Remaining Useful Life Estimation of Aircraft Engines Based on Deep Convolution Neural Network and LightGBM Combination Model

Lijun Liu^{1,2} · Lan Wang¹ · Zhen Yu¹

Received: 26 October 2020 / Accepted: 30 August 2021
© The Author(s) 2021

Abstract

Accurately predicting the remaining useful life (RUL) of aero-engines is of great significance for improving the reliability and safety of aero-engine systems. Because of the high dimension and complex features of sensor data in RUL prediction, this paper proposes a model combining deep convolution neural networks (DCNN) and the light gradient boosting machine (LightGBM) algorithm to estimate the RUL. Compared with traditional prognostics and health management (PHM) techniques, signal processing of raw sensor data and prior expertise are not required. The procedure is shown as follows. First, the time window of raw data of the aero-engine is used as the input of DCNN after normalization. The role of DCNN is to extract information from the input data. Second, considering the limitations of the fully connected layer of DCNN, we replace it with a strong classifier-LightGBM to improve the accuracy of prediction. Finally, to prove the effectiveness of the proposed method, we conducted some experiments on the C-MAPSS data set provided by NASA, and obtained good accuracy. By comparing the prediction effect with other commonly used algorithms on the same data set, the proposed algorithm has obvious advantages.

Keywords Deep convolution neural network · Light gradient boosting machine · Aero-engine · Remaining useful life prediction

1 Introduction

The aero-engine is a crucial component that provides thrust for a plane [1]. To ensure the safety of the aircraft, it is important to estimate the RUL of the engine. Prognostics and health management (PHM) is an emerging technology [2], which aims to monitor the reliability and the security of an engineering system for improving the maximum operating availability and reducing maintenance cost [3, 4]. As one of the most challenging technologies in PHM, the RUL estimation of aero-engines has attracted much attention.

RUL prediction methods are mainly divided into three categories: model-based methods, empirical knowledge-based methods and data-driven methods. The model-based methods establish the model through mechanical principles

which take a lot of time. Besides, it is difficult to build an accurate model due to the complex system structure and uncertain environments [6]. The empirical knowledge-based methods require industry experts to use extensive prior knowledge to establish the corresponding knowledge base [7]. This method does not require an accurate model, but the prediction accuracy cannot be guaranteed. Data-driven methods build estimate models based on historical run-to-failure data, which avoid the limitations of relying on physical failure models and expert knowledge [8]. Moreover, data-driven approaches have the advantages of low computing cost and high accuracy. This paper mainly focuses on data-driven methods for predicting the RUL.

Many data-driven prediction methods have been proposed and achieved good results in recent years, including support vector machines (SVM) [9], hidden Markov models [10], etc. Traditional data-driven methods analyze and mine sensor data by signal processing technologies, and extract features that reflect system degradation and failure, and implement the RUL prediction of equipment. However, it still remains challenging to develop an effective approach to

✉ Zhen Yu
yuzhen20@xmu.edu.cn

¹ School of Aerospace Engineering, Xiamen University, Xiamen, China

² Shenzhen Research Institute of Xiamen University, Shenzhen, China

mine complex data information of time series and achieve high accuracy prediction.

In the past years, deep learning has gradually emerged in the field of PHM. It is more capable to extract deep features of big data composed of multi-sensor performance parameters. Malhi et al. [11] adopted a recurrent neural network (RNN) approach to long-term prognostics of machine health status. As an optimization of the traditional RNN, a long short-term memory (LSTM) method which can make full use of the sensor sequence information and expose hidden patterns within sensor data was proposed for RUL estimation by Zheng et al. [12]. Li et al. [13] predicted the RUL of aero-engines by building a 2-dimensional (2D) DCNN based on time series from the sensor signals. Within the deep learning architecture, DCNN has fewer parameters than other methods because it adopts weight sharing technology and shows excellent feature extraction ability. However, these nonlinear combination features extracted by DCNN are only learned in a simple manner. Therefore, the ability of DCNN to search for the global optimum is limited.

