Tech Science Press

Ensemble Recurrent Neural Network-Based Residual Useful Life Prognostics of Aircraft Engines

Jun Wu[1,*], Kui Hu[1], Yiwei Cheng[2], Ji Wang[1], Chao Deng[2,*] and Yuanhan Wang[3]

[1]School of Naval Architecture and Ocean Engineering, Huazhong University of Science and Technology, Wuhan, 430074, China.

[2]School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan, 430074, China.

[3]China Electronic Product Reliability and Environmental Testing Research Institute, Guangzhou, 510610, China.

[*]Corresponding Author: Jun Wu. Email: wuj@hust.edu.cn; dengchao@hust.edu.cn.

Abstract: Residual useful life (RUL) prediction is a key issue for improving efficiency of aircraft engines and reducing their maintenance cost. Owing to various failure mechanism and operating environment, the application of classical models in RUL prediction of aircraft engines is fairly difficult. In this study, a novel RUL prognostics method based on using ensemble recurrent neural network to process massive sensor data is proposed. First of all, sensor data obtained from the aircraft engines are preprocessed to eliminate singular values, reduce random fluctuation and preserve degradation trend of the raw sensor data. Secondly, three kinds of recurrent neural networks (RNN), including ordinary RNN, long short-term memory (LSTM), and gated recurrent unit (GRU), are individually constructed. Thirdly, ensemble learning mechanism is designed to merge the above RNNs for producing a more accurate RUL prediction. The effectiveness of the proposed method is validated using two characteristically different turbofan engine datasets. Experimental results show a competitive performance of the proposed method in comparison with typical methods reported in literatures.

Keywords: Aircraft engines; residual useful life prediction; health monitoring; neural networks; ensemble learning

1 Introduction

With the application of a large number of highly sophisticated engineering equipment, residual useful life (RUL) prediction becomes an indispensable technique to improve the reliability of equipment and make accurate maintenance and utilization strategy [1-3]. Based on the analysis of historical sensor data, the method might deeply mine the degradation process of equipment performance state, and achieve the accurate prediction of its RUL. With the continuous development of intelligent algorithms, more and more new methods have been applied to the field of RUL prediction, and many good results have also been achieved. For example, Wang used an improved relevance vector machine (RVM) approach to accurately describe the degradation process from fault to failure [4]. Xu proposed a hybrid degradation model, which combines multiscale characteristic analysis (MCA) with modified Gaussian process regression (GPR), to predict the remaining useful life of a controller under various working conditions [5]. Li used a modified health index based hierarchical gated recurrent unit network for rolling bearing health prognosis [6]. However, due to different running conditions and operation environments, the health condition and RUL of each equipment might present differences inevitably [7-9]. In addition, environmental noise interference might affect the quality of monitoring data. Thus, the feasibility and accuracy of RUL prediction are still a problem to be solved.

In order to solve the problem mentioned above, recurrent neural networks, which have proved to be a good performance in dealing with time series problems, has been widely used in the field of life prediction [10-14]. Guo proposed a recurrent neural network based health indicator (RNN-HI) for RUL prediction of bearings [15]. Lei proposed a deep learning network, namely multi-scale dense gate recurrent unit network

DOI:10.32604/sdhm.2019.05571

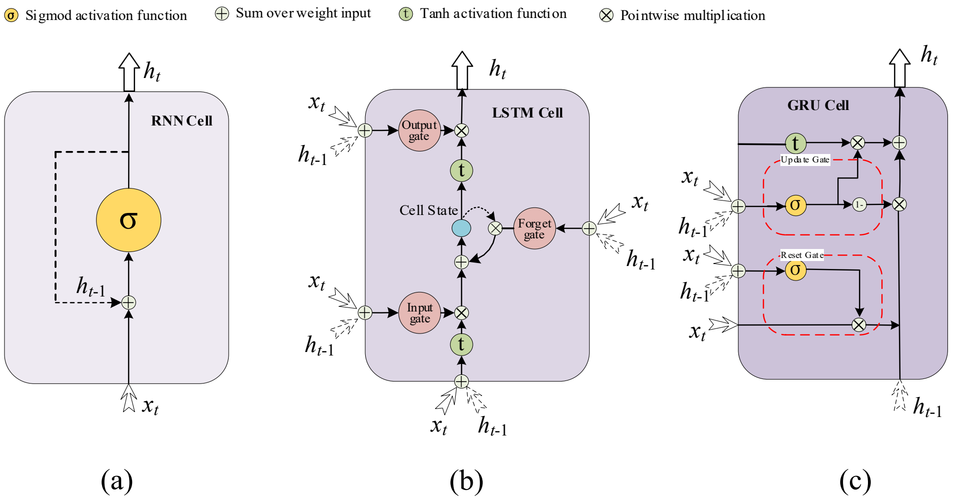
(MDGRU), which is composed of the feature layers initialized by pre-trained restricted Boltzmann machine (RBM) network, multi-scale layers, skip gate recurrent unit layers and dense layers [16]. Ahmed used an end-to-end deep framework based on convolutional and LSTM to deal with RUL estimation problems [17]. Zhang apply the LSTM networks to set up an architecture that is specialized in discovering the underlying patterns embedded in time series to track the system degradation and predict the RUL [18].

In this paper, a novel RUL prediction method is proposed by integrating several RNNs to automatically learn the historical data of the aircraft engine and predict its RUL. Three kinds of RNNs are built separately. Furthermore, a new ensemble method is applied to synthesize the established models.

