Long short-term memory neural network with scoring loss function for aero-engine remaining useful life estimation

Article *in* Proceedings of the Institution of Mechanical Engineers Part G Journal of Aerospace Engineering · May 2022

DOI: 10.1177/09544100221103731

CITATIONS
6

READS
77

3 authors, including:

Li-Hua Ren
Nanjing University of Aeronautics & Astronautics
3 PUBLICATIONS **53** CITATIONS
SEE PROFILE

Yong-Ping Zhao
Nanjing University of Aeronautics & Astronautics
57 PUBLICATIONS **826** CITATIONS
SEE PROFILE

Original Article

Institution of MECHANICAL ENGINEERS
Journal of AEROSPACE ENGINEERING

Proc IMechE Part G:
J Aerospace Engineering
2023, Vol. 237(3) 547–560

DOI: 10.1177/09544100221103731
journals.sagepub.com/home/pig
SAGE

Long short-term memory neural network with scoring loss function for aero-engine remaining useful life estimation

Li-Hua Ren, Zhi-Feng Ye and Yong-Ping Zhao

Abstract

Estimation of the aero-engine remaining useful life (RUL) is a significant part of prognostics and health management (PHM) and the basis of condition-based maintenance (CBM) which can improve the reliability and economy. Multiple operating conditions, nonlinear degradation, and early prediction are significant and distinctive issues compared with other prognostics problems. While these issues do not get enough attention and researches in aero-engine RUL estimation. In view of these points, three specific data preparation approaches and a novel loss function are introduced. The data preparation approaches can extract high-quality data for the long short-term memory (LSTM) neural network according to the characteristic of aero-engine degradation data. Among these approaches, operating condition normalization is an effective method to handle the multiple operating conditions problems, and RUL limitation identification is a novel method to identify the turning point of the nonlinear degradation process. The scoring function is an innovative loss function used to replace the mean square error (MSE) loss function which has a preference for early prediction. The comparisons with the original LSTM and some other approaches indicate that the combination of the data preparations and the scoring loss function is an effective solution for the above issues, and can achieve the best performance among the approaches.

Keywords

Aero-engine, remaining useful life, long short-term memory, scoring loss function, early prediction

Date received: 25 August 2021; accepted: 9 May 2022

Introduction

The reliability and economy of aero-engine will descend with aging. To handle this problem caused by degradation, prognostics and health management (PHM) is applied to the aero-engine field. PHM is an emerging engineering technology focusing on failure mechanisms and life cycle management.[1] It is the upgrade of condition-based maintenance (CBM) and aims at extending machinery service life while reducing maintenance costs. Prognostics is the process of monitoring the health of a product and predicting its remaining useful life (RUL) by assessing the extent of deviation or degradation from its expected state of health in its expected usage conditions.

Researches regarding RUL has sprung up in recent years. According to mainstream researches, prognostics approaches can be classified into three types: experience-based, physics model-based, and data-driven.[2] The simplest method of measuring RUL is to record engine running time and estimate the RUL according to the life usage provided by the manufacturer. The second one is based on the aero-engine physics model which combines mechanical dynamics, aerodynamics, engine structure, material characteristics, etc.[3] The approach is efficacious for estimating the RUL of some components while the complexity and difficulty limit the effectiveness when extended to the whole engine. Data-driven approaches use the airborne data acquisition systems, which provided complete usage records for a fleet of aircraft, and calculate the RUL by creating a machine learning model. The chosen degree of sophistication depends on the required life measurement accuracy and this, in turn, will typically be determined by perceived cost savings and safety requirements. To gain a worthwhile benefit, the life prediction and life measurement systems adopted need to have a better balance on accuracy and sophistication. The data-driven approaches have better accuracy than experience-based ones and lower sophistication than

College of Energy and Power Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, China

Corresponding author:
Yong-Ping Zhao, College of Energy and Power Engineering, Nanjing University of Aeronautics and Astronautics, No.29 Yudao Street, Nanjing 210016, China.
Email: y.p.zhao@163.com

physics model-based ones, which makes it a good alternative for RUL estimation at this stage.

There are plenty of research branches and methods on RUL estimation emerging over time. The degradation process of aero-engine is a typical time series problem, therefore, the algorithms being capable of handling time series problems are wildly studied and show better performance.[4] Badu et al.[5] made the first attempt to adopt convolution neural network (CNN) for RUL estimation and compared CNN with other traditional algorithms such as multi-layer perceptron (MLP), support vector regression (SVR), and relevance vector regression (RVR), CNN achieves the best performance. Although CNN is an alternative for time series problems, the lack of sequential order pattern leaves a gap in the accuracy between recurrent neural network (RNN) algorithms.[6] Long short-term memory (LSTM) neural network is the most wildly used RNN algorithm, the introduction of "gate" giving LSTM the ability to memorize or forget the information of a long time series.[7] Researches on RUL estimation with LSTM springs nowadays. Zheng et al.[8] made the first attempt to adopt LSTM for aero-engine RUL estimation, multiple layers of LSTM cells in combination with standard feed-forward layers can discover the hidden patterns from the sensor and the operational data with multiple operating conditions. Although operating conditions are considered as input variables, there still are accuracy gaps between the datasets with a single operating condition and the datasets with multiple operating conditions. Setting the operating conditions as the inputs of the LSTM model directly is not an effective way to handle the problem caused by multiple operating conditions. A dynamic difference technology was proposed by Wu et al.[9] to extract new features from original datasets. K-means clustering and "one-hot" encoding were introduced to distinguish different operating conditions while the early prediction was neglected in the RUL prediction. Wang et al.[10] applied bidirectional LSTM to RUL estimation, the hidden patterns with sensor data of multiple operating conditions were exposed by analyzing the hidden layers visually. The introduction of bidirectional LSTM can extract more hidden patterns while the accuracies of the datasets with multiple operating conditions were still not satisfactory.

Several novel feature extracting and ensemble frameworks sprung up in the past several years. Shi et al.[11] proposed a new Dual-LSTM framework that connects the change point detection and RUL prediction with a newly proposed HI construction function. The change point detection can filter out the unnecessary sensor signals irrelevant to the degradation and improve the accuracy. The autoregressive integrated moving average (ARIMA) model and the long short-term memory model are integrated by Fan et al.[12] The ARIMA model is a good tool for stable decline rate curves which can filter linear trends in the production time series data and improve the accuracy of the LSTM model. A similarity-based curve matching method combining RNN encoding for RUL estimation was proposed by Yu et al.[13] The trained RNN autoencoder serves as a feature extractor for the health index (HI) curves which represent various degradation patterns of the training instances and will be used to RUL estimation via curve similarity analysis. Although these frameworks can obtain more hidden patterns between sensors and RUL, the multiple operating conditions problem and early prediction have not been paid enough attention.

Most of these researches focus on the frameworks or the combinations of LSTM, and neglect the influence of data preparation and early prediction. Elsheikh et al.[14] paid attention to the early prediction demand and proposed the approach that uses asymmetric loss function for early prediction. However, they could not handle the problems caused by the exponential term in the scoring function. Thus, they proposed the asymmetric squared error (ASE) and the asymmetric absolute error (AAE) as the approximation of the scoring function. The most important contribution of this paper is using the scoring function as the loss function of LSTM for early prediction successfully. Meanwhile, few papers proposed improved methods for data preparation. In fact, the data preparations have a great influence on accuracy. Especially, the aero-engine data have sophisticated characteristics: strong nonlinearity, multi-conditions, and multi-sensors, etc. According to the characteristics of aero-engine RUL data, three special data preparation approaches with some improvements are introduced. The feature selection method with a linear degradation model (LDM) and the method to identify the RUL limitation in the target function are also the main contributions of this paper.

