Proceedings of the 2nd International Conference on Computing Innovation and Applied Physics
DOI: 10.54254/2753-8818/18/20230449

Research on combination of periodic inspection and data-driven maintenance system of turbofan engine

Runxing Wang

School of Energy and Power Engineering, East China University of Science and Technology, Shanghai, 200237, China

loganwilliamwrx@gmail.com

Abstract. In the field of aviation, turbofan engine is a crucial high-precision component of an aircraft which requires rigorous maintenance to ensure the running stability, safety, and efficiency. To reduce the high cost of general maintenance, the system of combination of periodic inspection and Data-driven maintenance is a meaningful research trend. In this pattern, the engine's condition study over sensor data serves as the foundation for the maintenance, to meet the safety requirements, improved sensor data processing and interpretation and precise prediction model plays important role. In this study, Long Short-Term Memory (LSTM) architecture was used to estimate RUL (remaining useful life) and support periodic inspection. The experiments were taken by using the dataset of FD001 and FD003 in *Turbofan engine degradation* dataset that contains operating and degradation information of several turbofan engines. After being analysed by different evaluation methods, the algorithm is capable to discover the features that hidden among sensor data, which indicates that the model has promising results for predictive maintenance. Therefore, the Data-driven maintenance structure that integrated with LSTM architecture is proposed as an application method. Furthermore, considering all aspects of the real application, management tools, Kubernetes and Kubeflow are introduced in this paper. Finally, regularly scheduled inspection is introduced, with the help of estimated RUL, inspection can be applied by parts rather than removing engine for overhaul and replacement of all parts including parts in good condition.

Keywords: periodic inspection, data-driven maintenance system, turbofan engine.

1. Introduction

With the development of technology, various of industrial facilities occurred and applied to different regions, as one of the greatest achievement in industrial, aero-engine represents the highest manufacturing level, the turbofan engine is a variation of basic aero-engine that is widely used in civil aviation, which represents a gas turbine that generates reaction thrust using the air expelled by the fan and the gas injected by the nozzle, its application determines its working condition is very tough, and is subject to various effects, such as changes in temperature and air pressure [1]. When failure occurs, the engine stops generating thrust as it used to be because of malfunction of certain parts. Research that covered 7571 airplanes that had registered in America since 1980 showed that mostly aircraft failure happened at the engine, considering the whole propulsion system, 57% of accidents were due to the system mechanical malfunction [1], so the engine maintenance is one of the dominant parts for the air safety to avoid such failure. However, the current maintenance pattern is less efficient and costs a lot, which costs 55% as much of the whole aircraft maintenance [2], therefore an advanced maintenance system and method is necessary to schedule the maintenance plan more efficiently at the premise of maintaining safety and reliability.

With the advent of era of big data and development of computer science, AI-based data-driven approach is becoming feasible to solve problems efficiently, different AI architectures has been proposed with different performance. Che et al. proposed an AI architecture combining multiple deep learning algorithms [3], predict the RUL by integrate Deep Belief Network and LSTM, the key concept is to use healthy indictors (HI) and different labels to build an architecture, the estimate of the RUL is obtained by setting a threshold. Hinchi et al. a deep RUL prediction based on Convolutional Neural Network and LSTM was proposed, which directly extracts the features of sensor data by convolutional layers and then predict the RUL by LSTM layers [4].

The new maintenance method given out in this paper is the combination of regularly scheduled inspection and Data-driven maintenance. Regularly scheduled inspection focus on the inspection of core components based on the data from sensor located on the certain components and pre-flight and post-flight normal inspection, refuelling, lubricating oil scrubbing and lubrication, etc. Data-driven maintenance focus on the RUL prediction based on LSTM architecture. During the turbofan engine performance degradation, there exist tight temporal correlations between sensor data [5], RUL prediction is to identify the implicate operating rule of the turbofan engine components and apply the rule to estimate the RUL based on the historical sensor data [6]. Furthermore, the deep learning architecture needs to be updated on account of different route and environmental condition, a private cluster of Kubernetes on local server is necessary to manage the whole process of the system, including the architecture development and update. In addition, a polit cueing system that sifts out abnormal data that possibly implies malfunction of the certain component of engine is integrated to avionics system to provide polit more reacting and thinking time in case of malfunction.

For the regularly scheduled inspection, the maintenance is based on the predicted RUL and worked time cycles of each part, so different parts have customized solution such that the cost of maintenance could reduce by cancelling unnecessary parts maintenance and replacement.