The rest of this paper is organized as follows. In Section 2, the concept of recurrent neural networks is descripted. In Section 3, the framework of proposed RUL prediction method is presented in detail. In Section 4, a practical experimental study is performed to demonstrate the effectiveness of the proposed method. Conclusions and future work of the research are discussed in Section 5.

2 Concepts of RNN

RNN is a class of artificial neural network and is mainly used to deal with sequence problems [11]. A large number of studies have shown that RNN have a very strong ability to process data sequences with strong correlation. Herein, three kinds of RNN model architectures are shown in Fig. 1.



(a) (b) (c)

Figure 1: Three types of RNNs

2.1 Ordinary RNN

Ordinary RNN has a stack of non-linear elements in which at least one connection between the elements forms a directed loop [11]. Its main structure is shown in Fig. 1(a). It can be seen that the input of the main loop structure of the RNN is not only from the input layer, but also an output from the hidden layers at last time. At the same time, the state of the current moment will also be transmitted to the following recurrent state. The equation expression of the network is as follows:

$$H_t = f(H_{t-1}, x_t, \theta) \tag{1}$$

where f is the mapping relation, which is considered as the activation function, θ is parameter, H_t is the output at current time, H_{t-1} is the output of the previous time, and x_t is the input of the current time. However, due to the limitations of its recurrent unit structure, Ordinary RNN is prone to the appear gradient vanishing problems during the backpropagation of model training.

2.2 LSTM

Long short term memory (LSTM) neural networks, proposed by Hochreiter and Schmidhuber in 1997 [19], may remember long-term information through special designed cell structure. Unlike the relatively simple repetitive units in RNN, LSTM can process information by using three well-designed gate structures in LSTM cell. The main structure of LSTM cell is shown in Fig. 1(b).

The following formulas are used to describe the structure of the LSTM cells. First of all, the input gate is used to process the input of the current sequence position, determine the information that needs to be updated, and update the cell status. It can be mathematically expressed as:

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

where i_t is output of input gate, W_i and b_i represents the weight coefficients and bias of input gate and σ denotes the sigmoid activation function. Next, Forget gate controls the content of cell state at the last moment that needs to be discarded. Through sigmoid activation function, the information which needs to be removed or retained from the content of cell state at the previous moment is finally decided. It can be expressed as:

$$f_t = \sigma\left(W_f [h_{t-1}, x_t] + b_f\right) \tag{3}$$

where f_t is output of forget gate, W_f are weight coefficients of the input data x_t and previous output h_{t-1} of the LSTM cell, b_f represents the bias. At the same time, the cell status is updated, It can be expressed as:

$$\widetilde{C_t} = tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{4}$$

where $\widetilde{C_t}$ is the cell states of time t, which is used to store important information. W_C and b_c are the weight coefficients and bias of input node. Finally, the output gate determines what to output based on the content saved by the cell state. It can be expressed as:

$$o_t = (W_o [h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t * tanh(C_t) \tag{6}$$

where h_t is the final output, o_t is a value between 0 and 1 which determines how much information is output in cell state.W_o and b_o are the weight coefficients and bias of input node.

2.3 GRU

GRU is essentially a variant of LSTM network. It is proposed by Kyunghyun Cho in 2014 [20]. In general, while keeping the main characteristics of LSTM, the GRU becomes more simple. Its main structure is shown in Fig. 1(c):

The main characteristic of this network is the merging of cell state and hidden layer state. It utilizes reset gate and update gate to process and output information. The reset gate is used to determine the weight of the hidden state h_{t-1} at the last time point in the new hidden state h_t, which is expressed in Eq. (7) and Eq. (8):

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \tag{7}$$

$$\widetilde{h_t} = tanh(W \cdot [r_t * h_{t-1}, x_t]) \tag{8}$$

where $\widetilde{h_t}$ is output of reset gate, which represents the hidden layer state after reset, r_t is the proportion of h_{t-1} in $\widetilde{h_t}$, W_r are the weight coefficients of input node.

The update gate is used to determine the current state of the hidden layer of h_t in the new state of hidden layer $\widetilde{h}_t$ held by weight z_t, which is expressed in Eq. (9) and Eq. (10):

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \tag{9}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \widetilde{h_t} \tag{10}$$

where h_t is the final output of update gate, z_t is the proportion of $\widetilde{h_t}$ in h_t, W_z are the weight coefficients of input node.

3 Proposed Ensemble Model for RUL Prediction

Although single RNN model has good performance, the generalization of the single RNN rarely performs well in a variety of applications [19]. The integration of multiple models and the appropriate combination of the multiple models' output have proved to be very effective for improving the generalization. In this paper, an ensemble model is proposed for predicting the RUL of aircraft engine by integrating multiple RNNs. In order to ensure the diversity of sub-learners and sufficient prediction accuracy, three different RNNs are adopted for integration.

However, it is not easy to set up a good ensemble. The most critical task is to obtain sub-learners with good individual performance and little correlation among the training errors. Previous literature has pointed out that the ideal integration should consist of base learners, which have good diversity and can produce smaller errors [21-24].

The widely used ensemble mechanism include bagging, boosting and stacking [25-27]. However, stacking has a better ensemble performance than other mechanisms, for its multiple sub-learners run in parallel [28-31]. In this paper, the stacking mechanism of ensemble learning is adopted to fuse different neural network models.