This paper is structured as follows. In the Data overview section, the data sets generated from commercial modular aero-propulsion system simulation (C-MAPSS) for aero-engine RUL estimation are introduced in detail and their characteristics are analyzed. In the Data preparation section, three special data preparation approaches with improvements are introduced according to the data characteristics. In the Long short-term memory with scoring section, the mathematical model of LSTM is dwelled on, and the LSTM is improved by replacing the loss function with a scoring function. In Long short-term memory with scoring for RUL estimation section, the proposed Scoring LSTM with the improved data preparation approaches are tested with C-MAPSS and PHM08 data sets, then the performance is compared with the original LSTM and other reported approaches. Finally, conclusions follow.

Data overview

The C-MAPSS dataset provided by the researchers of the National Aeronautics and Space Administration (NASA) has been employed to evaluate the performance of the proposed method.[15] They also launched a challenge in which they provided some new C-MAPSS data as the "PHM08 challenge data" for competitors to evaluate their results.[16]

Table 1. Data set details.

	C-MAPSS				PHM08
Data set	FD001	FD002	FD003	FD004	
Training trajectories	100	260	100	249	218
Test trajectories	100	259	100	248	218
Operating conditions	1	6	1	6	6
Fault modes	1	1	2	2	2

Table 2. Content of the matrix.

Engine number	Time/Cycle	Operating condition 1	Operating condition 2	Operating condition 3	Sensor 1	Sensor 2	…	Sensor 21
1	1	-0.0007	-0.0004	100	518.67	641.82	…	23.4190
1	2	0.0019	-0.0003	100	518.67	642.15	…	23.4236
…	…	…	…	…	…	…	…	…

Data sets: The C-MAPSS data set is further divided into four sub-datasets (FD001 to FD004), and each sub-dataset is further divided into training and testing subsets as shown in Table 1. Each trajectory represents a complete run-to-failure data of an engine in the training set, and part of the run-to-failure data is provided in the test set for RUL estimation. The operating conditions and fault modes are various for different sub-datasets, the sets with multiple operating conditions and fault modes make it harder to predict.

The data sets are arranged in an n-by-26 matrix where n corresponds to the number of data points. Each row is a snapshot of data taken during a single time cycle with 26 columns, where the 1st column represents the engine number, the 2nd column represents the time, 3–5 columns represent the operating conditions, and 6–26 columns represent the 21 sensor values. The content of the matrix is shown in Table 2, and more information about the data sets can be found in paper.[15]

Data preparation

The data has some sophisticated characteristics: strong nonlinearity, multi-conditions, and multi-sensors, etc. Considering these characteristics, some specific data preparations are taken to improve the performance of the training algorithm.

Operating condition normalization

Operating condition normalization (OCN) is a recognized and effective approach to improve the accuracy of the multi-condition data. The approach was first proposed in paper,[17] and is integrated into the Matlab toolbox nowadays. It is the most critical and essential step in this data preparation and the following shows the main ideas. The amplitude of sensor data varies when distributed in different operating conditions. If there is no operating condition segmentation, the data will not show a visible degradation trend. Figure 1(a) shows the selected sensor data of FD004 which has six operating conditions appears ruleless when normalized with the Z-Score method directly. While the data shows an obvious degradation trend when normalized with condition segmentation as shown in Figure 1(b). Clustering is a frequently used

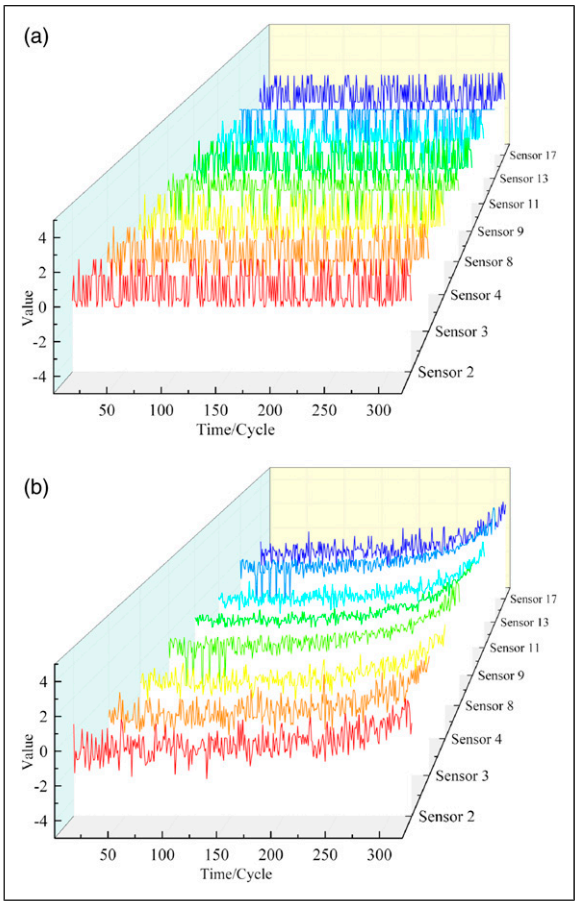

Figure 1. Normalized values with different approach. (a) Z-Score normalize directly and (b) Normalize with operating condition segmentation.

condition segmentation approach. The quantity of conditions is known in this problem, hence K-means clustering will be a good alternative. For the training set, the data need to be clustered with K-means according to the corresponding operating conditions and conserve the centers and labels, then normalize the data with equation (1). For the testing set, the cluster of the data needs to be found according to Euclidian distance from the centers obtained from the training set, and then normalize the data with equation (1).

$$\mathbf{X}_{norm}^{(c,s)} = \frac{\mathbf{X}^{(c,s)} - \mu^{(c,s)}}{\delta^{(c,s)}}, \forall c,s \tag{1}$$

where X denotes all the data of sensors, c denotes the cluster index, and s denotes sensor index, μ and δ are mean value and standard deviation, respectively. The data normalized with OCN can be also considered as one operating condition and the operating conditions are no longer the input of the network.

Feature selection with linear degradation model

There are many sensor signals in the data and most of them are irrelevant to the RUL, so feature selection is significant. However, few papers provide a systematic approach for feature selection and most of them rely on experiences and trials. A feature selection approach based on LDM is proposed. Some researchers have proved that mechanical degradation usually follows LDM or exponential degradation model (EDM).[18–21] LDM describes the degradation behavior as a linear stochastic process with an offset term and EDM describes the degradation behavior as an exponential stochastic process with an offset term. EDM is useful when the test component experiences cumulative degradation.

Aero-engine experiences cumulative degradation generally and the EDM is more suitable for estimating the RUL. While the adoption of the degradation model is not for the RUL estimation, but the feature selection in this paper. The function of the degradation model is to reveal the degradation trend. Intuitively, the slope amplitude of the signal which can be obtained directly via the parameter $\theta(t)$ in LDM is a reasonable criterion for evaluating the trend. In addition to this, the slope of the LDM is a time-varying variable which makes the model fit nonlinear characteristics. Thus, LDM will be a better choice for feature selection.