In recent years, decision tree ensemble methods have been widely used by data scientists [14]. Extreme gradient boosting (XGBoost) which is based on gradient boosting decision tree (GBDT) has achieved promising results in many machine learning challenges [15]. As an improvement of XGBoost, the light gradient boosting machine (LightGBM) is better at processing data with high dimension data [16] and adopts a leaf-wise strategy to improve prognostic accuracy.

To solve the issues in DCNN prediction methods, a model combining DCNN and LightGBM for predicting the RUL of aircraft engines is proposed in this paper. We use the deep features extracted by DCNN as the input of LightGBM to get more accurate prediction results. The effectiveness of this approach is validated on C-MAPSS datasets provided by NASA.

The rest of the paper is organized as follows. In Sect. 2, we briefly review CNN and describe the specific structure of deep learning models. Then the method of model improvement is presented in Sect. 3, along with an introduction of LightGBM. In Sect. 4, we analyze C-MAPSS datasets and demonstrate the superiority of DCNN-LightGBM algorithm by comparisons with other methods. Finally, concluding remarks are provided in Sect. 5.

2 Deep Learning Architecture

2.1 Convolutional Neural Network

Convolutional neural networks (CNNs) were first proposed by LeCun, which have many outstanding achievements in the fields of image processing and natural language processing

[17]. In general, CNNs are structured by three types of hidden layers composed of convolutional layers, pooling layers, and fully connected layers [18].

2.1.1 Convolutional Layer

The convolutional layer is the most important part of convolutional networks. Feature maps are produced by sliding the convolution kernel on data and convolving with the covered data. And the property of shared weights reduces model parameters and the risk of overfitting. The calculation process of the i -th feature map of the l -th convolutional layer x_i^l is as follows:

$$x_i^l = \varphi(z_i^l), \quad (1)$$

$$z_i^l = k_i^l * x^{l-1} + b_i^l = \sum_{c=1}^C k_{i,c}^l * x_c^{l-1} + b_i^l, \quad (2)$$

where z_i^l represents the output of convolution operations, $*$ denotes the convolution operator, k_i^l is the i -th convolution kernel, x^{l-1} is the input volume, b_i^l and $\varphi(\cdot)$ represent the bias term and nonlinear activation function, respectively. C is the number of input channels.

2.1.2 Pooling Layer

The purpose of a pooling layer is to merge similar features into one using nonlinear down-sampling functions and speed up the calculation. The max-pooling layer is the most commonly used pooling layer. The inputs of the pooling layer are the feature map from the previous layers, and the outputs are the maximum of a local patch of the inputs. The function is as follows:

$$x_i^l = \max(x_i^{l-1}, p, s), \quad (3)$$

where x_i^l the is i -th feature map of the l -th pooling layer, x_i^{l-1} is the i -th feature map in the previous layer $l-1$, $\max(\cdot)$ means the max-pooling, p and s represent the pooling size and the stride size, respectively.

2.1.3 Fully Connected layer

As the last layer of the convolutional neural network, the fully connected layer summarizes the features and outputs the prediction results. The output x^l of the l -th fully connected layer is as follows:

$$x^l = \varphi(\omega^l x^{l-1} + b^l), \quad (4)$$

where x^{l-1} is the output of the previous layer $l-1$, ω^l and b^l represent the weight matrix and the bias vector, respectively.

Table 1 The result of the model error with or without pooling layers

Pooling layer	RMSE
With	19.7989
Without	14.8625

Fig. 1 The effect of the number of convolutional layers

2.2 Proposed Deep Convolutional Neural Network Structure

DCNN has excellent learning ability, which is mainly achieved using multiple nonlinear feature extractions. It can automatically learn hierarchical representations from data. Therefore, the number of pooling layers and convolution layers and the size of the convolution kernel will have a great impact on the prediction results. The aero-engine degradation simulation data used in this paper are numerical data and the dimension of the raw feature is relatively low. Although the pooling operation improves

the computing efficiency, some useful information is filtered for this kind of prognostic problems. Table 1 shows the different forecasting effect with or without pooling layers in the model. It can be seen that the network structure without pooling layers has better results.

By analyzing the data set in this paper and considering the poor correlation of features from different sensors and the results of literatures 13 and 15, the filter size is 10×1 and the filter number is 10. A larger convolution kernel can reduce the impact of noise. The effect of the number of convolution layers on the result error is shown in Fig. 1. It can be seen that the network with 5 convolutional layers has the best performance.