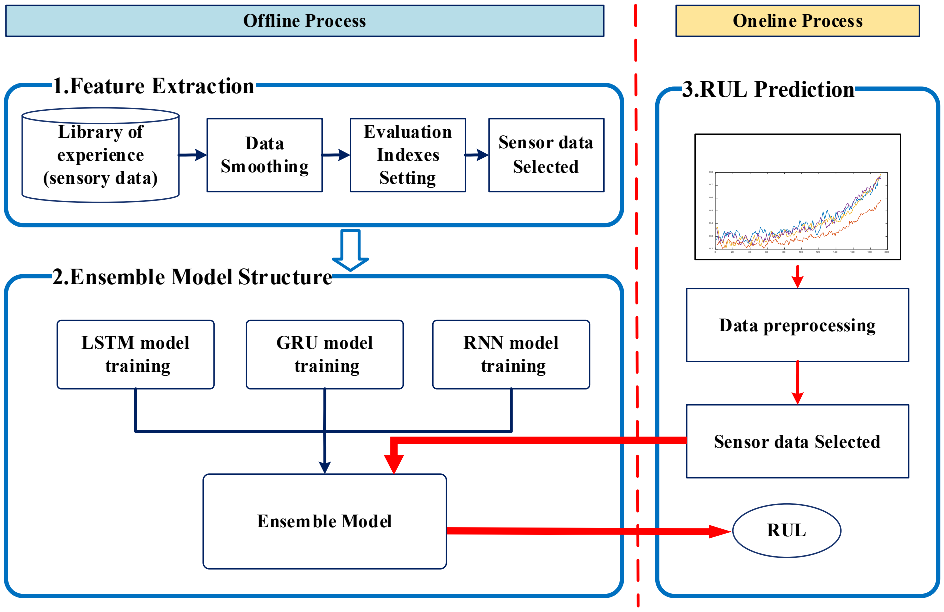


Figure 2: Schematic diagram for the proposed prediction method

The RUL prediction system proposed in this paper is shown as Fig. 2. It contains the process of modeling the history data and using online data to predict RUL. It is mainly divided into the following three steps:

Step 1: Feature Extraction

In the first step, the sensor data is preprocessed. The exponential smoothing algorithm is used to remove noise from original sensor data. By comparing the monotony and the correlation, the suitable sensors are selected.

Step 2: Model Structure

In the second step, model structure is built, which include three main processes. Initially, three neural network models are trained using that historical sensor data selected out by step 1. Then, the prepared data is used to verify the network models. Finally, the three verified network models are integrated by ensemble learning algorithms, and an ensemble model is obtained.

Step 3: Online RUL Prediction

After obtaining the model that conforms to the precision requirement, the online sensor data that have been preprocessed and filtered are inputted into the trained RNN model. Finally, the RUL prediction of the aircraft engine is acquired.

4 Experimental Study

4.1 Experimental Setup and Data Description

To verify the effectiveness of the proposed method, the data set of NASA turbofan engine is used in this paper. This data set is generated by a simulation model of turbofan engines built on the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS), which was used as challenge data for the Prognostics and Health Management (PHM) data competition at PHM'08 [32]. Fig. 3 shows the main components of the turbofan engine simulation model. It mainly contains fan, low pressure compressor (LPC), high pressure compressor (HPC), high pressure turbine (HPT) and low pressure turbine (LPT). During the test, the turbofan engine in a healthy state was running until there was a failure that caused the system to fail. In addition, input parameters such as speed and pressure are changed to simulate different faults and degradation processes of the various rotating components of the turbofan engine.

As shown in Tab. 1, there are four different operating conditions, resulting in four datasets: FD001~FD004. Each dataset includes training and testing subsets. The training set contains all the sequences that run until it fails, while the testing set only contains multiple sensor data of the engine which stop running before failure. In this paper, the dataset of FD001 is adopted to test the ensemble RNN model presented above. Tab. 2 shows all the 21 sensor signal categories in the turbofan engine dataset, including temperature, pressure, speed, etc.

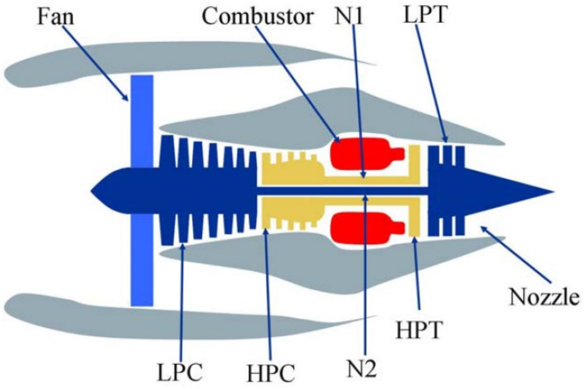


Figure 3: Structure of turbofan engine in C-MAPSS simulation

Table 1: Description of C-MAPSS dataset

Sub-datasets	Training Engines	Testing Engines	Fault Modes	Conditions
FD001	100	100	1	1
FD002	260	259	1	6
FD003	100	100	2	1
FD004	248	248	2	6