2. Data-driven maintenance based on LSTM architecture

2.1. Algorithms architecture

By combining multiple sequences and connecting before and after, RNN can make predictions based on current information and historical information. However, with the increasing complexity of neural networks, RNNs often suffer from information overload and over-optimization. As a variant of RNN algorithm, the LSTM is composed of cell, output gate, input gate, forget. The three gates control certain information enters and leaves the cell to store values throughout time. Compared to RNN network, the cell memory of an LSTM network can store new data as well as erase part of its previously recorded data. The vanishing gradient issue that can arise when training conventional RNNs is resolved by the forget gates in LSTM units.

As a result, the information extraction is more selective by using LSTM, thereby boosting the use of information and the precision of time series prediction [7], obtains shorter training time and little error. Figure 1 shows general LSTM network structure, x denotes input data, while o denotes output for each unit of time t in a network.

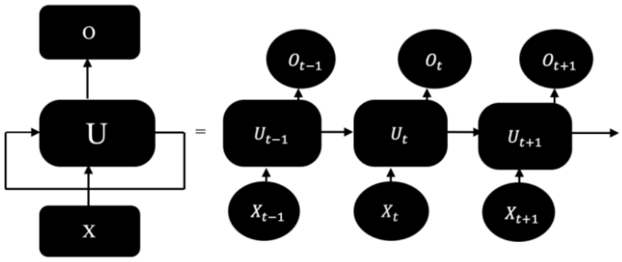

Figure 1. LSTM structure.

Figure 2 indicates the internal LSTM structure, the term c_t refers to the input from the preceding cell memory, h_t refers to the output that is sent to output layer and following hidden layer, respectively. Inputs for time t, sigmoid activation function adopts x_t and h_{t-1} [8], and the output determines the state of forget gate. To identify the memory line state, the identical combination of x_t and h_{t-1} is sent through the tanh function and sigmoid function. Cell state and hidden state and different gates provide abundant internal states. Suppose that engines that have N components are monitored by m sensors during operation, the sensor data that each engine has collected during its lifespan is displayed as a matrix: Xn=[x1,x2,···,xT], Xn ∈ Rm×T, where T stands for the time of failure, xt=[x_1^t, x_2^t,···, x_m^t] stands for the m-dimensional sensor measurements at time t, and xi=[x_i^1, x_i^2,···, x_i^T] refers to the time sequence of sensor data. During the trainning section, the LSTM model adopts the sensor measurement sequence Xn and learns changing trend of RUL. The algorithm takes in the data, then acquires predicted RUL_{est}^t as output at time t.

After data is supplied to the LSTM unit, it undergoes flowing steps:

Firstly, the forget gate f_t controls the long-term memories that should be forgotten.

$$f_t = \sigma(W_f h_{t-1} + U_f \mathbf{x}_t + b_f) \tag{1}$$

Secondly, the input gate discovers the information of the input data and decides which part of it should be stored into the memory cell.

$$g_t = \tanh(W_g h_{t-1} + U_g \mathbf{x}_t + b_g) \tag{2}$$

$$i_t = \sigma(W_i h_{t-1} + U_i \mathbf{x}_t + b_i) \tag{3}$$

After that, renew the information in the cell state.

$$c_t = c_{t-1} \otimes f_t + g_t \otimes i_t \tag{4}$$

Finally, according to the input and cell state, with function of output gate, hidden layer state is updated.

$$o_t = \sigma(W_o h_{t-1} + U_o \mathbf{x}_t + b_o) \tag{5}$$

$$h_t = o_t \otimes \tanh(c_t) \tag{6}$$

where W_f, W_g, W_i, W_o stands for the variable weight between the current and previous hidden layer; U_f, U_g, U_i, U_o stands for the weight between the current input and hidden layer; b_f, b_g, b_i, b_o stands for bias vector; *tanh* stands for activation function; σ stands for sigmoid function; ⊗ stands for element-wise multiplication.

