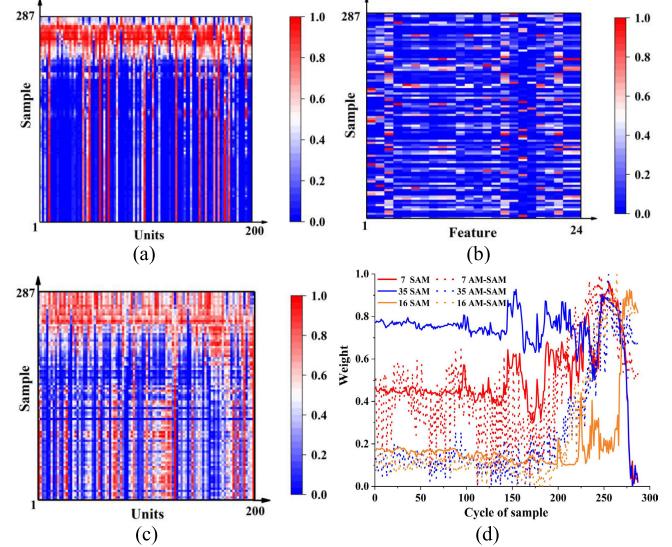


Fig. 11. Analysis of AM or SAM in different algorithms. (a) Thermodynamic diagram of SAM. (b) Thermodynamic diagram of AM. (c) Thermodynamic diagram of AM-SAM. (d) Weight change process of some neurons.

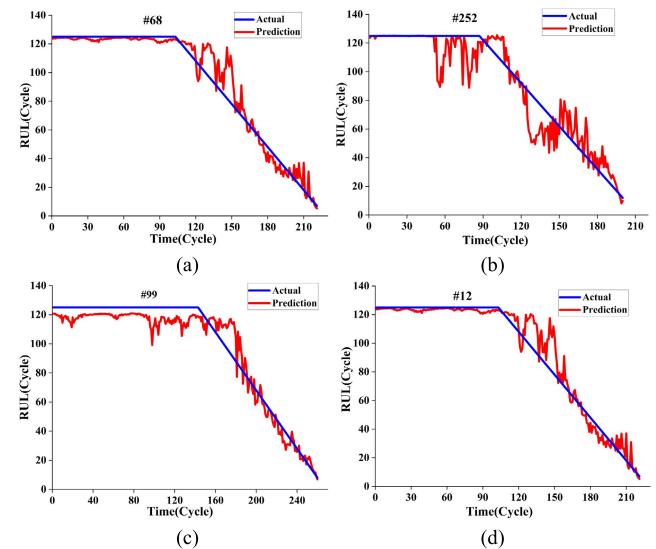


Fig. 12. Comparison of the predicted RUL and actual RUL for a single engine. Results of (a) FD001, (b) FD002, (c) FD003, and (d) FD004.

TABLE VI
COMPARATIVE STUDY OF COMPUTATION TIME

Methods	MLP	LSTM	CL	CLS
Percentage of computation time	177%	100%	78%	82%

than CL and less than LSTM. Therefore, it can be concluded that although the model structure of CLS is more complex, it does not lead to a significant increase in computation time.

E. Comparisons With Other Algorithms

Many approaches have been applied to C-MAPSS dataset, including support vector machine (SVM) [36], multilayer perceptron (MLP) [36], multiobjective belief network (MODBNE) [36], deep belief network (DBN) [36], LSTM [17], DCNN [12], feature-attention-based

TABLE VII
RMSE COMPARISON OF DIFFERENT ALGORITHMS

Methods	FD001	FD002	FD003	FD004	Average
SVM [36]	40.7	53	46.3	60	45
MLP [36]	16.8	28.8	18.5	31	23.7
DBN [36]	15.2	27.1	14.7	29.9	21.7
MODBNE [36]	15	25	12.5	28.7	20.3
LSTM [17]	16.1	24.5	16.2	28.1	21.2
DCNN [12]	12.6	22.4	12.6	23.3	17.7
AGCNN [28]	12.4	19.4	13.4	21.5	16.7
CEEMD-DLSTM [37]	14.7	29.0	17.7	33.4	23.7
BiGRU [38]	-	26.5	-	29.1	27.8
BLSTM [38]	-	25.1	-	26.6	25.9
BiGRU-TSAM [39]	12.6	18.9	12.5	20.5	16.1
SAM-CNN-LSTM	12.6	15.3	13.8	18.6	15.1
Imp	—	19.0%	—	9.3%	6.2%