A primary network structure suitable for the RUL prediction is designed based on the above experiments. Figure 2 shows the proposed network structure for the RUL estimation in this study. First, the input data is two-dimensional (2D). One dimension is the time sequences of sensors, denoted as L_s , and the other is the number of features, denoted as L_f . The raw features are signals collected by the multiple sensors. (The detail of the data preparation will be illustrated in Sect. 4). Next, four convolutional layers with the same structure extract input data features. Zero-padding ensures that the dimensions of the output feature map are consistent with that of the input. The dimension of the input is $L_s \times L_f$. Then, a convolutional layer with 1 filter combines the feature maps. The filter size is 3×1 . The small size of the convolution kernel is conducive to extracting more subtle features. In this way, advanced features hidden in the raw data are extracted. Afterwards, the two-dimensional feature will be flattened and passed to a fully connected

layer which has 100 neurons. In addition, we use dropout technique to relieve overfitting after the flattened layer. Finally, the output layer contains a neuron whose output represents the predicted value of the RUL.

Fig. 2 Proposed deep convolution neural network structure for the RUL estimation

The activation function of each layer is RELU. *He_normal* initializer is used to initialize the weights of DCNN [19]. DCNN is trained to minimize the loss based on the back propagation algorithm. Adam algorithm is chosen as the optimizer in our experiment. The initial learning rate is set as 0.001, and it will be divided by 10 for every 30 epochs until convergences. Considering the actual situation of the aero-engine datasets, we increase the penalty for late predictions, and then the loss function can be represented as:

$$loss = \frac{1}{N} \sum_{i=1}^N l_i, \quad (5)$$

$$l_i = \begin{cases} 2(y_i - \hat{y}_i), & y_i < \hat{y}_i \\ (y_i - \hat{y}_i), & y_i \geq \hat{y}_i \end{cases}, \quad (6)$$

where y_i is the actual value of the i -th test engine, $\hat{y}_i$ is the predicted value of the i -th test engine, and N is the number of the validation set.

3 Model Improvement

Convolutional neural network has shown the excellent ability of feature extraction. These features that have been abstracted by multiple convolutions are integrated by a fully connected layer. However, the fully connected layer learns the nonlinear combination features in a simple manner, this method will fall into the local optimal value when there is serious noise mixed in the raw data. Therefore, we replace the fully connected layer with a strong classifier namely LightGBM.

3.1 The Light Gradient Boosting Machine

The light gradient boosting machine (LightGBM), proposed by Microsoft in 2017, is mainly applied to solve the problem of accuracy and efficiency when using gradient boosting decision tree (GBDT) to process massive amounts of data. Like GBDT, LightGBM also learns a decision tree (DT) through negative gradient fitting residuals in each iteration [20, 21]. In this paper, the input of LightGBM is a feature vector created by DCNN. Given the dataset $D = \{(x_i, y_i)\} (x_i \in R^{L_s \times L_f}, y_i \in R, i \leq n)$, where n is the number of samples, y_i is the target RUL value. The LightGBM architecture is described in Fig. 3.

K additive functions are used to predict the output, which is defined as follows:

$$\hat{y}_i = \sum_{k=1}^K f_k(x_i), f_k \in F \quad (6)$$

Fig. 3 The structure of LightGBM for the RUL prediction

where K is the number of trees, F is a set of regression tree, f_k is one of the trees with the leaf scores. The predicted values of all trees are added to get the RUL estimation. The training loss function is defined as:

$$L(\varphi) = \sum_{i=1}^n l(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k), \quad (7)$$

where l is the training loss function, Ω is the regularization function. The loss function is the square error. To improve the optimization speed and the generalization of model, the second-order Taylor expansion is applied [15]. The loss can be represented as:

$$L^{(k)} = \sum_{i=1}^n l(y_i, \hat{y}_i^{(k-1)} + f_k(x_i)) + \Omega(f_k), \quad (8)$$

$$\approx \sum_{i=1}^n \left[l(y_i, \hat{y}_i^{(k-1)}) + g_i f_k(x_i) + \frac{1}{2} h_i f_k^2(x_i) \right] + \Omega(f_k)$$

$$g_i = \partial_{\hat{y}_i^{(k-1)}} L(y_i, \hat{y}_i^{(k-1)}), \quad (9)$$

$$h_i = \partial_{\hat{y}_i^{(k-1)}}^2 L(y_i, \hat{y}_i^{(k-1)}). \quad (10)$$

Then LightGBM uses the following two methods to speed up the forecasting speed without sacrificing accuracy.