Table 2: Sensor data type

Index	Symbol	Description	Units
1	T2	Total temperature at fan inlet	°R
2	T24	Total temperature at LPC inlet	°R
3	T30	Total temperature at HPC inlet	°R
4	T50	Total temperature at LPT inlet	°R
5	P2	Pressure at fan inlet	psia
6	P15	Total Pressure in bypass-duct	psia
7	P30	Total Pressure at HPC outlet	psia
8	Nf	Physical fan speed	rpm
9	Nc	Physical core speed	rpm
10	epr	Engine pressure ratio (P50/P2)	--
11	Ps30	Static pressure at HPC outlet	psia
12	phi	Ratio of fuel flow to Ps30	pps/psi
13	NRf	Corrected fan speed	rpm
14	NRc	Corrected core speed	rpm
15	BPR	Bypass Ratio	--
16	farB	Burner fuel-air ratio	--
17	htBleed	Bleed Enthalpy	--
18	Nf_dmd	Demanded fan speed	rpm
19	PCNfR_dmd	Demanded corrected fan speed	rpm
20	W31	HPT coolant bleed	lbm/s
21	W32	LPT coolant bleed	lbm/s

4.2 Data Preprocessing and Sample Labeling

According to the method described in Section 2, the experimental data are preprocessed. Exponential algorithm is adopted to smooth the raw sensor signals. And, the window width is set to be 7 [33,34].

In order to control the scale of features in the same range, the selected sensor data need be normalized. In this experiment, min-max standardization is used to conduct linear transformation of the original data to make the data mapped between 0 and 1. The transformation function is descripted as:

$$x^* = \frac{x-x_{min}}{x_{max}-x_{min}} \tag{11}$$

where x_{max} is the maximum value of the selected data, and x_{min} is the minimum value.

Then the monotonic and correlation of each sensor are calculated respectively and the composite index selection criteria (SC) is obtained. Mon and Corr are expressed as:

$$\begin{cases} \text{Mon} = \frac{\left|\sum_k \delta\left(f_T(t+1)-f_T(t)\right)-\sum_k \delta\left(f_T(t)-f_T(t+1)\right)\right|}{K-1} \\ \text{Corr} = \frac{|(K\sum_k f_T(t)t-K\sum_k f_T(t)\sum_k t)|}{\sqrt{[K\sum_k f_T(t)^2-(\sum_k f_T(t))^2][K\sum_k t^2-(\sum_k t)^2]}} \end{cases} \tag{12}$$

where *K* is the total number of the sampling points and $(\cdot)$ is the sign function. $f_T(t)$ is trend value that can be obtain by the mean of upper and lower envelope of sensor data at time t.

Arrange them in order from small to large. As shown in the Fig. 4, since the SC value is greater than the threshold value, S2, S3, S4, S7, S8, S11, S12, S13, S15, S17 S20 and S21 are picked up to continue the next steps.

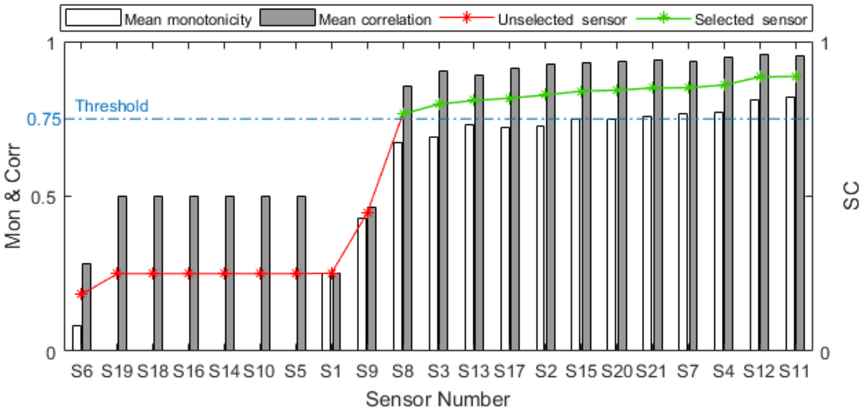


Figure 4: Mon, Corr and SC of 21 sensors in FD001

The final step is to label the life of each engine for the next model training. A large number of cases indicate that the label value has a significant impact on prognostic performance [35]. For this dataset, a piece-wise linear degradation model has proven to be appropriate and effective [36]. This model divides the running process of each engine into two stages. In the first stage, the engine is considered to run in a healthy state, so the label is a constant value. In the second stage, the engine begins to degenerate until it fails. The RUL label begins to degrade linearly. This paper is finally set to a constant value of 125 as the target label for the first stage. Fig. 5 shows the steps of data preprocessing.