TABLE VIII
SCORE COMPARISON OF DIFFERENT ALGORITHMS

Methods	FD001	FD002	FD003	FD004	Average
SVM [36]	7703	316483	22541	141122	129462
MLP [36]	561	14027	480	10444	6377
DBN [36]	418	9032	442	7954	4461
MODBNE [36]	334	5585	422	6557	3224
LSTM [17]	338	4450	852	5550	2797
DCNN [12]	274	10412	284	12466	5858
AGCNN [28]	226	1492	227	3392	1334
CEEMD-DLSTM [37]	262	6953	452	15069	5684
BiGRU [38]	-	6352	-	6339	6346
BLSTM [38]	-	4793	-	4971	4882
BiGRU-TSAM [39]	213	2264	233	3610	1580
SAM-CNN-LSTM	261	1156	253	2425	1024
Imp	—	22.5%	—	28.5%	23.2%

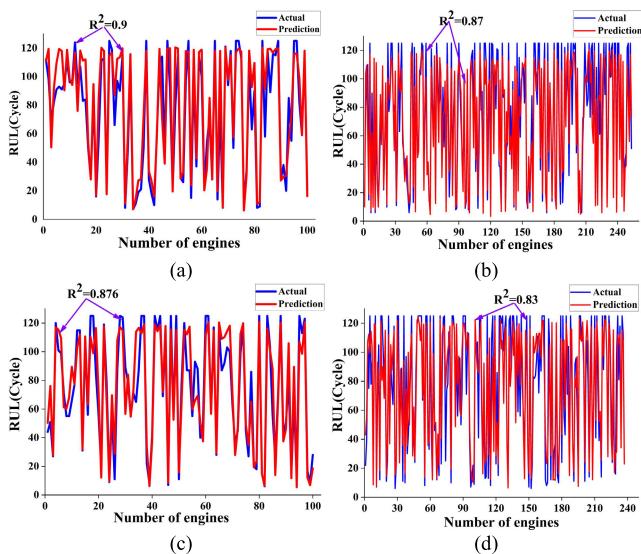


Fig. 13. Comparison of the predicted RUL and actual RUL of results. Results of (a) FD001, (b) FD002, (c) FD003, and (d) FD004.

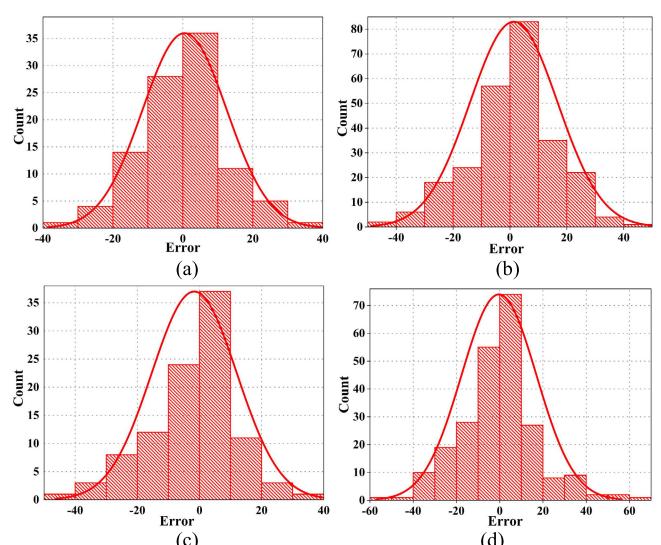


Fig. 14. Diagram of error analysis. Results of (a) FD001, (b) FD002, (c) FD003, and (d) FD004.

bidirectional bidirectional gated recurrent unit CNN model (AGCNN) [28], and so on [37], [38], [39]. The pro-

posed approach is compared with these public approaches to verify the superiority of CLA in this article. The

comparison results are shown in Tables VII and VIII. Compared with other approaches, CLA has significantly improved the prediction performance. Taking the FD002 (MOP mode) and FD004 (MOP mode) datasets as examples, compared with the best existing results, the RMSE of our approach is reduced by 19.0% and 9.3%, respectively, and the score is reduced by 22.5% and 28.5%, respectively. Although the performance of the proposed model is not the best for FD001 (SOP mode) and FD003 (SOP mode), comparable results are obtained. In particular, compared with the best existing approach, the averages of RMSE and score have been reduced by 6.2% and 23.2%, respectively. The turbofan engines often work in the MOP mode in reality, and therefore, the proposed approach offers a good application prospect for RUL prediction.