- (1) Gradient-based one-side sampling (GOSS). Sample points with large gradients will contribute more information gain. GOSS algorithm retains these sample points with large gradients and randomly sample the sample points with small gradients for keeping the accuracy of the information gain evaluation.
- (2) Exclusive feature bundling (EFB). LightGBM uses the histogram algorithm for the merge of exclusive features. EFB algorithm puts many features of high-dimensional data together in a sparse feature space to reduce the number of features.

3.2 The DCNN-LightGBM Model

In this paper, we combine a deep convolutional neural network and the LightGBM algorithm for the RUL estimation. The prognostic structure is shown in Fig. 4. The features of aero-engine data are extracted by convolutional layer of DCNN. Then LightGBM further learns the information of the output of the flatten layer to complete the prediction.

The detail of the forecasting process is given as follows:

- (1) Data preprocessing. Exploratory data analysis is used to select sensor signals with significant changes for the RUL prediction. Sliding window technique is utilized to construct time sequence characteristics. The training dataset, the testing dataset, and the RUL labels are generated after data normalization.
- (2) Model training. The entire training process can be divided into two stages. The first stage is the DCNN training. DCNN consists of five convolutional layers, a flatten layer, a fully connected layer and an output layer. The goal of the DCNN training is to predict the RUL, and the training loss function is shown in Eq. (5). In the second stage, the output layer and the full connection layer are removed from the DCNN model trained in the first stage, and the output of the training data is extracted from the flatten layer as the transformed features. Then, the features extracted by DCNN are used to train LightGBM. The loss function is the root mean square error (RMSE) in the second stage. It can be considered that DCNN and LightGBM are the front and the back ends of the entire model, respectively. The deep features are extracted from the original data of the aero-engines by using DCNN. Furthermore, the RUL of the aero-engines is predicted by LightGBM based on these characteristics. The two parts of training are completely independent.

- (3) RUL estimating. The degradation features of engines are extracted from the output of the flatten layer based on the test data by the trained DCNN model. LightGBM is utilized aiming at the features to obtain the final predicted values of the RUL. Finally, the results are evaluated.

4 Experiment Studies

4.1 Aero-engine Dataset

This paper selects C-MAPSS datasets provided by NASA to verify the effectiveness of the above method. The C-MAPSS are widely used in prognostic studies, which contains four sub-datasets of aero-engines under different operating conditions and failure modes. Each sub-dataset contains training set, testing set and testing RUL values, and it is consist of 21 sensors and 3 operation settings [22]. Each engine unit has varying degrees of wear. Over time, the engine units begin to degrade, until they reach the system failure which is described as an unhealth time cycle. The sensor records in the testing set are terminated before the system failure. The purpose of the experiment is to predict the RUL of each engine unit in the testing set. The dynamic characteristics of aero-engine operating data under different operating conditions are significantly different, which leads to different network structures for extracting features. The structure of the proposed DCNN in this paper is designed for the prediction of the RUL of aero-engines under a single operating condition. Therefore, this paper utilizes the data sets FD001 and FD003 obtained under a single operating condition of the aero-engines for experimental analysis. FD001 and FD003 are composed of 100 training samples and 100 test samples, respectively.

Fig. 4 The forecasting process of DCNN-LightGBM model

4.2 Data Analysis and Preprocessing

In this experiment, 14 sensor signals with a significant trend are selected (2,3,4,7,8,9,11,12,13,14,15,17,20 and 21) and the irregular or unchanged sensor signals are eliminated. The range of each feature is normalized to [0,1] by using min–max normalization.

$$x_{norm}^{i,j} = \frac{x^{i,j} - x_{min}^j}{x_{max}^j - x_{min}^j}, \quad (11)$$

where $x^{i,j}$ is the i -th measuring point of the j -th sensor. $x_{norm}^{i,j}$ is the $x^{i,j}$ normalized result. x_{max}^j and x_{min}^j mean the maximum and minimum values of the j -th sensor, respectively.