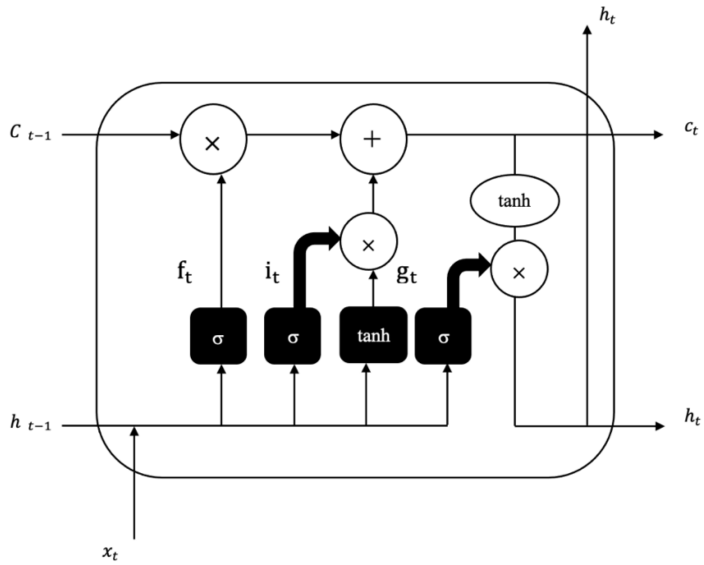

Figure 2. LSTM unit structure.

The purpose of the model is to calculate the parameters to obtain the purpose of objective function minimization so that prediction is accurate enough. The objective function is as follows:

$$\sum_{t} \| (RUL_{est}^{t} - RUL_{\mathrm{cal}}^{t}) \|^2 \tag{7}$$

where RUL_{est}^{t} stands for estimated RUL, and RUL_{cal}^{t} stands for acquired result between current time and failure time.

2.2. Turbofan engine degradation dataset

This paper makes use of Turbofan engine degradation dataset that has 4 subsets with different operation conditions and failure models. Each row records data in a certain period, with engine index as columns, current operational cycle number, setting of impact on engine performance that is significant, and values of sensors [9]. At the beginning of each time cycle, the engine is in good condition and encounters a failure at some time that is unknown. For the training set of each subset, the failure gradually appeared, and a portion of the data is delivered up to sometime before the failure. in each test set. Table 1 lists the composition of dataset.

Table 1. The composition of dataset [5].

Dataset	Operating conditions	Fault modes	Train size	Test size
FD001	1	1	100	100
FD002	6	1	260	259
FD003	1	2	100	100
FD004	6	2	248	249

2.3. Data preparation

When the engine is operating, data acquired from different parts of the engine and different sensors are unprocessed, it's necessary to normalize the invalid data respectively before applying to the model, since the value scale of various sensors may differ [10], for this purpose, normalization method of Min-Max is introduced to deal with the acquired data, normalized data is within the range of (0, 1).

Min-max normalization:

$$x_i' = \frac{x_i - min\, x_i}{\max x_i - min\, x_i} \tag{8}$$

where x_i' is the normalized sensor data.

2.4. Evaluation

The evaluation of the predictive architecture is vital to have feedback of the RUL estimation performance on the test data. There are multiple evaluation methods to apply to different applications, two of them are given in this paper [11].

2.4.1. Scoring function evaluation. Following shows the definition of the function:

$$s = \begin{cases} \sum_{\mathrm{i}=1}^{n} (\mathrm{e}^{\frac{\mathrm{h_i}}{10}} - 1), & h_i > 0 \\ \sum_{\mathrm{i}=1}^{n} (\mathrm{e}^{-\frac{\mathrm{h_i}}{13}} - 1), & h_i < 0 \end{cases} \tag{9}$$

where *n* is the number of samples, $h_i = RUL_{est,i} - RUL_i$, RUL_i is true RUL.

In this scoring function, different penalty is given when the model outputs different estimate results. If acquired RUL is less than the true RUL, since according to the prediction, the maintenance will be applied to the engine before the failure, penalty is lesser; conversely, it means that because of the bad estimation, maintenance would not be scheduled properly, the engine may encounter malfunction while operating, obviously, the penalty will be larger.

2.4.2. Root Mean Square Error (RMSE) evaluation. It is widely used of RMSE method as performance indicator for evaluating regression prediction models.

RMSE is sensitive to outliers because greater errors have an unreasonably large impact. In general, the smaller the value, the smaller the deviation between predicted values and true values, the better performance and higher accuracy the model. No matter the projected RUL is less or more than the actual RUL, RMSE assigns the model equal penalty weights.

Formally it is shown as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} h_i^2} \tag{10}$$

2.5. Experiment results and further optimization

2.5.1. Experiments results. The predicted results of turbofan engines in the test set FD001 and FD003 are shown in Figure 3 and Figure 4. The closer the vertical distance between two points, the more accurate the prediction results. In general, the performance of model prediction on the RUL value is good, of course for several individual engine there exists a large deviation between estimated RUL and the actual RUL. In order to acquire better architecture performance, the number of nodes, layers was changed in several experiments. Each experiment was repeated five times and results were averaged.