It can be seen that the proposed method works better for MOP mode than for SOP mode. The possible reason is that the working state of the engine under SOP is stable, and its RUL features are relatively obvious. Therefore, compared with other neural networks, the proposed algorithm has no significant advantage in the extraction ability of RUL feature, while when the engine works in MOP mode, the state information of the engine is extremely nonlinear. SAM can better sort out the turbulence information and extract the key information at this time, thus improving the prediction effect.

IV. CONCLUSION

In this work, a CNN-LSTM approach based on the SAM is proposed to solve complex multidimensional time series prediction problems. First, the time window is set to intercept the specified data sample size as the input of the neural network. Second, the spatial feature of the engine sensors is extracted by CNN and the time feature is extracted by LSTM. Finally, different weights are assigned according to different importance of the input features by SAM, and thus, the neural network pays more attention to the important parts of the inputs and suppresses uselessness information. This considerably improves the accuracy of the prediction results. The proposed approach was validated on the C-MAPSS dataset and the detailed comparison results were obtained.

The experimental results show that when the turbofan engine is in the MOP mode, the proposed prediction model can adaptively weigh the importance of the extracted features and the useful information is effectively captured. CLS is compared with other typical deep learning-based algorithms in the MOP mode, and CLS outperforms the existing algorithms in RMSE and score index by 14.2% and 25.5%, respectively. In addition, compared with the existing best results, the RMSE and score are reduced by an average of 6.2% and 23.2%, respectively. Therefore, the proposed model achieved higher accuracy and stability in RUL prediction. In the following, a frequency-domain analysis will be incorporated into deep learning models to further improve the prediction performance in our future study.