Because the time series contains more information than a single point, this paper adopts a time window to use the multivariable temporal information, as the literature 20, 23 and 24 did. The method of selecting the length of the time window will be given in detail in Sect. 4.4. At each time step, all the historical data in the time window are collected to form a high-dimensional vector of length $L_s \times 14$ as the input data.

The normalized results about the 14 signals of the first engine in the FD001 and FD003 datasets are shown in Fig. 5. The operating data will show a significant abnormal trend due to the degradation of the aero-engines. At the initial stage of engine operation, the variation trend of the engine operation is not obvious and cannot provide effective information for the RUL prediction, as shown in the 0–75 time cycles in Fig. 5a. Therefore, the piecewise linear function has been adopted for the RUL target label. Figure 6. shows the piecewise RUL of the first engine unit

Fig. 5 Normalized results of the 14 signals of the first engine in the FD001 and FD003 datasets, respectively

Fig. 6 Piecewise RUL of the first engine in the FD001 and FD003 datasets, respectively

in the FD001 and FD003 datasets, respectively. We set the RUL value in the early stage of degradation to a fixed value of 125 as an upper limit, as the literature 23 and 25 did. This threshold is denoted by R_{early} . The effectiveness of piecewise linear function on this forecasting problem has been confirmed in the literature 13,15,20 and 26. The processed label values are smoothed.

4.3 Performance Metrics

In this paper, two metrics are used to evaluate prognostic performance [12], i.e., scoring function (Score) and the root mean square error (RMSE). The scoring function is widely used in the International Conference on Prognostics and Health Management Data Challenge. The formulation of Score is defined as:

$$\text{Score} = \begin{cases} \sum_{i=1}^N (e^{-\frac{d_i}{13}} - 1), & d_i \leq 0 \\ \sum_{i=1}^N (e^{\frac{d_i}{10}} - 1), & d_i > 0 \end{cases}, \quad (12)$$

where $d_i = RUL'_i - RUL_i$, that is the error between the predicted value and the true value of the i -th testing data sample, and N is the number of engines in the test set. Score penalizes late predictions more than early predictions since late predictions may cause serious accidents. Another metric is RMSE, which measures the average distance between the predicted values and the actual values, RMSE is presented as:

Fig. 7 Comparison between the two evaluation metrics

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N d_i^2}. \quad (13)$$

Figure 7 shows the relationship between Score and RMSE with respect to different error values.

4.4 Result Analysis

The processor used in the experiments is Intel(R) Core (TM) i7-8565U, 8 GB memory, Microsoft Windows 10 64 bit. The python version is 3.6.

4.4.1 Model Parameters and Training Results

The time window size is an important factor affecting the prediction accuracy of the proposed method. Figure 8 shows the effect of the time window size on the model performance. The prediction results of the RUL are affected by the amount of historical information. As shown in the Fig. 8, increasing the time window size can improve the prediction accuracy of the RUL of the engine. Note that the selected time window is determined by the number of the shortest cycle of the engine test set. Therefore, the time window sizes L_s of the FD001 and the FD003 data sets are 31 and 38, respectively. Furthermore, we train the DCNN-LightGBM model separately for 10 times to exclude the effects of random disturbances and take the average of the results. The key parameters of the proposed model are summarized in Table 2.

Figure 9 shows the RUL prediction results of 100 testing engine units in descending order. It can be observed that the predicted values of DCNN-LightGBM are closer to the actual values than that of DCNN. The prediction results of the proposed method are more accurate along with the engine degeneration. This is because the model can extract more failure features from the sensor data with increasing

Fig. 8 The effect of the time window size on the model performance for the DCNN training process on the FD001 dataset

degradation. The security of the system can be improved by accurately predicting the RUL near the stage of the engine failures.