The LDM model is described as follows

$$S(t) = \phi + \theta(t)t + \varepsilon(t) \tag{2}$$

where ϕ is the model intercept which is a constant. $\theta(t)$ is the model time-varying slope and is modeled as a random variable with a normal distribution. $\varepsilon(t)$ is the model additive noise and is modeled as a normal distribution with zero mean. More details of solving equation (2) can be found in paper.[20] The absolute mean slope of the LDM is defined as equation (3).

$$|\mu_\theta| = \sum_{i=1}^{T} |\theta(t)| \tag{3}$$

where T is the length of the time series. It is also logical that the larger descending or rising slope of the sensor means the more obvious trend of the degradation in the same magnitude. Thus, the sensor with a larger $|\mu_\theta|$ is deemed to have a more evident degradation trend. Thus, the approach computes the LDM of each sensor and selects the sensors with larger $|\mu_\theta|$ as features. For each sub-dataset, eight sensors with the largest slopes are selected, and their number and slopes are listed in Table 3, in which sensor 11 (Ps_{30})and sensor 4 (T_{50}) are the most obvious in all the sub-datasets which corresponds to the aero-engine mechanism. In addition, the selections are closed to other papers. Figure 2 gives the trend of sensor 11 (selected sensor in FD001) and sensor 14 (unselected sensor in FD001), sensor 11 indeed presents a more apparent trend. The figure indicates that the LDM can select the sensors with more apparent trend as the input of the network. Meanwhile, the figure also indicates that there is no degeneration trend for the sensors at the initial stage of the aero-engine (the area "Before degradation"), and the method to identify the threshold for degradation will be discussed in the next section.

Remaining useful life target function

The provided target RULs decrease with time in the C-MAPSS dataset. When the engines are completely

Table 3. Sensor selected and their absolute mean slopes.

FD001		FD002		FD003		FD004	
Sensor	$\|\mu_\theta\|$	Sensor	$\|\mu_\theta\|$	Sensor	$\|\mu_\theta\|$	Sensor	$\|\mu_\theta\|$
11	0.0125	11	0.0126	11	0.0105	11	0.0097
4	0.0123	4	0.0121	4	0.0102	4	0.0092
12	0.0121	15	0.0119	17	0.0094	17	0.0085
7	0.0119	17	0.0106	2	0.0090	3	0.0082
15	0.0116	2	0.0104	3	0.0090	9	0.0077
21	0.0115	3	0.0100	9	0.0077	2	0.0075
20	0.0114	12	0.0088	13	0.0074	13	0.0069
17	0.0109	7	0.0083	8	0.0074	8	0.0069

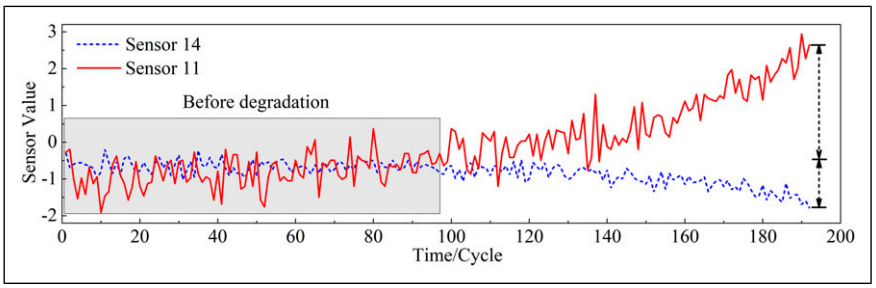

Figure 2. Degradation trend of selected and unselected sensors.

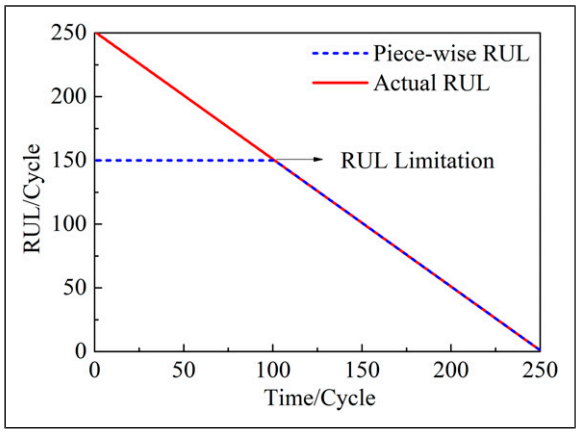

Figure 3. Piece-wise linear RUL target function.

healthy (Figure 2 "Before degradation"), the sensor data will not present a degradation trend at the beginning which will cause wrong mappings (same sensor data maps different RULs) and influence the prediction accuracy. Consequently, a piece-wise linear RUL target function was proposed in paper,[22] which limits the maximum value of the RUL function in training sets as illustrated in Figure 3. The linear RUL function follows the definition of RUL in the strictest sense which is defined as the time to failure. While the piece-wise RUL is also a more logical model as the degradation of the system typically only starts after a certain degree of usage. The RUL target function can prevent the algorithm from overestimating the RUL and achieve higher accuracy. Now, piece-wise RUL target function is a recognized approach because it can remarkably improve the accuracy. However, the RUL limitation is usually chosen based on trials in most papers,[5,8,10,23] and few papers give a method to identify the limitation.

Remaining useful life limitation identification with Euclid distance

In this paper, a valid and simple method for the RUL limitation identification is proposed. The main idea is to find a piece-wise linear function (normalized RUL) to regress the normalized sensor value and ensure the minimum error, and the error is evaluated with Euclid distance between the curves (Figure 4(a)). The process of identifying the RUL limitation can also be considered as

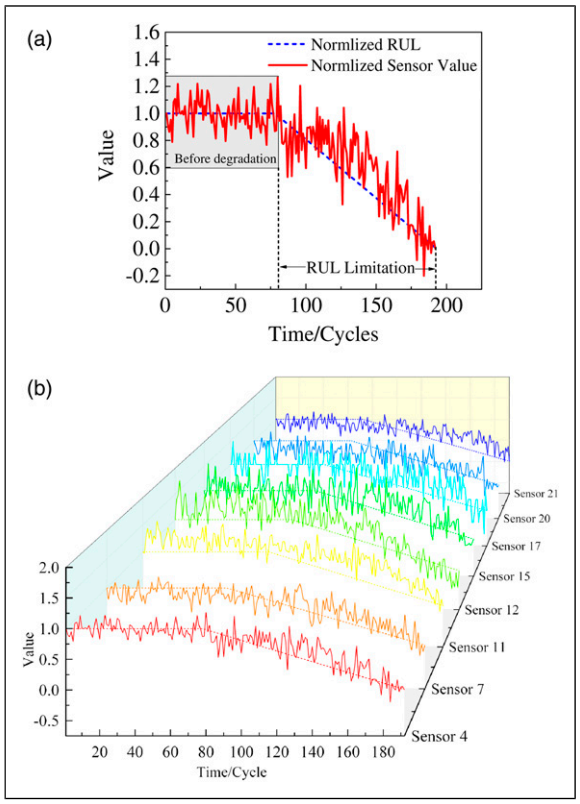

Figure 4. Normalized sensor value and piece-wise RUL. (a) Normalized value of the first select sensor and (b) Normalized value of all the select sensors.

identifying the length of the "Before degradation" area. The normalized RUL can be obtained from the piece-wise RUL (blue line in Figure 3) with max-min normalization. The normalized sensor value is fixed and the piece-wise RUL can be uniquely identified by the RUL limitation, thus the optimization problem is to find the RUL limitation to minimize the distance between the curves in Figure 4. For the reason that the RUL limitation is an integer between 0 and 400, enumeration will be a faster choice than optimizing algorithms. The following are the steps of the method.