REFERENCES

- [1] L. B. Andersen, D. Häger, S. Maberg, M. B. Næss, and M. Tunland, "The financial crisis in an operational risk management context—A review of causes and influencing factors," *Rel. Eng. Syst. Saf.*, vol. 105, pp. 3–12, Sep. 2012.
- [2] T. Kontogiannis and S. Malakis, "A systemic analysis of patterns of organizational breakdowns in accidents: A case from helicopter emergency medical service (HEMS) operations," *Rel. Eng. Syst. Saf.*, vol. 99, pp. 193–208, Mar. 2012.
- [3] M. D. C. Moura, J. M. Santana, E. L. Drogue, I. D. Lins, and B. N. Guedes, "Analysis of extended warranties for medical equipment: A Stackelberg game model using priority queues," *Rel. Eng. Syst. Saf.*, vol. 168, pp. 338–354, Dec. 2017.
- [4] J. Xu, Y. Wang, and L. Xu, "PHM-oriented integrated fusion prognostics for aircraft engines based on sensor data," *IEEE Sensors J.*, vol. 14, no. 4, pp. 1124–1132, Apr. 2014.
- [5] K.-Y. Chen, "Forecasting systems reliability based on support vector regression with genetic algorithms," *Rel. Eng. Syst. Saf.*, vol. 92, no. 4, pp. 423–432, Apr. 2007.
- [6] H. Li, W. Zhao, Y. Zhang, and E. Zio, "Remaining useful life prediction using multi-scale deep convolutional neural network," *Appl. Soft Comput.*, vol. 89, Apr. 2020, Art. no. 106113.
- [7] J. Lee, F. J. Wu, W. Y. Zhao, M. Ghaffari, L. X. Liao, and D. Siegel, "Prognostics and health management design for rotary machinery systems-reviews, methodology and applications," *Mech. Syst. Signal Process.*, vol. 42, nos. 1–2, pp. 314–334, Jan. 2014.
- [8] H. Cai, J. Feng, W. Li, Y.-M. Hsu, and J. Lee, "Similarity-based particle filter for remaining useful life prediction with enhanced performance," *Appl. Soft Comput.*, vol. 94, Sep. 2020, Art. no. 106474.
- [9] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, May 2006.
- [10] P. Li et al., "An end-to-end neural network framework for state-of-health estimation and remaining useful life prediction of electric vehicle lithium batteries," *Renew. Sustain. Energy Rev.*, vol. 156, Mar. 2022, Art. no. 111843.
- [11] L. Chen, Y. Ding, B. Liu, S. Wu, Y. Wang, and H. Pan, "Remaining useful life prediction of lithium-ion battery using a novel particle filter framework with grey neural network," *Energy*, vol. 244, Apr. 2022, Art. no. 122581.
- [12] Q. Zhang et al., "A deep learning method for lithium-ion battery remaining useful life prediction based on sparse segment data via cloud computing system," *Energy*, vol. 241, Feb. 2022, Art. no. 122716.
- [13] T. Hu, Y. Guo, L. Gu, Y. Zhou, Z. Zhang, and Z. Zhou, "Remaining useful life prediction of bearings under different working conditions using a deep feature disentanglement based transfer learning method," *Rel. Eng. Syst. Saf.*, vol. 219, Mar. 2022, Art. no. 108265.
- [14] S. Siahpour, X. Li, and J. Lee, "A novel transfer learning approach in remaining useful life prediction for incomplete dataset," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–11, 2022.
- [15] A. Saxena, G. Kai, D. Simon, and N. Eklund, "Data-driven remaining useful life prediction via multiple sensor signals and deep long short-term memory neural network," in *Proc. Int. Conf. Prognostics Health Manage.*, 2008, pp. 1–9.
- [16] L. Liu, X. Song, and Z. Zhou, "Aircraft engine remaining useful life estimation via a double attention-based data-driven architecture," *Rel. Eng. Syst. Saf.*, vol. 221, May 2022, Art. no. 108330.
- [17] C. Chen, J. Shi, N. Lu, Z. H. Zhu, and B. Jiang, "Data-driven predictive maintenance strategy considering the uncertainty in remaining useful life prediction," *Neurocomputing*, vol. 494, pp. 79–88, Jul. 2022.
- [18] Y. Fan, S. Nowaczyk, and T. Rögnvaldsson, "Transfer learning for remaining useful life prediction based on consensus self-organizing models," *Rel. Eng. Syst. Saf.*, vol. 203, Nov. 2020, Art. no. 107098.
- [19] J. Zhang, Y. Jiang, X. Li, M. Huo, H. Luo, and S. Yin, "An adaptive remaining useful life prediction approach for single battery with unlabeled small sample data and parameter uncertainty," *Rel. Eng. Syst. Saf.*, vol. 222, Jun. 2022, Art. no. 108357.
- [20] C. Zhang, S. Zhao, and Y. He, "An integrated method of the future capacity and RUL prediction for lithium-ion battery pack," *IEEE Trans. Veh. Technol.*, vol. 71, no. 3, pp. 2601–2613, Mar. 2022.
- [21] S. Zhao, C. Zhang, and Y. Wang, "Lithium-ion battery capacity and remaining useful life prediction using board learning system and long short-term memory neural network," *J. Energy Storage*, vol. 52, Aug. 2022, Art. no. 104901.
- [22] K. Wiktorowicz and T. Krzeszowski, "Training high-order Takagi–Sugeno fuzzy systems using batch least squares and particle swarm optimization," *Int. J. Fuzzy Syst.*, vol. 22, no. 1, pp. 22–34, Nov. 2019.
- [23] J. Li, R. Li, Y. Jia, and Z. Zhang, "Prediction of short-term photovoltaic power via codec neural network and mode decomposition based deep learning approach," *Energy Sci. Eng.*, vol. 10, no. 6, pp. 1794–1811, Nov. 2021.