4.4.2 Comparing with Other Popular Method

To verify the superiority of the proposed model, the methods including XGBoost, RNN, Deep Neural Network (DNN), DCNN and LightGBM are used to predict the RUL. Like DCNN-LightGBM, all comparison models are independently trained ten times. The prognostic results compared with different methods are presented in Table 3. DCNN-LightGBM has an excellent performance in two metrics. Although LightGBM improves the accuracy based on XGBoost, it has lower accuracy than DCNN-LightGBM. This is because DCNN can learn advanced features from original features through complex network calculations. These advanced features contain more concentrated effective information and are low dimensional. And many of these features cannot be constructed manually. So these features can achieve a better fitting effect when used on LightGBM. Therefore, the improved method based on DCNN achieves higher accuracy.

4.4.3 Comparing with Other Popular Method

To verify the superiority of the proposed model, the methods including XGBoost, RNN, Deep Neural Network (DNN), DCNN and LightGBM are used to predict the RUL. Like DCNN-LightGBM, all comparison models are independently trained 10 times.

The prognostic results compared with different methods are presented in Table 3. DCNN-LightGBM has an excellent performance in two metrics. Although LightGBM improves the accuracy based on XGBoost, it.

Table 2 The parameter values of different models

Model	Parameter	Description	Value
DCNN	Batch size	Number of samples per gradient update	512
	Dropout size	Fraction of the input units to drop	0.5
	Epoch number	Number of epochs to train the model	200
	Activation	Activation function to use	ReLU
LightGBM	Num leaves	The maximum number of leaves on a tree	300
	Learning rate	The learning parameter controls the magnitude of this change in the estimates	0.08
	Maximum depth	The maximum depth of a tree	12
	Num_iterations	The number of trees	300
	Subsample	Subsample ratio of the training instance	0.8

Fig. 9 Sorted prediction for the 100 testing engine units

has lower accuracy than DCNN-LightGBM. This is because DCNN can learn advanced features from original features through complex network calculations. These advanced features contain more concentrated effective information and are low dimensional. And many of these

Table 3 Performance comparisons of different methods on the FD001 and FD003 test sets

Method	FD001	FD003
XGBoost		
RMSE	15.32	15.87
Score	356.9	388.8
RNN		
RMSE	15.06	14.39
Score	348.9	326.7
DNN		
RMSE	14.91	14.98
Score	328.2	368.9
DCNN		
RMSE	14.83	15.68
Score	290.4	307.0
LightGBM		
RMSE	14.24	14.96
Score	273.6	342.6
DCNN-LightGBM		
RMSE	12.79	13.21
Score	232.0	277.8

features cannot be constructed manually. So these features can achieve a better fitting effect when used on LightGBM. Therefore, the improved method based on DCNN achieves higher accuracy.

4.4.4 Comparing with Related Works

Table 4 presents the research results of the current commonly used methods on the FD001 and FD003 data sets of C-MAPSS. Compared with traditional machine learning methods such as SVR and Random forest, deep learning method achieves better results in both Score and RMSE. Combined deep learning method such as Autoencoder-BLSTM has higher accuracy than the Deep LSTM which is the traditional deep learning method. Gradient boosting

Table 4 Performance comparisons with the commonly used methods

Model	FD001	FD003
SVR [27]		
RMSE	20.96	21.05
Score	1382	1598
Random forest [28]		
RMSE	17.91	20.27
Score	497	711.1
MODBNE [28]		
RMSE	15.04	12.51
Score	334.2	421.9
Deep LSTM [12]		
RMSE	16.14	16.18
Score	338	852.0
Gradient boosting [20]		
RMSE	15.67	–
Score	474	–
ANN-EN [24]		
RMSE	14.39	15.42
Score	337	533
Autoencoder-BLSTM [29]		
RMSE	13.63	–
Score	261	–
CNN-XGB [30]		
RMSE	12.61	13.01
Score	224.73	279.36
DCNN-LightGBM		
RMSE	12.79	13.21
Score	232.0	277.8

also has a good performance in the RUL estimation. In addition, Table 4 shows that both the proposed algorithm and the CNN-XGB algorithm can provide accurate prediction results for the RUL of aero-engines. The CNN-XGB algorithm takes the average of the prediction results provided by CNN and XGB as the final result. The nature of the algorithm does not propose structural innovations for CNN and XGB. The proposed algorithm is different from the CNN-XGB. As a new prediction method, the algorithm obtained by the combination of DCNN and LightGBM can integrate the advantage of DCNN to extract degradation features and the advantage of LightGBM to obtain the final RUL prediction values.