1. Normalize the sensor data with the first and last value of each degradation trajectory. Considering the strong noise and the macroscopic monotonicity, the data is normalized with the first and last value instead of the

maximum and minimum value in this algorithm. #equation (4)
2. Obtain the piece-wise RUL functions of each trajectory with the enumerated RUL limitations, and then normalize these functions. #equation (5)
3. Calculate the Euclid distance between the normalized sensor data and normalized piece-wise RUL target function for each enumerated RUL limitation. The limitation with the minimum distance is the most suitable limitation for the trajectory. #equations (6) and (7)
4. Calculate the mean value of limitations obtained in step 3 as the RUL limitation of the data set. #equation (8)

$$\mathbf{X}_{norm\_rul}^{(s,t)} = \frac{\mathbf{X}^{(s,t)} - x_{T_{s,t}}^{(s,t)}}{x_{T_{s,t}}^{(s,t)} - x_1^{(s,t)}}, \forall s,t \tag{4}$$

$$\mathbf{Y}_{norm\_rul}^{(s,t),l} = \frac{\mathbf{Y}^{(s,t)}}{l^{(s,t)}}, \forall s,t,l \in [0, T_{s,t}] \tag{5}$$

$$D^{l^{(s,t)},l} = f(l^{(s,t)}) = \left\| \mathbf{X}_{norm\_rul}^{(s,t)} - \mathbf{Y}_{norm\_rul}^{(s,t),l} \right\| \tag{6}$$

$$p^{(s,t)} = \text{argmin}(f(l^{(s,t)})) \tag{7}$$

$$P = \frac{\sum_{s=1}^{S} \sum_{t=1}^{T} L^{(s,t)}}{S \times T} \tag{8}$$

where,

1. X_{norm_rul} and Y_{norm_rul} denote the normalized sensor data and piece-wise target RUL, respectively, s and t are the sensor index and trajectory index, respectively, S is the total number of sensors, and T represents the length of the trajectory.
2. D represents the Euclid distance and l represents the variable of the enumerated RUL limitation; $f(\cdot)$ represents calculation function of the Euclid distance.
3. p is the RUL limitation of each trajectory and P is the mean value of the limitations.

Table 4 lists the RUL limitations of different data sets. The data sets with multi-condition (FD002, FD004, and PHM08) have higher limitations than the data sets with a single condition. Figure 4(a) and (b) visualize the normalized sensor data and piece-wise RUL target function with the given RUL limitation of the first trajectory in FD001. The figure indicates that the RUL target function with the given RUL limitation has the same degradation trend and turning-point with the sensors.

Long short-term memory with scoring

Long short-term memory

Long short-term memory is a type of RNN network for time series and sequence problems and has obtained many achievements in speech recognition and machine translation. LSTM is explicitly designed to avoid long-term dependency by introducing input gate, forget gate, cell candidate, and output gate which can control information

Table 4. Identified RUL limitations on C-MAPSS and PHM08 data set.

	C-MAPSS				PHM08
Data set	FD001	FD002	FD003	FD004	
RUL limitation	117	127	121	126	130

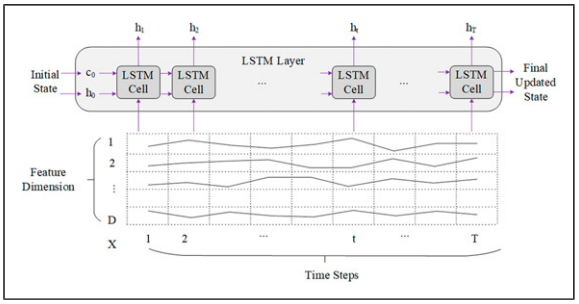

Figure 5. LSTM layer architecture.

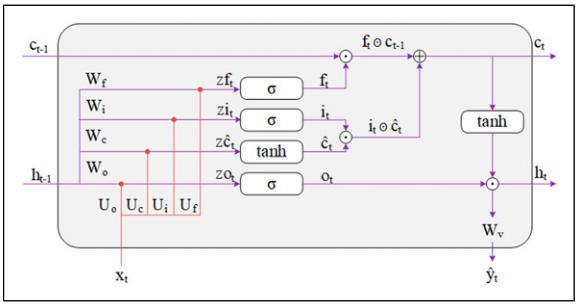

Figure 6. The forward propagation of LSTM cell.

flow. The layer structure of LSTM is in the form of a chain of repeating module called LSTM cell. Figure 5 illustrates the flow of a time series $X \in \mathbb{R}^{D \times T}$ with D features and T time steps through an LSTM layer. In this diagram, h denotes the hidden state and c denotes the cell state. Like other feedback neural networks, LSTM has three main training process: forward propagation, backpropagation, and updating gradients. The difference is that LSTM has time continuity, the inputs are a series of continuous points instead of snapshots and the propagation mechanisms are based on time.

Long short-term memory cells have a complicated structure instead of a single tanh activator in RNN. There are four components interacting in a special way which are forget gate $\boldsymbol{f}$, input gate $\boldsymbol{i}$, cell candidate $\widehat{\boldsymbol{c}}$, and output gate $\boldsymbol{o}$. Figure 6 illustrates the forward propagation details of the cell at time t. The cell receives h_{t-1}, c_{t-1} and the input $\boldsymbol{x}_{t-1}$ obtained at time $t-1$, then h_t and c_t will be calculated for the next time step and output $\widehat{\boldsymbol{y}}_t$ through forward propagation. The transition equations of the propagation are shown as equation (9) to equation (12).

$$\begin{aligned} \mathbf{f}_t &= \sigma(\boldsymbol{W}_f \boldsymbol{h}_{t-1} + \boldsymbol{U}_f \boldsymbol{x}_t + \boldsymbol{b}_f) \\ \boldsymbol{i}_t &= \sigma(\boldsymbol{W}_i \boldsymbol{h}_{t-1} + \boldsymbol{U}_i \boldsymbol{x}_t + \boldsymbol{b}_i) \\ \widehat{\boldsymbol{c}}_t &= tanh(\boldsymbol{W}_c \boldsymbol{h}_{t-1} + \boldsymbol{U}_c \boldsymbol{x}_t + \boldsymbol{b}_c) \\ \mathbf{o}_t &= \sigma(\boldsymbol{W}_o \boldsymbol{h}_{t-1} + \boldsymbol{U}_o \boldsymbol{x}_t + \boldsymbol{b}_o) \end{aligned} \tag{9}$$

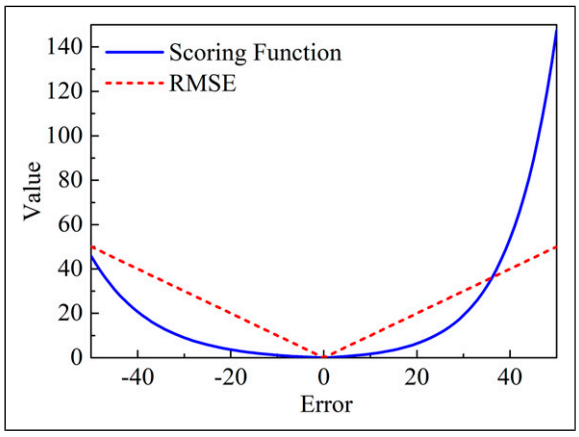

Figure 7. Comparison of Scoring function and RMSE.

$$\mathrm{c}_t = \boldsymbol{i}_t \mathrm{e} \widehat{\mathrm{c}}_t + \boldsymbol{f}_t \mathrm{e} \mathrm{c}_{t-1} \tag{10}$$

$$\boldsymbol{h}_t = \mathrm{o}_t e tanh(\mathrm{c}_t) \tag{11}$$

$$\widehat{\mathrm{y}}_t = \boldsymbol{W}_v \cdot \boldsymbol{h}_t + \boldsymbol{b}_v \tag{12}$$

where $\boldsymbol{W}$ and $\boldsymbol{U}$ denote the corresponding recurrent and forward weights; b denotes the corresponding biases; $\delta(\cdot)$ and $tanh(\cdot)$ denote sigmoid and tanh activation function, respectively.