For the subset FD001, different experiments with the optimization of number of nodes and layers were recorded in Table 2. As an illustration, for the first experiment, (32,64) indicates that the architecture has two LSTM layers with 32 and 64 nodes respectively, (8,4) indicates that following the LSTM layers, there are two layers of Neural Network with 8 and 4 nodes respectively.

DOI: 10.54254/2753-8818/18/20230449

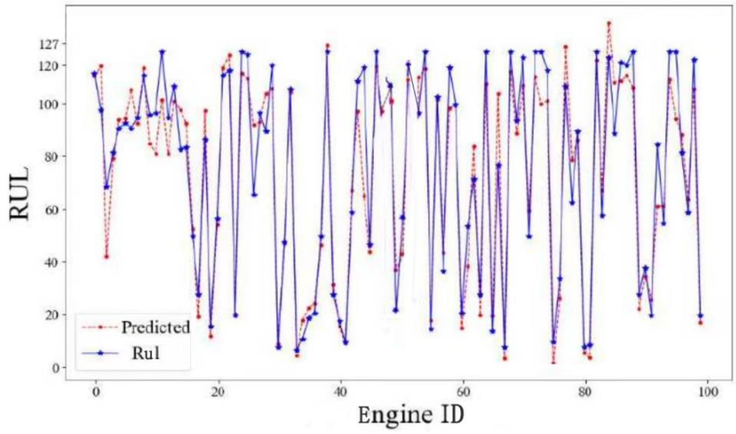

Figure 3. Predicted result of FD001 [4].

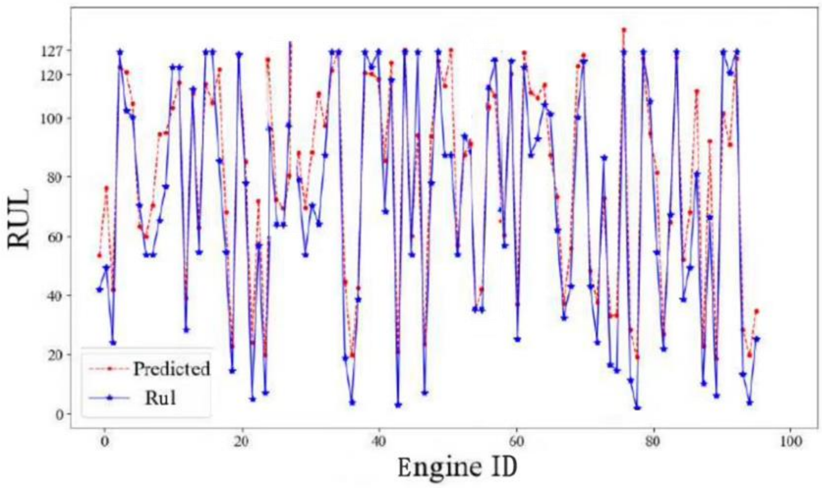

Figure 4. Predicted result of FD003 [4].

Table 2. The composition of dataset [5].

Experiments on subset FD003		
Evaluation **Network/** LSTM layer: Neural network layer:	**Rmse**	**Score**
(32,64) (16,18)	23.17	201.92
(64,128) (16,8)	26.69	231.98
(64,128) (8,8)	22.62	173.46
(32,64,32) (8,4)	25.22	209.43
(32,64) (16,8,4)	18.76	192.

Table 3. Results of experiments with various architecture [8].

Experiments on subset FD001		
Evaluation **Network/** LSTM layer: Neural network layer:	**Rmse**	**Score**
(32,64) (8,4)	15.17	158.39
(64,64) (8,8)	13.75	137.17
(64,128) (8,8)	13.62	169.21
(32,64,32) (8,4)	14.37	198.32
(32,64) (16,16)	14.15	183.82

According to the results, the model is capable to discover the hidden features of sensor data with enough efficiency under multiple operating and failure conditions.

The model has worse performance on subset FD003 since compared to subset FD001, the FD003 has more complex operating conditions and failure models, so the hidden features among them is difficult by contrast, in addition the data from sensors is easily be polluted by data noise, which means the dataset includes interfering data, i.e., random error or variance in the measured variable that disturb the model fit the data.