5 Conclusion

In this paper, we propose to use the model combining DCNN and LightGBM to predict the RUL of the aero-engine, and confirm the excellent performances of this model on the C-MAPSS data. The role of DCNN is to extract deep

features, and LightGBM is used to complete the prediction of the RUL. Comparing the scores of different models, we can see that the ensemble learning model has better prediction accuracy than other models. As the degree of degradation increasing, the prediction results are more accurate.

While the proposed method gets good experiment results, future architecture optimization is necessary. According to the literature 31, we will further optimize the model structure and hyperparameters to reduce training time and computational load. In future work, the proposed method will be applied for the RUL prediction of aero-engines with different operation conditions. When the operation conditions are more complex, the RUL prediction is more challenging and this kind of problems deserve further studies.

Acknowledgements We are grateful to our colleagues and the reviewers for their suggestions.

Funding This work was supported by the AEAC Advanced Jet Propulsion Creativity Center Foundation under Grant HKCX2020-02-029, the Basic Research Program of Science and Technology of Shenzhen under Grant JCYJ20190809163009630, the Natural Science Foundation of Fujian under Grants 2020J01052 and 2020J01054, and the Natural Science Foundation of Shanghai under Grant 20ZR1463300.

Declarations

Conflict of interest Authors have no conflict of interest to declare.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Ren, Z., Si, X., Hu, C., et al.: Remaining useful life prediction method for engine combining multi-sensors data. *Acta Aeronautica et Astronautica Sinica* **40**(12), 134–145 (2019)
2. Q. Wang, S. Zheng, A. Farahat, et al. Remaining useful life estimation using functional data analysis, arXiv 1904.06442, 2019.
3. G. Zhao, S. Wu, and H. Rong. A multi-source statistics data-driven method for remaining useful life prediction of aircraft engine, *Hsi-An Chiao Tung Ta Hsueh/Journal of Xi'an Jiaotong University*, vol. 51, no. 11, pp.150–155 and 172, 2017.
4. Zhang, Z.X., Si, X.S., Hu, C.H., Lei, Y.G.: Degradation data analysis and remaining useful life estimation: a review on Wiener-process-based methods. *Eur. J. Oper. Res.* **271**(3), 775–796 (2018)
5. Lee, J., Wu, F., Zhao, W., et al.: Prognostics and health management design for rotary machinery systems—reviews, methodology