Scoring loss function

Definition of scoring loss function. To evaluate the models' performance on the data, a scoring function is given by paper[15] and equation (13) shows the definition.

$$S = \begin{cases} \sum_{i=1}^{n} (e^{-\frac{d_i}{13}} - 1), & \text{for } d_i < 0 \\ \sum_{i=1}^{n} (e^{\frac{d_i}{10}} - 1), & \text{for } d_i \geq 0 \end{cases} \tag{13}$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} d_i^2} \tag{14}$$

where $d = \widehat{t}_{RUL} - t_{RUL}$ (estimated RUL – actual RUL) and n is the number of aero-engines.

This scoring function penalizes late predictions (too late to perform maintenance) more than early predictions (no big harms although it could waste maintenance resources) as Figure 7 shows. This is in line with the risk-averse attitude in aerospace industries. Compared with the frequently used evaluation function, viz. root mean square error (RMSE) shown as equation (14), the scoring function has better evaluating capability for aero-engine RUL estimation. Almost all the papers on aero-engine RUL estimation adopted the scoring function to evaluate their algorithms' performance, while none utilized the function to improve their algorithms.

Mean square error is generally used as the loss function to train the LSTM regression neural network, and it gives equal weights to negative errors and positive errors. The loss function of the LSTM is replaced with the scoring function for the sake of early prediction (negative errors). In order to distinguish from the original and some other improved LSTMs, the approach is named LSTM with scoring. In the LSTM with scoring, the loss function is defined as equation (15) which needs to be minimized in the backpropagation process.

$$J_t = \begin{cases} e^{-\frac{\widehat{y}_t - y_t}{13}} - 1, & \text{for } \widehat{y}_t < y_t \\ e^{\frac{\widehat{y}_t - y_t}{10}} - 1, & \text{for } \widehat{y}_t \geq y_t \end{cases} \tag{15}$$

Backpropagation through time of the scoring loss function. Backpropagation through time (BPTT) is the training algorithm used to update weights in recurrent neural networks. Like the forward propagation in LSTM, the calculation also flows through time while in the opposite direction. The calculation starts from time T and backward to calculate the learnable weights respected to the loss function. Figure 8 illustrates the BPTT details of the cell at time t. The cell receives the gradients obtained at time $t+1$ and other values obtained in the forward propagation process and then calculates the gradients at time t for the next step backpropagation.

The gradients of the output weights $\delta \mathrm{W}_v$ and biases $\delta \mathrm{b}_v$ which do not participate in the cell recurrence can be obtained directly like general backpropagation (BP) neural network, the calculation formulas are given as equation (16) and (17).

$$\delta \mathrm{W}_v^t = \frac{\partial J_t}{\partial \widehat{\mathrm{y}}_t} \cdot \mathrm{h}_t^\top \tag{16}$$

$$\delta \mathrm{b}_v^t = \frac{\partial J_t}{\partial \widehat{\mathrm{y}}_t} \tag{17}$$

where, $\frac{\partial J_t}{\partial \widehat{y}_t}$ is calculated as equation (18)

$$\frac{\partial J_t}{\partial \widehat{y}_t} = \begin{cases} -\frac{1}{13} e^{-\frac{\widehat{y}_t - y_t}{13}} \\ \frac{1}{10} e^{\frac{\widehat{y}_t - y_t}{10}} \end{cases} = \begin{cases} -\frac{1}{13}(J_t + 1), & \text{for } \widehat{y}_t < y_t \\ \frac{1}{10}(J_t + 1), & \text{for } \widehat{y}_t \geq y_t \end{cases} \tag{18}$$

The gradients of hidden state $\delta \mathrm{h}_t$ and cell state $\delta \mathrm{c}_t$ play an important role in the backpropagation and will be calculated firstly. Define the gradients of the gates and the gradients of the cell candidate (before activation) as equation (19)

$$\delta \mathrm{z}_{t+1} = \frac{\partial \sum_{\tau=t+1}^{T} J_\tau}{\partial \mathrm{z}_{t+1}} = \begin{bmatrix} \delta \mathrm{zf}_{t+1} & \delta \mathrm{zi}_{t+1} & \delta \mathrm{z}\widehat{\mathrm{c}}_{t+1} & \delta \mathrm{zo}_{t+1} \end{bmatrix}^\top \tag{19}$$

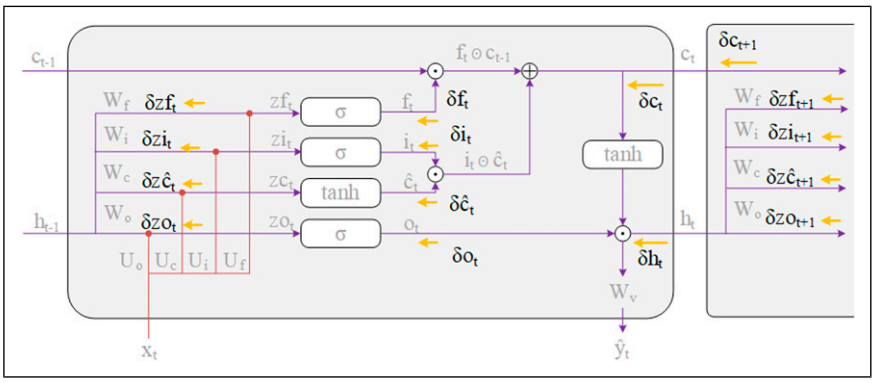

Figure 8. BPTT of LSTM cell.

Then δh_t and δc_t can be calculated as equation (20) and (21) with chain derivative rules

$$\delta h_t = W_v^{\top} \cdot \frac{\partial J_t}{\partial \hat{y}_t} + W^{\top} \cdot \delta z_{t+1} \tag{20}$$

$$\delta c_t = \delta h_t e o_t e \left(1 - tanh^2(c_t)\right) + \delta c_{t+1} e \delta f_{t+1} \tag{21}$$

where $W = [W_f \quad W_i \quad W_c \quad W_o]$. The gradients of the gates and cell candidate at time t are given as equation (22) and equation (23) according to the chain derivation of equation (10) and (11)

$$\begin{aligned} \delta f_t &= \delta c_t e c_{t-1} \\ \delta i_t &= \delta c_t e \hat{c}_t \\ \delta \hat{c}_t &= \delta c_t e i_t \\ \delta o_t &= \delta h_t e tanh(c_t) \end{aligned} \tag{22}$$

$$\begin{aligned} \delta z f_t &= \delta f_t e f_t e (1 - i_t) \\ \delta z i_t &= \delta i_t e i_t e (1 - i_t) \\ \delta z \hat{c}_t &= \delta \hat{c}_t e \left(1 - \tanh^2\left(\hat{c}_t\right)\right) \\ \delta z o_t &= \delta o_t e o_t e (1 - o_t) \end{aligned} \tag{23}$$

Then the gradients of input weights δU_t, recurrent weights δW_t and biases b_t can be calculated according to the derivation of equation (9) as equation (24) shown. Sum over time, then the weights and biases for updating can be obtained by equation (25)

$$\begin{aligned} [\delta U_t \quad \delta W_t] &= \delta z_t \cdot [x_t \quad h_{t-1}] \\ \delta b_t &= \delta z_t \end{aligned} \tag{24}$$

$$\begin{aligned} [\delta U \quad \delta W] &= \left[\sum_{t=1}^{T} \delta U_t \quad \sum_{t=1}^{T} \delta W_t\right] \\ \delta b &= \sum_{t=1}^{T} \delta z_t \end{aligned} \tag{25}$$

The adaptive moment estimation (Adam) optimization algorithm is adopted to update the weights and biases after getting the gradients. Adam is an extension to stochastic gradient descent that has recently been adopted and well works across a wide range of deep learning architectures.[24] The algorithm is a first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments. The advantages of this algorithm are high computational efficiency and little memory requirements. The algorithm is not insensitive to the learning rate which is set as 0.001 in this paper.