2.5.2. Optimization. Overfitting is one of common problems of deep learning architectures, which occurs when a model fits precisely almost perfectly and follows the trend of each data on the training set, but obviously the real situation varies, the same trend as the training set does not recur, some parts of the network architecture learned useless information that has no contribution to the overall trend of certain situation, which means the model performs almost perfect on the training data but bad on testing

DOI: 10.54254/2753-8818/18/20230449

data, the model will have a difficult time generalizing on a new unseen dataset. In order to avoid such problem, adding Dropout layer and Batch Normalization layer is very useful. In the experiments, different dropout rates were tried and 0.2 was kept as appropriate rate, therefore Dropout layer and Batch Normalization layer were added between two LSTM layers.

As time passes, the system health deteriorates linearly, in the aspect of processing target values assignment, because in practical, the components degrade negligibly at the beginning of operation, after a certain operating cycle that approaches end-of-life, the degradation increases, so a piece-wise liner RUL target function lifts the model performance according to the experiments. After a given level of usage, it starts linear decline and restricts the maximum RUL to a fixed value and helps the model fits data better. According to most of corresponding research, time cycle that within the range of [120, 130] is apropos to the real situation [12], so 127 time cycles is set as degradation limitation.

The loss of model training also affects the performance. The model loss is shown in the Figure 5. Obviously, the loss value of the model training after 40 iterations converges to around 0.3, so the maximum number of iterations for the model training is selected as 50.

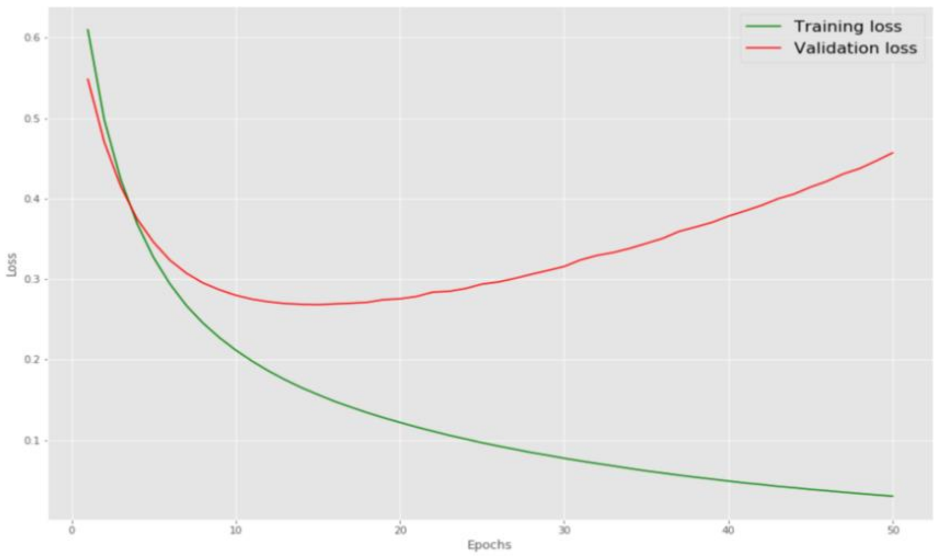

Figure 5. Training and validation loss of FD001.

2.6. Data-driven maintenance structure

The Data-driven maintenance structure is shown in Figure 6, the system has two components, one is RUL prediction that determines the maintenance decision, and the other is the Polit Cueing System.

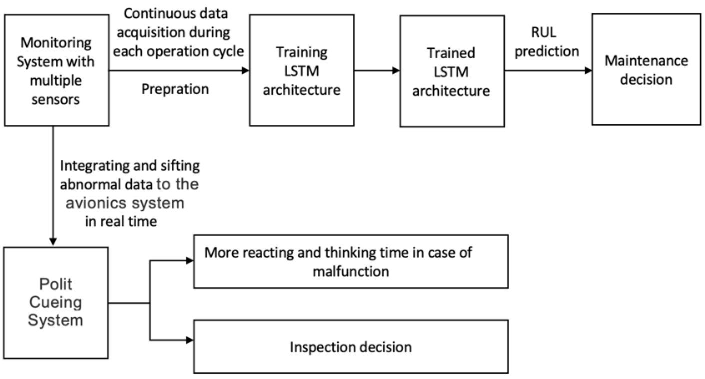

Figure 6. Data-driven maintenance structure.