- and applications. *Mech. Syst. Signal Process.* **42**(1–2), 314–334 (2014)
6. El-Thalji, I., Jantunen, E.: A summary of fault modelling and predictive health monitoring of rolling element bearings. *Mech. Syst. Signal Process.* **60–61**, 252–272 (2015)
 7. A. Rahman, A. Hossain, Z. A. M., and M. Rashid, Fuzzy knowledge-based model for prediction of traction force of an electric golf car, *Journal of Terramechanics*, vol. 49, no. 1, pp. 13–25, 2012.
 8. M. Xia, T. Li, T. X. Shu, J. F. Wan, Silva, D. C. W. Silva, Z. R. Wang, A two-stage approach for the remaining useful life prediction of bearings using deep neural networks, *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3703–3711, 2019.
 9. Khelif, R., Chebelmorello, B., Malinowski, S., et al.: Direct remaining useful life estimation based on support vector regression. *IEEE Trans. Industr. Electron.* **64**(3), 2276–2285 (2017)
 10. Tang, D.Y., Cao, J.R., Yu, J.S.: Remaining useful life prediction for engineering systems under dynamic operational conditions: asemi-Markov decision process-based approach. *Chin. J. Aeronaut.* **32**(3), 627–638 (2019)
 11. Malhi, A., Yan, R., Gao, R.X.: Prognosis of defect propagation based on recurrent neural networks. *IEEE Trans Instrum Meas* **60**(3), 703–711 (2011)
 12. S. Zheng, K. Ristovski, A. K. Farahat, et al., Long short-term memory network for remaining useful life estimation, in *Proc. ICPHM*, Dallas, TX, USA, 2017, pp. 88–95.
 13. X. Li, Q. Ding, and J. Q. Sun. Remaining useful life estimation in prognostics using deep convolution neural networks, *Reliability Engineering & System Safety*, pp. 1–11, 2018.
 14. Li, W., Ding, S., Chen, Y., Yang, S.: Heterogeneous ensemble for default prediction of peer-to-peer lending in China. *IEEE Access* **6**, 54396–54406 (2018)
 15. T. Chen and C. Guestrin, Xgboost: A scalable tree boosting system, in *SIGKDD. ACM.*, pp. 785–794, 2016.
 16. G. Ke, et al., LightGBM: A highly efficient gradient boosting decision tree, in *Proc. Adv. Neural Inf. Process. Syst.*, pp. 3146–3154, 2017.
 17. P. Li, J. Li, and G. Wang, Application of convolutional neural network in natural language processing, *Int. Comput. Conf. Wavelet Act. Media Technol. Inf. Process., ICCWAMTIP*, Chengdu, Sichuan, China, 2018, pp. 120–122.
 18. A. Krizhevsky, I. Sutskever, and G. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, in *Adv. neural inf. proces. syst.*, Lake Tahoe, NV, USA, 2012, pp. 1097–1105.
 19. K. He, X. Zhang, S. Ren, et al., Delving deep into rectifiers: surpassing human-Level performance on imageNet classification[J]. *CVPR*, 2015.
 20. F. LI, et al., A light gradient boosting machine for remaining useful life estimation of aircraft engines, *IEEE Conf Intell Transport Syst Proc., ITSC*, Maui, HI, USA, 2018, pp. 3562–3567.
 21. K. Zhou, Y. Hu, H. Pan, et al., Fast prediction of reservoir permeability based on embedded feature selection and LightGBM using direct logging data, *Measurement Science and Technology*, vol. 31, no. 4, 2020.
 22. A. Saxena, G. Kai, D. Simon, and N. Eklund, Damage propagation modeling for aircraft engine run-to-failure simulation, in *IEEE Int. Conf. Prognostics Health Manage.*, pp. 1–9, 2008.
 23. Ramasso, E.: Investigating computational geometry for failure prognostics. *Int. J. Prognostics Health Manage.* **5**, 1–18 (2014)
 24. Laredo, D., Chen, Z., Schütze, O., et al.: A neural network-evolutionary computational framework for remaining useful life estimation of mechanical systems[J]. *Neural Netw.* **116**, 178–187 (2019)
 25. Zhao, Z., Liang, B., Wang, X., Lu, W.: Remaining useful life prediction of aircraft engine based on degradation pattern learning. *Rel. Eng. Syst. Safety* **164**, 74–83 (2017)
 26. G. S. Babu, P. Zhao, and X. L. Li, Deep convolutional neural network based regression approach for estimation of remaining useful life, in *Int. Conf. Database Syst Advan. Applications*, pp. 214–228, 2016.
 27. C Louen, S X Ding, and C Kandler. A new framework for remaining useful life estimation using Support Vector Machine classifier. *IEEE*, 2013.
 28. Zhang, C., Lim, P., Qin, A., et al.: Multi-objective deep belief networks ensemble for remaining useful life estimation in prognostics. *IEEE Trans Neural Netw Learn Syst.* **28**(10), 2306–2318 (2017)
 29. Song, T., Xia, T.B., et al.: Remaining useful life of turbofan engine based on Autoencoder-BLSTM. *Comput. Integr. Manuf. Syst.* **25**(7), 1611–1619 (2019)
 30. X. Zhang, P. Xiao, Y. Yang, et al. Remaining Useful Life Estimation Using CNN-XGB with Extended Time Window[J]. *IEEE Access*, 2019, PP (99):1–1.
 31. J. Li, D. He, A Bayesian Optimization AdaBN-DCNN Method with Self-optimized Structure and Hyperparameters for Domain Adaptation Remaining Useful Life Prediction[J]. *IEEE Access*, 2020, PP (99):1–1.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.