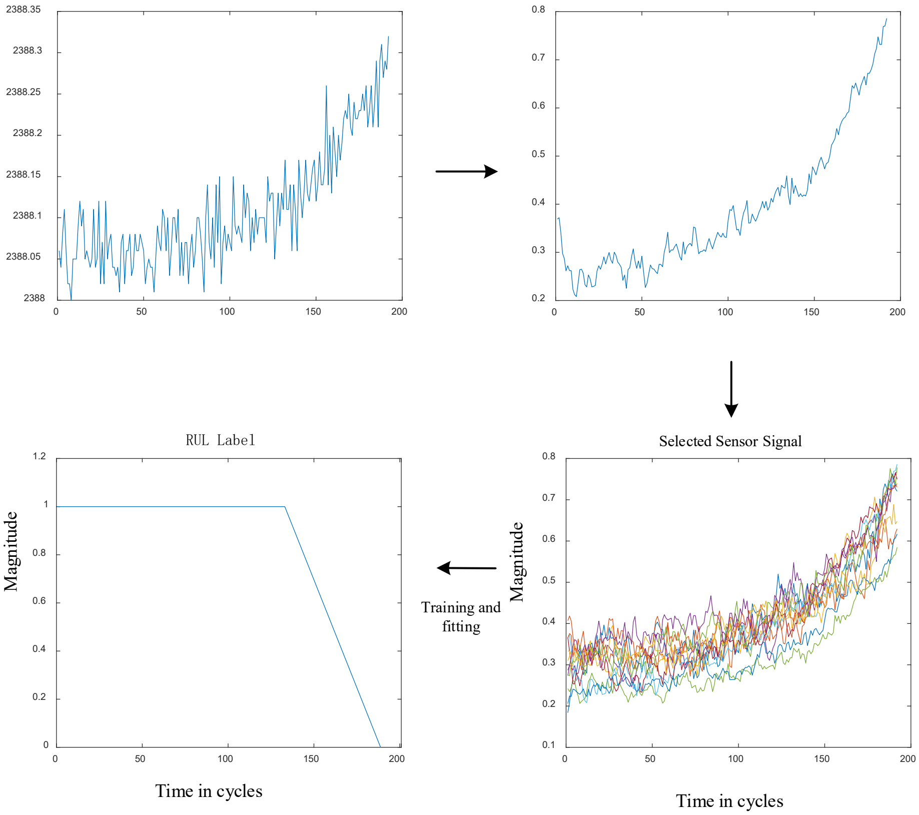


Figure 5: Data preprocessing steps

4.3 Performance Evaluation Metrics

Three evaluation indicators are adopted to assess the performance of the proposed method.

The first indicator is Score. It is employed by the international conference on prognostics and health management (PHM08) data challenge, and has been applied to experiments by many researchers [37]. Compared with early prediction, the scoring function tends to penalize late prediction since late prediction may have more serious consequences in practice. Besides, higher prediction errors are also penalized more severely. The scoring function is express as:

$$s_i = \begin{cases} \exp\left(-\frac{\widehat{R_\iota}-R_i}{13}\right)-1 & if\ \widehat{R_\iota}-R_i<0 \\ \exp\left(\frac{\widehat{R_\iota}-R_i}{10}\right)-1 & if\ \widehat{R_\iota}-R_i\geq 0 \end{cases} \tag{12}$$

$$\text{Score} = \sum_{i=1}^{n} s_i \tag{13}$$

where $\widehat{R_\iota}$ represents the predicted RUL and R_i indicates the real RUL, s_i represents the score of a single engine, n is the total number of testing engines. The sum of the scores of all engines is the final total score.

The second indicator to evaluate prediction accuracy is RMSE [38]. This measure has also been widely used although it penalizes both early and late predictions equally, which is expressed as:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\widehat{R_\iota}-R_i)^2} \tag{14}$$

Fig. 6 shows the function image of scoring function and RMSE.

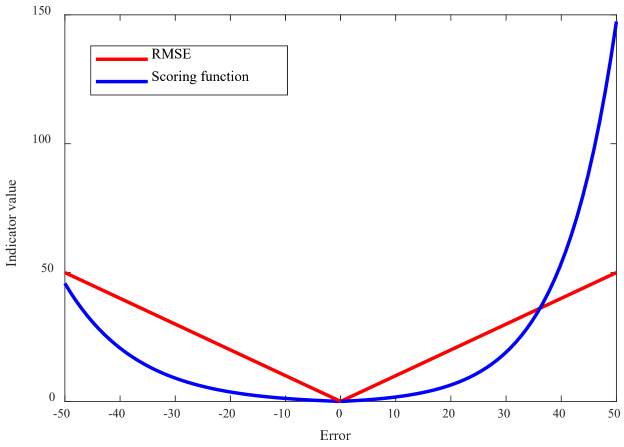


Figure 6: RMSE and Scoring function

The third evaluation index is RUL error range, which represents the error range of all RUL predictions compared with real values. Smaller RUL error range indicate higher accuracy and stability of the proposed ensemble model.

4.4 Results and Discussion

After the raw sensor data collected from 100 engines is preprocessed, these sensor data is selected to establish the training dataset. The 90 engines among them are randomly chosen for the ensemble model training, and the remaining 10 engines are used to verify the effectiveness of the trained ensemble model.

Next, based on the characteristics of the sensor data, a number of RNNs begin to be constructed. The constructed neural network model will be trained and verified. The neural network model having the best performance will be selected for the final integration.

In the training process of the neural network models, the number of neurons has a great impact on the performance of the model. In order to explore the optimal network structure, all three kinds of RNNs with the number of neurons set to 50 to 300 are trained. The validation engines are used to verify each model and RMSE are adopted for comparing model training results. Fig. 7 shows the training results of different RNNs. It can be seen from Fig. 7 that when the number of neurons in the three neural networks (RNN, LSTM, GRU) is 50, 150 and 200, respectively, the network model has the smallest RMSE, which means the best performance.

Fig. 8 shows the final verification result of the three kinds of neural networks. It can be seen from Fig. 8 that all of them have a good fitting ability to predict the degradation process of the engines, which means that the training of the model is satisfactory.