For comprehensively evaluation, the loss function of LSTM is also replaced with mean absolute error (MAE) and normalized root mean square error (NRMSE), the definations of MAE and NRMSE are shown as equation (26) and equation (27) separately. The BPTT processes are similar to the scoring loss function

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |d_i| \tag{26}$$

$$NRMSE = \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n} d_i^2}}{\frac{1}{n}\sum_{i=1}^{n} \hat{t}_{RUL}} \tag{27}$$

Long short-term memory with scoring for RUL estimation

Architecture

Given the excellent results achieved in paper,[8] the same Deep LSTM architecture is adopted, and the scoring loss function is set as loss function instead of MSE. The proposed architecture is shown in Figure 9. The network contains an input layer, two LSTM layers, and three fully connected layers. The input layer receives the prepared data in the form of 3-dimensional matrices. The input matrix $\boldsymbol{X} = [x_1 \quad x_2 \quad \cdots \quad x_T] \in \mathbb{R}^{10\times T\times 8}$, the 1st dimension is the batch size which is 10 for mini-batch learning. The mini-batch gradient descent can accelerate the training process compared with stochastic gradient descent. However, the lengths of trajectories are inconsistent, so the shorter trajectories need to be padded zeros according to the length of the longest trajectories in the mini-batch. The 2nd dimension size is the length of the longest trajectories and the third one is eight which is the number of selected features. What follows two LSTM layers and three fully connected layers. Their nodes are 32, 64, 8, 8, 1, respectively.

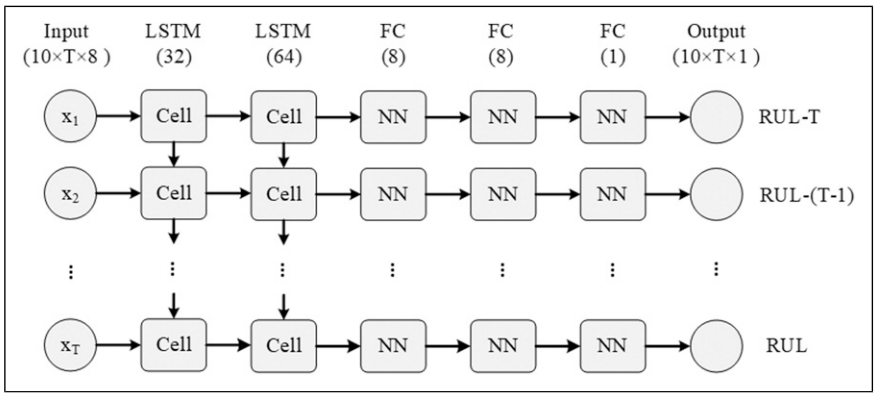

Figure 9. The architecture of LSTM with scoring for RUL estimation.

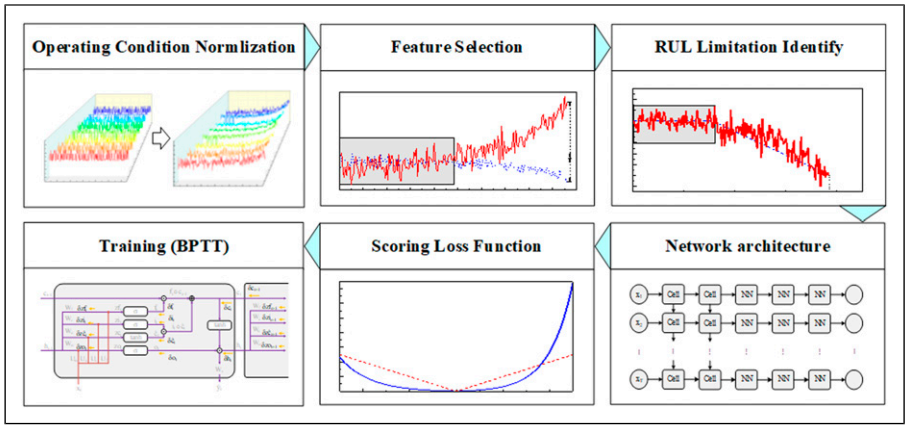

Figure 10. The training process of the proposed approach.

Training process

The flow chart of the whole training process is shown in Figure 10, and the training steps are described as follows.

1. Normalize the training sets of FD001 and FD003 with the Z-Score method directly, and apply operating condition normalization to FD002 and FD004 which have multiple conditions.
2. Calculated the mean slopes of the normalized features with LDM, and select the first eight features with the larger mean slope as the input features.
3. Construct the RUL target functions as the output of the LSTM, and identify the RUL limitations with Euclid distance.
4. Construct the architecture of the LSTM network, take the normalized features as input and the RUL target functions as output.
5. Set the scoring function as the loss function.
6. Iterate the BPTT process hundreds of times to minimize the scoring loss function, and get well-trained models for the RUL estimation.

The predicting process is the step 1 to step 4 above and replace the training sets with testing sets. The predicting results and analysis are arranged in the next section.

Table 5. Score comparison on the improvement of the OCN.

Data set approaches	FD001	FD002	FD003	FD004
Without OCN	451	4652	807	5764
With OCN	**438**	**3345**	**766**	**3258**

Table 6. Score comparison on the improvement of the RUL limitation identify.

Data set approaches	FD001	FD002	FD003	FD004
Limitation 150	438	3345	766	3258
Identified limitations	**304**	**3170**	**609**	**2945**

Results and analysis

Comparison with the original long short-term memory

There are four improvements compared with the original LSTM approaches: feature selection with LDM, OCN, RUL limitation identify with Euclid distance, and the scoring loss function. The first one gives a systematic solution for feature selection. The last three ones can improve the prediction performance and lead the preference to early prediction. To verify these three improvements separately, the improvements are introduced step by

step. First, OCN is introduced to the original LSTM in which the RUL limitations are 150 and the loss functions are MSE. Table 5 lists the scores on four C-MAPSS sub-datasets with and without OCN separately (the scores of the improved approaches are in bold in the following tables). The operating condition of FD001 and FD003 are single while the operating condition of FD002 and FD004 is 6, thus the OCN works only on FD002 and FD004. The scores decreased by 28% and 43% on FD002 and FD004 with the operating condition normalization. The results indicate that OCN can significantly improve the performance on the sub-datasets with multiple operating conditions.

Table 7. Score comparison on the improvement of the scoring loss function.