For the first part, the system is developed for ground maintenance. For one operating cycle, firstly, there is a monitor system with multiple sensors to collect various engine operating performance parameters and build datasets that have been divided into training sets and test sets; then apply them to the LSTM network to train the model; finally, the engine maintenance is determined based on the predicted RUL. In practical application, it is often necessary to provide long-term prediction information to peruse a higher accuracy and make different maintenance decisions with the purpose of safe flight, so the prediction architecture needs to be updated duly after applying different cycles of data. In order to finalize the update of architecture efficiently, CI/CD technique helps a lot, which is indicated in section 2.6.2.

For the second part, abnormal data that possibly implies malfunction of the certain component of engine from sensor of monitoring system is sifted and integrated to avionics system in real time as Polit Cueing System such that pilots can get an overview of the engine overall operating status, which provides pilots more thinking time and enables them to come up with more reasonable solutions in case of malfunction happens during flight. Besides, the presence of a certain amount of abnormal data may represent the loss of certain component, which provides precise recommendation of later engine inspection and replacement of easy-to-spoil components.

2.6.1. Sensor selection. There are a large number of sensors are mounted on the different components of engine, the appropriate selection of sensor enables acquirement of valuable data of crucial components that determine the operating state. Some of representative correlation between sensor data and RUL is shown in Figure 7, obviously, the correlation of sensor 1 is a flat line that indicates the sensor data remains constant throughout time and hold no useful information; sensor 4 shows a rising trend and sensor 7 shows a declining trend; sensor 6 shows the data peaks downwards occasionally but with no strong relation to the RUL [13]. Based on the correlation graphs, sensors providing data that carrying no hidden features increase the amount of computation merely even influence the regression decision of the model rather than having contributions to elevating the performance, so certain sensors need to be added to the elimination list.

DOI: 10.54254/2753-8818/18/20230449

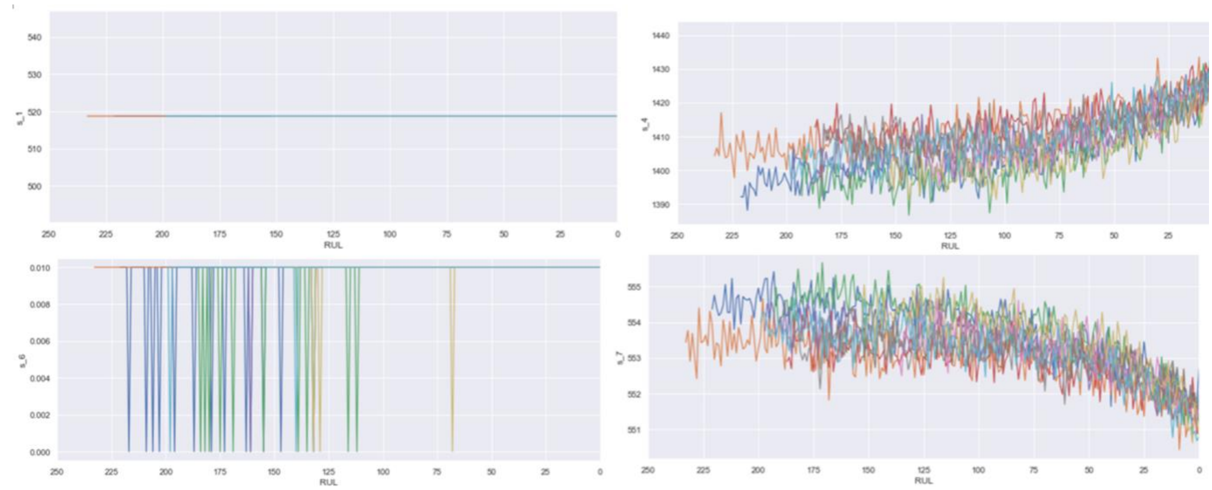

Figure 7. Representative correlation between sensor data and RUL [9].

For turbofan engine, the fan, compressor, turbine, combustion chamber are the most important parts, and their operating condition largely determines the engine's performance.

(1) Fan factor

As the efficiency of the fan decreases, the required work of the fan to maintain the speed and pressure ratio increases, and it is necessary to increase the power of low-pressure turbine to keep converted speed of engine fan from decreasing. To increase power of low-pressure turbine, gas flow and inlet temperature need to be increased.

(2) Compressor factor

When the compressor efficiency decreases, the required work of the compressor increases, the required power of the compressor is greater than the power generated by high-pressure turbine, the compressor speed decreases, the air flow at the compressor inlet decreases, the engine bypass ratio increases, the low-pressure turbine power decreases. In order to maintain the engine speed, the fuel flow is increased, resulting in an increase in the combustion chamber outlet temperature.