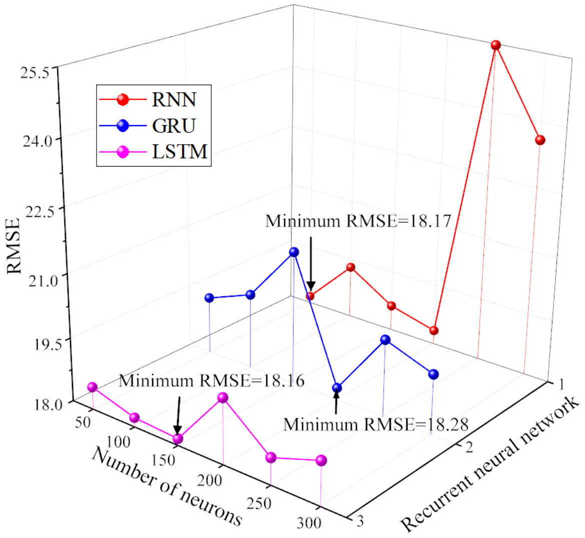


Figure 7: Training results of RNN with different neuron numbers

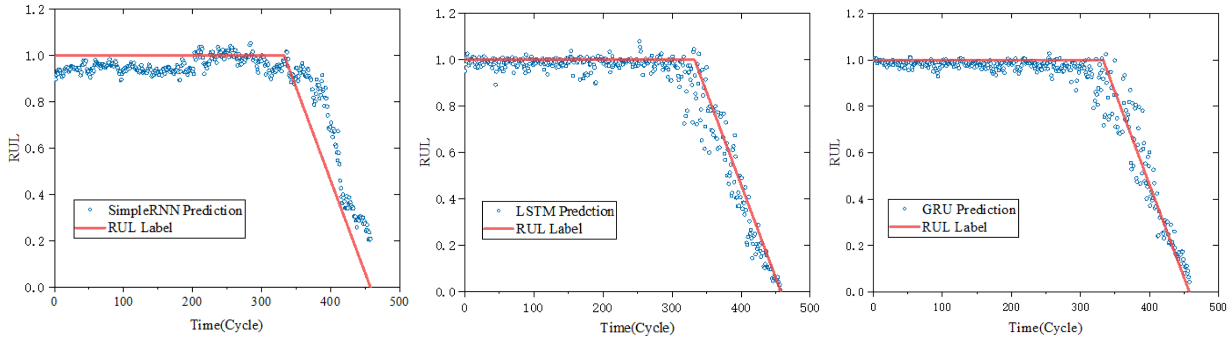


Figure 8: Prediction of validation engines. (a) RNN. (b) LSTM. (c) GRU

Then, the three RNN models are integrated using ensemble learning algorithm. Herein, the framework of stacking is used, and the RNN is relearned by random forest algorithm. The testing set of 100 engines is used to predict the RUL of the engines using trained ensemble model. Fig. 9 shows the comparison between the predicted results and the real RUL. Their correlation is up to 0.91, which shows the good prognostic performance of the ensemble model.

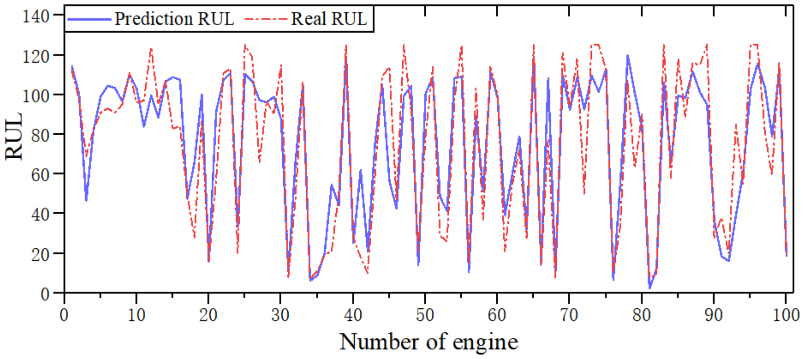


Figure 9: True RUL versus predicted RUL for 100 engines

Fig. 10 shows the box plot of the score for the ensemble model and the three kinds of RNNs. Compared with the three single network models, the score of the proposed ensemble model is more concentrated, which shows the better stability of RUL prediction.