Data set approaches	FD001	FD002	FD003	FD004
MSE	304	3170	609	2945
MAE	292	3036	906	2886
NRMSE	308	3124	324	3078
Scoring	**228**	**2460**	**253**	**2080**

Second, the RUL limitation identification with Euclid distance is introduced based on the models with OCN. The effect is verified by using the RUL limitations in Table 4 for each sub-dataset rather than using a constant value.

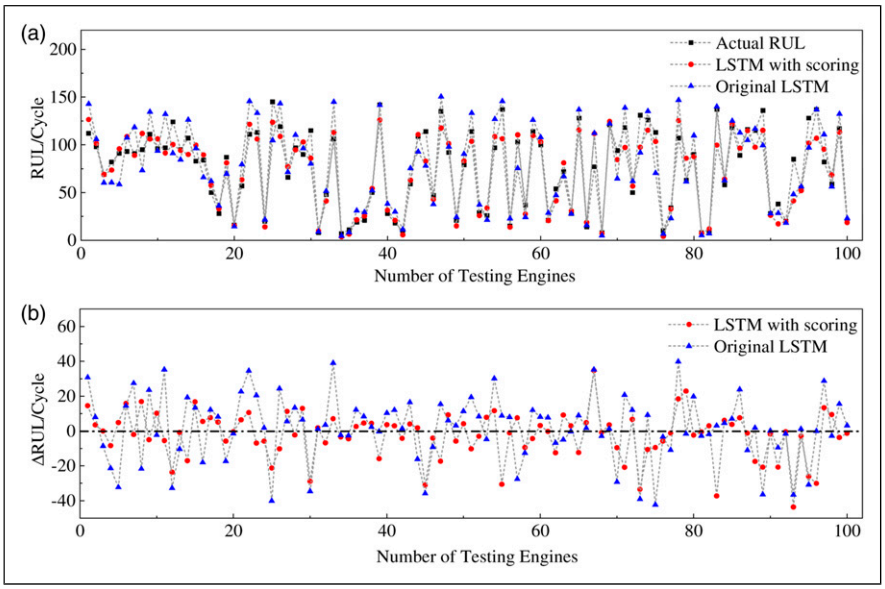

Figure 11. Comparison of the LSTM with scoring with the original LSTM on FD001. (a) Predicted and actual RULs and (b) Predicted residuals.

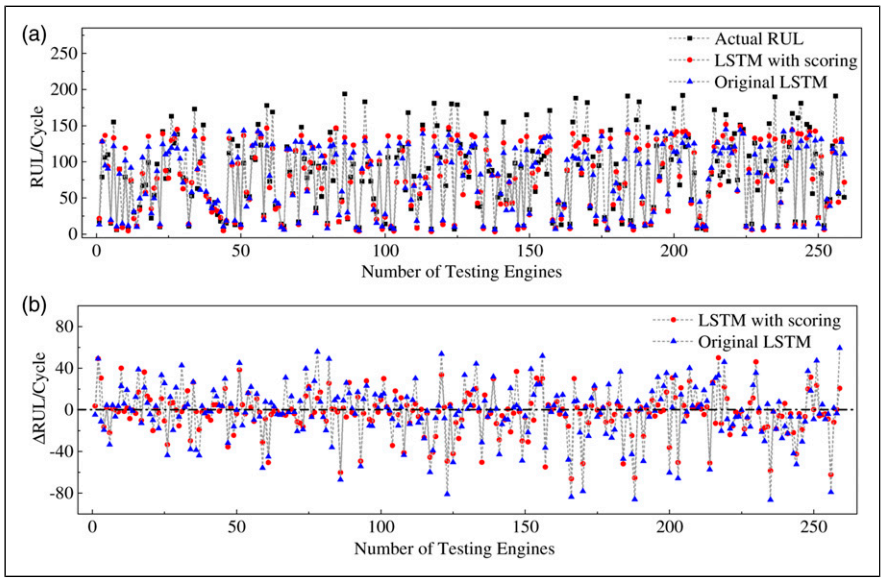

Figure 12. Comparison of the LSTM with scoring with the original LSTM on FD002. (a) Predicted and actual RULs and (b) Predicted residuals.

The scores in Table 6 shows that the performance of all the sub-datasets can be further improved based on the models with OCN.

Finally, the scoring loss function is introduced based on the models with OCN and RUL limitation identification, and the scores of different loss functions are listed in Table 7 for comparative analysis. The scores of MSE, MAE, and NRMSE are close to each other while the scores of scoring loss function are superior to others. The results indicate that OCN, the identified RUL limitations, and the scoring loss function can improve the performance separately and the combination of the improvements will get the best score.

The detailed comparison of the original LSTM and the LSTM with scoring is shown in Figures 11, 12, 13, and 14. In the figures, (a) compares the estimated RULs of the two approaches with the actual RULs and (b) compares the errors (ΔRULs, estimated RULs—actual RULs) on the four sub-datasets. The figures of ΔRULs indicate that the errors of LSTM with scoring are closer to zero which means that the accuracy of the proposed approach is higher.

The introduction of the scoring function is mainly for early prediction. The ΔRUL range of the two approaches are compared with boxplots in Figure 15. The boxplot indicates that most of the ΔRULs obtained by the LSTM with scoring are lower than zero which means the approach prefers early prediction, while the ΔRULs of the original LSTM distribute symmetrically.

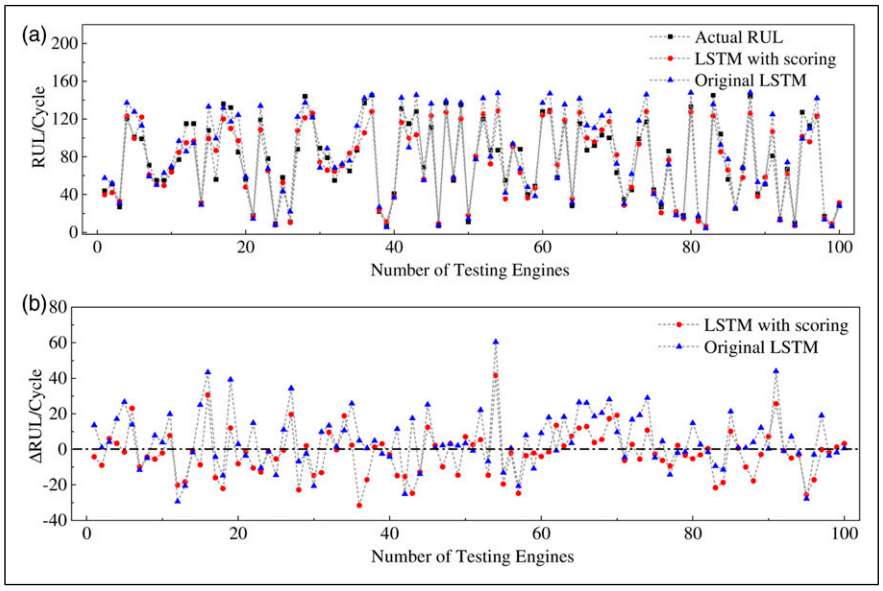

Figure 13. Comparison of the LSTM with scoring with the original LSTM on FD003. (a) Predicted and actual RULs and (b) Predicted residuals.