When fluctuation of sensor data occurs, it indicates that the operating conditions of important components have changed, which possibly implies the occurrence of malfunction. Therefore, sensors from these components acquire valuable data that has strong correlation with RUL, which increases the performance of prediction model.

2.6.2. CI/CD technique. Continuous Integration (CI) and Continuous Delivery (CD) are acronyms for these processes. In the simplest words possible, CI is a contemporary software development technique where code changes are regularly made, code updates are integrated into the private repository by CI. As part of the CD process, the code is swiftly and easily supplied. The term "CI/CD pipeline" refers to the automation that enables developers to deploy incremental code changes rapidly and reliably to production. More specifically, the update process of the system introduced in this paper belongs to MLOps which is a derivative of CI/CD that aims to deploy and maintain machine learning models in production reliably and efficiently. The tools that will be introduced in this section is the pipeline tool Kubeflow and the management tool Kubernetes.

In real application, the system takes in engine data of every operating cycle and computational tasks are finalized on whether Cloud Service or local server (Higher data security), in aviation local server is a better choice to meet various requirements of security, following steps help to accelerate the development and update:

(1). Deploy private Kubernetes cluster to manage the maintenance system on local server.

(2). Develop update of the model on Kubeflow and push to the private repository with specific tag.

DOI: 10.54254/2753-8818/18/20230449

Kubeflow provides various components to meet demands of different development stage. For example, in training phase, pipeline tool creates a workflow that can record the training results and parameters, which provides clear overview of network performance. Kubeflow provides environment information such that integration of different developers' work is more compatible.

(3). Deploy the image of the updated model from repository to the Kubernetes cluster.

After the above process, the relationship between system users and developers is established, users can manage and update the maintenance system on powerful tool Kubernetes.

3. Regularly scheduled inspection

By analysing the sensor data, the distribution of the engine operating cycles is shown in Figure 8, according to the graph, most engines encounter failure around 200 operating cycles, right skewed distribution shows few engines remain good condition more than300 cycles.

With the help of predictive system, by analysing the operating data and predicted RUL, the distribution of the number of cycles to failure can be obtained. In addition, the components manufacturers will have the failure testing to the parts before delivery. By having the overall consideration of distribution and the failure tasting of the engine parts, making regularly scheduled inspection and maintenance decision based on parts.

Regularly scheduled inspection focus on the inspection of core parts including Core, Fan Blades, Combustor, Compressor, Bleed Air Ducting, Turbine Cases, etc. and pre-flight and post-flight normal inspection, refuelling, lubricating oil scrubbing and lubrication, etc.

In this new engine maintenance method, first set the appropriate threshold value of RUL and the predicted RUL as the first factor in deciding whether to apply maintenance to the engine, then compare the time to failure of different parts and the RUL respectively to determine if the maintenance of parts is required.

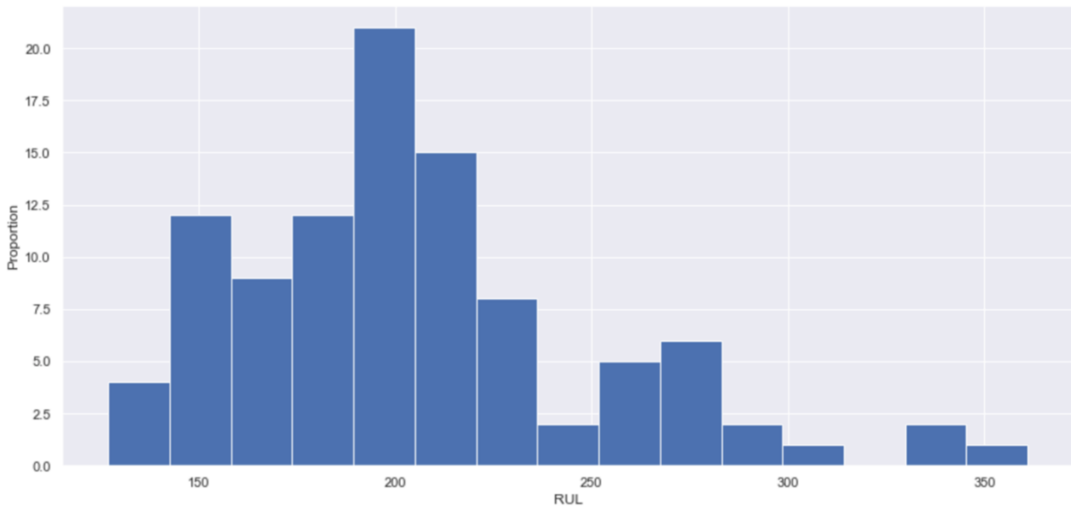

Figure 8. Distribution of engine operating cycles.