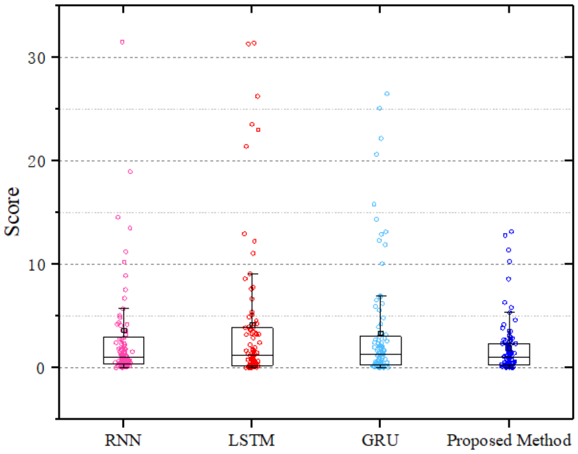


Figure 10: The boxplots of RUL error and RMSE of five models

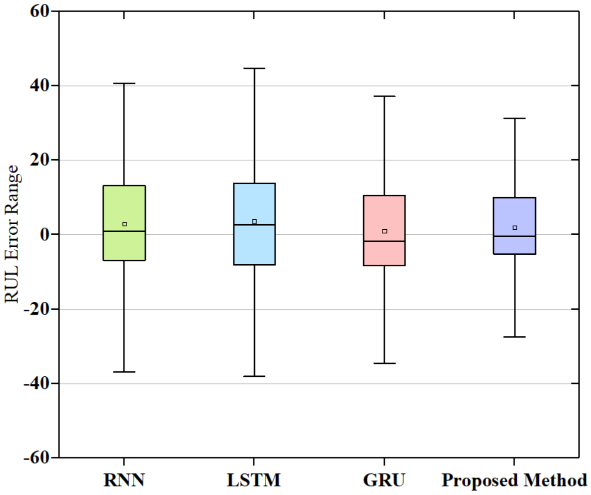


Figure 11: The boxplots of RUL error and RMSE of five models

Fig. 11 shows the box plot of the RUL error and the line chart of RMSE obtained by four models on the testing set. As can be seen from Fig. 11, the proposed method has smaller RUL error range and lower RMSE than the other three RNN. This shows that the model constructed by this method can learn and fit the engine's historical data better, and the result is more accurate and has less bias.

Tab. 3 shows the performance of models based on LSTM, GRU, RNN, and Ensemble model. As can be seen from Tab. 3 that all the evaluation indexes of the ensemble mode have been improved on the basis of the three recurrent networks, which means that the model successfully integrates the features extracted from the previous three networks and shows excellent life prediction performance.

Table 3: Comparison result

Indicator	Score	RMSE	RUL error range
LSTM Model	788.54	19.39	[-36,40]
GRU Model	838.99	19.64	[-38,44]
RNN Model	911.58	19.78	[-34, 39]
Ensemble Model	**699.99**	**18.82**	**[-29,38]**

5 Conclusion

In this paper, a new ensemble recurrent neural network-based RUL prediction method for aircraft engines is proposed. Multiple sensor data are preprocessed and selected to form the input of the ensemble RNN model. Then, three kinds of RNN models with different structures are constructed using LSTM, GRU and Ordinary RNN neurons, respectively. Next, the three RNN models are trained by the sensor data processed in the first step, and these models are combined with the stacking framework of ensemble learning. The random forest algorithm is used as the second-level meta-learner of the stacking framework. Finally, an ensemble recurrent neural network model which can utilize the features extracted by three recurrent neural networks is established.

The performance of the proposed RUL prediction method was evaluated by an experimental study of NASA's aeroengine dataset. The experimental results show that the prediction results of the ensemble model are better compared with the single current neural network model, which proves the superiority of the proposed method.

In the future, we will continue to explore the impacts of different neural network structures for the multiple sensor data's feature extraction. The comparative analysis will also be conducted on different ensemble methods to achieve higher prediction accuracy.

Acknowledgement: This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant Nos. 51875225 and 51605095, in part by the National Key Research and Development Program of China under the Grant No. 2018YFB1702302, and in part by the Key Research and Development Program of Guangdong Province under the Grant No. 2019B090916001.

References

1. Yin, S., Li, X., Gao, H., Kaynak, O. (2015). Data-based techniques focused on modern industry: an overview. *IEEE Transactions on Industrial Electronics, 62(1),* 657-667.
2. Malinowski, S., Chebel-Morello, B., Zerhouni, N. (2015). Remaining useful life estimation based on discriminating shapelet extraction. *Reliability Engineering & System Safety, 142,* 279-288.
3. Li, X., Ding, Q., Sun, J. Q. (2017). Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliability Engineering System Safety, 172.*
4. Wang, X. L., Jiang, B., Lu, N. Y. (2018). Adaptive relevant vector machine based RUL prediction under uncertain conditions. *ISA Transactions, 87,* 217-224.
5. Xu, D., Sui, S. B., Zhang, W., Xing, M., Chen, Y. et al. (2018). RUL prediction of electronic controller based on multiscale characteristic analysis. *Mechanical Systems and Signal Processing, 113,* 253-270.

6. Li, X., Jiang, H., Xiong, X. (2019). Rolling bearing health prognosis using a modified health index based hierarchical gated recurrent unit network. *Mechanism and Machine Theory, 133,* 229-249.

7. Wu, J., Su, Y., Cheng, Y., Shao, X., Deng, C. et al. (2018). Multi-sensor information fusion for remaining useful life prediction of machining tools by adaptive network based fuzzy inference system. *Applied Soft Computing, 68,* 13-23.

8. Gebraeel, N. Z., Lawley, M. A., Li, R., Ryan, J. K. (2005). Residual-life distributions from component degradation signals: a bayesian approach. *IIE Transactions, 37(6),* 543-557.

9. Wu, J., Wu, C., Lv, Y., Deng, C., Shao, X. (2017). Design a degradation condition monitoring system scheme for rolling bearing using emd and pca. *Industrial Management & Data Systems, 117(4),* 713-728.

10. Madhusudana, C. K., Gangadhar, N., Hemantha, K., Narenaranath, S. (2018). Use of discrete wavelet features and support vector machine for fault diagnosis of face milling tool. *Structural Durability & Health Monitoring, 12(2),* 111-127.

11. Cheng, Y., Zhu, H., Wu, J., Shao, X. (2019). Machine health monitoring using adaptive kernel spectral clustering and deep long short-term memory recurrent neural networks. *IEEE Transactions on Industrial Informatics, 15(2), 987-997.*

12. Alamelu, M. T. M., Jegadeeshwaran, R., Sugumaran, V. (2017). Brake fault diagnosis through machine learning approaches-a review. *Structural Durability & Health Monitoring, 11(1),* 43-67.