- [24] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu, "Recurrent models of visual attention," in *Proc. NIPS*, 2014, pp. 2204–2212.
- [25] W. Yin, H. Schütze, B. Xiang, and B. Zhou, "ABCNN: Attention-based convolutional neural network for modeling sentence pairs," *Trans. Assoc. Comput. Linguistics*, vol. 4, pp. 259–272, Dec. 2016.
- [26] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
- [27] Y. Chen, G. Peng, Z. Zhu, and S. Li, "A novel deep learning method based on attention mechanism for bearing remaining useful life prediction," *Appl. Soft Comput.*, vol. 86, Jan. 2020, Art. no. 105919.
- [28] H. Liu, Z. Liu, W. Jia, and X. Lin, "Remaining useful life prediction using a novel feature-attention-based end-to-end approach," *IEEE Trans. Ind. Informat.*, vol. 17, no. 2, pp. 1197–1207, Feb. 2021.
- [29] M. Sajjad et al., "CNN-based anti-spoofing two-tier multi-factor authentication system," *Pattern Recognit. Lett.*, vol. 126, pp. 123–131, Sep. 2019.
- [30] W. Sun, R. Zhao, R. Yan, S. Shao, and X. Chen, "Convolutional discriminative feature learning for induction motor fault diagnosis," *IEEE Trans. Ind. Informat.*, vol. 13, no. 3, pp. 1350–1359, Jun. 2017.
- [31] H. Zhang, Q. Zhang, S. Shao, T. Niu, and X. Yang, "Attention-based LSTM network for rotatory machine remaining useful life prediction," *IEEE Access*, vol. 8, pp. 132188–132199, 2020.
- [32] C. Raffel and D. P. W. Ellis, "Feed-forward networks with attention can solve some long-term memory problems," 2015, *arXiv:1512.08756*.
- [33] L. Wu, Y. Wang, X. Li, and J. Gao, "Deep attention-based spatially recursive networks for fine-grained visual recognition," *IEEE Trans. Cybern.*, vol. 49, no. 5, pp. 1791–1802, May 2019.
- [34] X. Fang, K. Paynabar, and N. Gebraeel, "Multistream sensor fusion-based prognostics model for systems with single failure modes," *Rel. Eng. Syst. Saf.*, vol. 159, pp. 322–331, Mar. 2017.
- [35] H. Wu and S. Prasad, "Semi-supervised deep learning using pseudo labels for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1259–1270, Mar. 2018.
- [36] C. Zhang, P. Lim, A. K. Qin, and K. C. Tan, "Multiobjective deep belief networks ensemble for remaining useful life estimation in prognostics," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2306–2318, Oct. 2017.
- [37] S. Zhao, Y. Zhang, S. Wang, B. Zhou, and C. Cheng, "A recurrent neural network approach for remaining useful life prediction utilizing a novel trend features construction method," *Measurement*, vol. 146, pp. 8792–8802, Jun. 2019.
- [38] C. Huang, H. Huang, and Y. Li, "A bidirectional LSTM prognostics method under multiple operational conditions," *IEEE Trans. Ind. Electron.*, vol. 66, no. 11, pp. 2306–2318, Nov. 2019.
- [39] J. Zhang, Y. Jiang, S. Wu, X. Li, H. Luo, and S. Yin, "Prediction of remaining useful life based on bidirectional gated recurrent unit with temporal self-attention mechanism," *Rel. Eng. Syst. Saf.*, vol. 221, May 2022, Art. no. 108297.



Yuanjie Jia received the B.S. degree in measurement and control technology and instrument from the Minsheng College, Henan University, Kaifeng, Henan, China, in 2018. He is currently pursuing the M.S. degree in control engineering with Chang'an University, Xi'an, China.

His current research interests include deep learning and its applications.



Mingbo Niu received the B.Eng. degree in electronic engineering and the M.Sc. (Eng.) (Hons.) degree in communication and information systems from Northwestern Polytechnical University, Xi'an, China, in 2003 and 2006, respectively, and the Ph.D. degree in electrical and computer engineering from the University of British Columbia, Vancouver, BC, Canada, in 2013.

His current research interests include the Internet of Vehicles (IoV), vehicle-to-road (V2R) infrastructure, cooperative artificial intelligence (AI) microgrids, massive multiple-input–multiple-output (MIMO), image signal processing, low-carbon smart cities, energy harvesting, and electronic circuit theory.



Wei Zhu received the B.Eng. and Ph.D. degrees in microelectronics from Nanyang Technological University (NTU), Singapore, in 2007 and 2013, respectively.

She is currently an Associate Professor of Artificial Intelligence (AI) Theory and Applications with Chang'an University, Xi'an, China. Her current research interests include machine learning and intelligent design.



Jie Li received the B.S., M.S., and Ph.D. degrees in automation from Northwestern Polytechnical University, Xi'an, China, in 2006, 2009, and 2012, respectively.

He is currently an Associate Professor of Artificial Intelligence (AI) Theory and Applications with Chang'an University, Xi'an. His current research interests include machine learning and its applications.



Fanxi Meng (Member, IEEE) received the B.S. degree in measurement and control technology and instrumentation from the Zhongyuan University of Technology, Zhengzhou, China, in 2020. He is currently pursuing the M.S. degree in control engineering with Chang'an University, Xi'an, China.

His current research interests include deep learning and its applications.