4. Conclusion

This paper gives out a new turbofan engine maintenance method of periodic inspection and Data-driven maintenance that tries to apply appropriate solution to reduce the cost of traditional maintenance method under the premise of ensuring reliability and safety.

For this purpose,

(1) A deep artificial neural network structure, LSTM was proposed to learn engine degradation trend to estimate the engine RUL to schedule maintenance. In addition to improving the fitting precision, the model reduces the model overfitting and overestimation of the model prediction, which can reduce the

accident rate caused by engine damage in practical applications to a certain extent. Relatively, there is improvement for the model, it can be combined with other convolutional neural networks to strengthen the learning of temporal features, and some genetic optimization algorithms can be introduced. By analysing the results, it has been discovered that predicted RUL is close to the true situation, so it proved that the application of this predictive system is feasible, which provides the foundation for other maintenance process in the entire system. In addition, the solution of implementing the system was proposed, which provides powerful tools to manage the whole life cycle of AI architecture, accelerates both development and update process.

(2) A periodic inspection based on estimated RUL and fatigue limit of parts was proposed to apply customized maintenance to the engine. Compared to general maintenance methods, the new method reduced costs by providing precise parts repair and replacement recommendations rather than removing engine for overhaul and replacement of all parts including parts in good condition during each maintenance.

With expansion of aviation and resource shortage, this kind of low-cost maintenance method will be a very useful solution for aircraft health management.

References

[1] Mercer D L., Simon, G W. 2020 Fundamental Technology Development for Gas-Turbine Engine Health Management. Energies, 34: 307-31.
[2] Price J, Jimenez O, Parthasarathy V. 2000 Ceramic Stationary Gas Turbine Development Program -Seventh Annual Summary. American Society of Mechanical Engineers, 20: 101-108.
[3] Che C., Wang H., Fu Q., Ni X.. 2020 Combining multiple deep learning algorithms for prognostic and health management of aircraft. Aerospace Science and Technology, 94: 107-109.
[4] Ahmed Z. 2020 Rolling element bearing remaining useful life estimation based on a convolutional long-short-term memory network. Procedia Computer Science, 127: 123-132.
[5] Yu W., Chris M. 2021 An improved similarity-based prognostic algorithm for RUL estimation using an RNN autoencoder scheme. Reliability Engineering & System Safety, 199: 208-210.
[6] Xia J., Feng Y., Cheng L. 2018 LSTM-based multi-layer self-attention method for remaining useful life estimation of mechanical systems. Engineering Failure Analysis, 125: 56-67.
[7] Zhang Y., Xiong R., He H. 2021 Long Short-Term Memory Recurrent Neural Network for Remaining Useful Life Prediction of Lithium-Ion Batteries. In IEEE Transactions on Vehicular Technology, 67(7): 5695-5705.
[8] Düdükçü H M., Taşkıran M. 2020 LSTM and WaveNet Implementation for Predictive Maintenance of Turbofan Engines. 2020 IEEE 20th International Symposium on Computational Intelligence and Informatics (CINTI), 20: 000151-000156.
[9] Emmanuel A. 2014 Performance Benchmarking and Analysis of Prognostic Methods for CMAPSS Datasets. International Journal of Prognostics and Health Management, 5(2): 1-15.
[10] Zheng S., Ristovski K. 2017 Long Short-Term Memory Network for Remaining Useful Life estimation. 2017 IEEE International Conference on Prognostics and Health Management (ICPHM), 20: 88-95.
[11] Zhang J, Lang X, Yang J, Wang W, Wang D. 2020 Comparative Analysis of Domestic and Foreign Gas Turbine Emission Standards. Environmental Protection and Circular Economy, 40(9): 71-74.
[12] Liu Z, Shu X, Yang A, Li Y. 2020 Selection of air pollutant emission standard limits for stationary gas turbines. China Power, 53(8): 117-124.
[13] Saxena A., Goebel K. 2008 Damage propagation modeling for aircraft engine run-to-failure simulation. 2008 International Conference on Prognostics and Health Management, 20: 1-9.