13. Manghai, T. M. A., Jegadeeshwaran, R. (2014). Feature-based vibration monitoring of a hydraulic brake system using machine learning. *Structural Durability & Health Monitoring, 11(2),* 149-167.

14. Wu, J., Wu, C. Y., Cao, S., Or, S. W., Deng, C. et al. (2019). Degradation data-driven time-to-failure prognostics approach for rolling element bearings in electrical machines. *IEEE Transactions on Industrial Electronics, 66(1),* 529-539.

15. Guo, L., Li, N., Jia, F., Lei, Y. G., Lin, J. (2017). A recurrent neural network based health indicator for remaining useful life prediction of bearings. *Neurocomputing, 240,* 98-109.

16. Lei, R., Cheng, X., Wang, X. (2018). Multi-scale dense gate recurrent unit networks for bearing remaining useful life prediction. *Future Generation Computer Systems, 94,* 601-609.

17. Ahmed, Z. H., Mohamed, T. (2018). Rolling element bearing remaining useful life estimation based on a convolutional long-short-term memory network. *Procedia Computer Science, 127,* 123-132.

18. Zhang, J., Wang, P., Yan. R., Gao, R. (2018). Long short-term memory for machine remaining life prediction. *Journal of Manufacturing Systems, 48,* 78-86.

19. Hochreiter, S., Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9(8),* 1735-1780.

20. Kyunghyun, C., Bart, M., Caglar, G., Dzmitry, B., Fethi, B. et al. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *Computer Science, 1,* 1724-1734.

21. Coop, R., Mishtal, A., Arel, I. (2013). Ensemble learning in fixed expansion layer networks for mitigating catastrophic forgetting. *IEEE Transactions on Neural Networks and Learning Systems, 24(10),* 1623-1634.

22. Fernández-Navarro, F., Gutiérrez, P. A., Hervás-Martánez, C., Yao, X. (2013). Negative correlation ensemble learning for ordinal regression. *IEEE Transactions on Neural Networks and Learning Systems, 24(11),* 1836-1849.

23. Hansen, L. K., Salamon, P. (1990). Neural network ensembles. *IEEE Transactions on Pattern Analysis & Machine Intelligence, 12(10),* 993-1001.

24. Krogh, A., Vedelsby, J. (1995). Neural network ensembles, cross validation, and active learning. *International Conference on Neural Information Processing Systems,* 231-238.

25. Dietterich, T. G. (1997). Machine-learning research. *AI Magazine, 18(4),* 97-136.

26. Opitz, D., Maclin, R. (1999). Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research, 11,* 169-198.

27. Chandra, A., Yao, X. (2006). Ensemble learning using multi-objective evolutionary algorithms. *Journal of Mathematical Modelling and Algorithms in Operations Research, 5(4),* 417-445.

28. Zhang, X., Tian, Y., Cheng, R., Jin, Y. (2015). An efficient approach to nondominated sorting for evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation, 19(2),* 201-213.

29. Gee, S. B., Tan, K. C., Shim, V. A., Pal, N. R. (2015). Online diversity assessment in evolutionary multiobjective optimization: a geometrical perspective. *IEEE Transactions on Evolutionary Computation, 19(4),* 542-559.

30. Chen, B., Zeng, W., Lin, Y., Zhang, D. (2016). A new local search-based multiobjective optimization algorithm. *IEEE Transactions on Evolutionary Computation, 19(1),* 50-73.

31. Ali, M. Z., Suganthan, P. N., Reynolds, R. G., Al-Badarneh, A. F. (2016). Leveraged neighborhood restructuring in cultural algorithms for solving real-world numerical optimization problems. *IEEE Transactions on Evolutionary Computation, 20(2),* 218-231.

32. Ramasso, E., Saxena, A. (2014). Review and analysis of algorithmic approaches developed for prognostics on CMAPSS dataset. *Conference of the Prognostics & Health Management Society,* 612-622.

33. Yuan, Y., Xu, H., Wang, B., Zhang, B., Yao, X. (2016). Balancing convergence and diversity in decomposition-based many-objective optimizers. *IEEE Transactions on Evolutionary Computation, 20(2),* 180-198.

34. Hu, J., Tang, H., Tan, K. C., Li, H. (2016). How the brain formulates memory: a spatio-temporal model research frontier. *IEEE Computational Intelligence Magazine, 11(2),* 56-68.

35. Zhao, Z., Bin, L., Wang, X., Lu, W. (2017). Remaining useful life prediction of aircraft engine based on degradation pattern learning. *Reliability Engineering & System Safety, 164,* 74-83.

36. Brown, G., Wyatt, J., Harris, R., Yao, X. (2005). Diversity creation methods: a survey and categorisation. *Information Fusion, 6(1),* 5-20.

37. Saxena, A., Goebel, K., Simon, D., Eklund, N. (2008). Damage propagation modeling for aircraft engine run-to-failure simulation. *International Conference on Prognostics & Health Management,* 1-9.

38. Ramasso, E. (2014). Investigating computational geometry for failure prognostics in presence of imprecise health indicator: results and comparisons on C-MAPSS datasets. *Second European Conference of the PHM Society, 5,* 1-18.