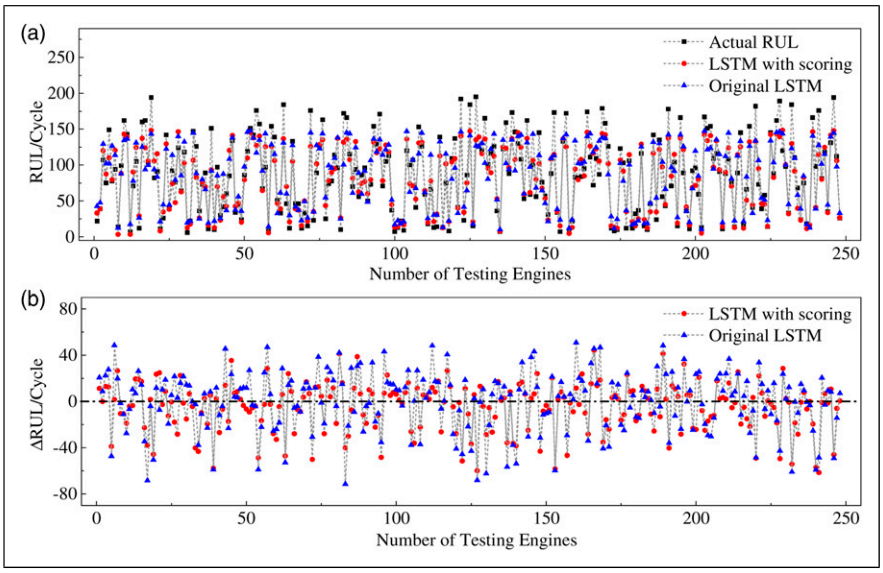

Figure 14. Comparison of the LSTM with scoring with the original LSTM on FD004. (a) Predicted and actual RULs and (b) Predicted residuals.

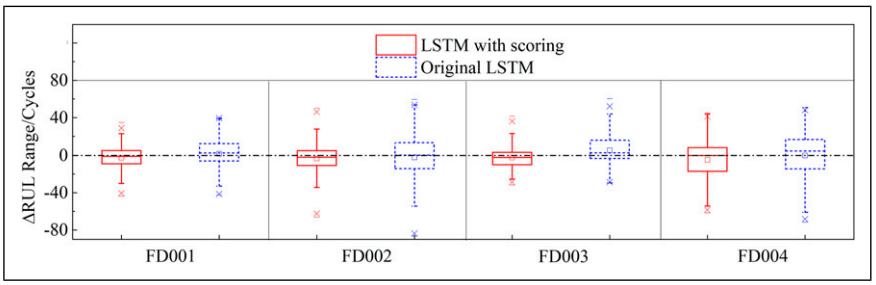

Figure 15. Comparison of the Δ RUL range on four sub-datasets.

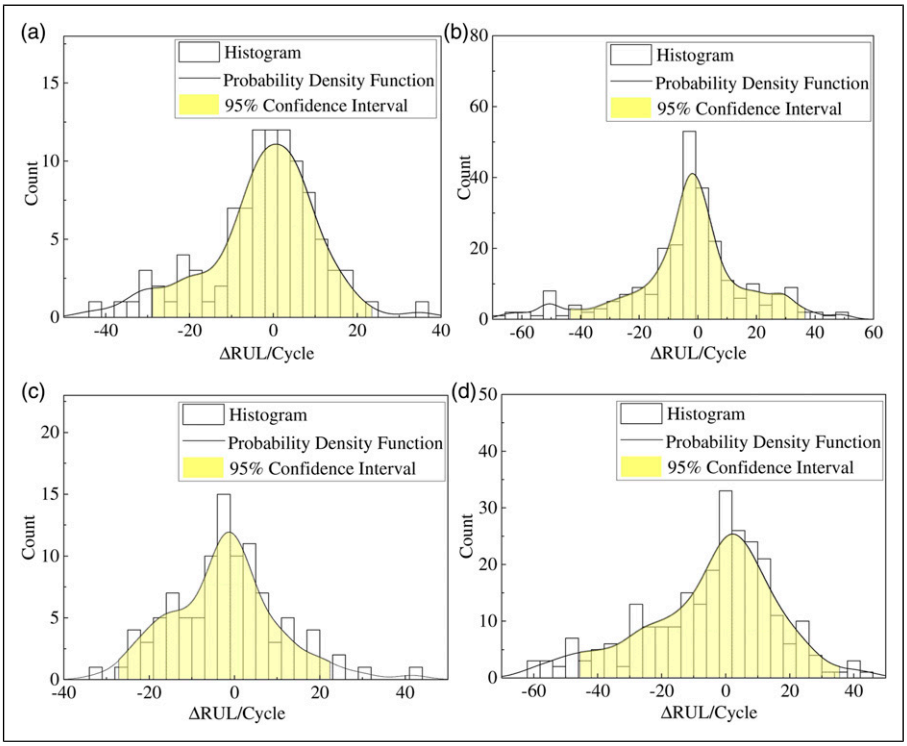

Figure 16. Statistical analysis of the errors. (a) FD001, (b) FD002, (c) FD003, and (d) FD004.

The histogram, probability density function, and confidence interval of the errors are shown in Figure 16 for comprehensive analysis. Where the Y-coordinate is the sample number of the histogram; the probability density function is the kernel density estimation of the histogram; the upper limit and lower limit of the 95% confidence intervals are listed in Table 8. The distributions of the errors also tend to be less than 0 which can be seen from the probability density functions and the confidence intervals.

Based on the above analysis, the LSTM with scoring outperforms the original LSTM and has an obvious preference for early prediction which can satisfy the demand of the aero-engine RUL estimation.

Table 8. Confidence Intervals on C-MAPSS data set.

Data set	FD001	FD002	FD003	FD003
Confidence intervals	[-29, 24]	[-42, 37]	[-27, 23]	[-46, 36]

Comparison with other reported results

The proposed approach is also compared with some other reported ones including MLP, SVR, RVR, and some state-of-the-art methods. Table 9 lists the scores where N/A represents the information unavailable. The results indicate that the proposed approach outperforms most of the other algorithms as well as Deep LSTM.[8] The architecture of the proposed LSTM is the same as the Deep LSTM while achieving better performance. This demonstrates that the data preparation improvements and the scoring loss function did improve the LSTM's performance on aero-engine RUL estimation.

If there are no special approaches or improvements added in MLP, SVR, and RVR,[5] the results are not satisfactory because these algorithms are weak in time series problems. CNN[5] shows better performance than

Table 9. Score comparison with other reported results.

	C-MAPSS				PHM08
Data set	FD001	FD002	FD003	FD004	
MLP[5]	1.80×10^4	7.80×10^6	1.74×10^4	5.62×10^6	3212
SVR[5]	1.38×10^3	5.90×10^5	1.60×10^3	3.71×10^5	15,886
RVR[5]	1.50×10^3	1.74×10^4	1.43×10^3	2.65×10^4	8242
CNN[5]	1.29×10^3	1.36×10^4	1.60×10^3	7.89×10^3	2056
Deep LSTM[8]	3.38×10^2	4.45×10^3	8.52×10^2	5.55×10^3	1862
BiLSTM[10]	2.95×10^2	4.13×10^3	3.17×10^2	5.43×10^3	N/A
Ensemble SBI[13]	2.28×10^2	2.65×10^3	1.73×10^3	2.90×10^3	N/A
BHLSTM[14]	3.77×10^2	N/A	1.42×10^3	N/A	N/A
DLSTM[23]	6.55×10^2	N/A	8.28×10^2	N/A	N/A
LSTM fusion[25]	2.55×10^2	1.40×10^3	2.11×10^2	7.73×10^3	N/A
LSTM with scoring	2.28×10^2	2.46×10^3	2.53×10^2	2.08×10^3	**738**

traditional algorithms while could not reach the accuracy level of LSTM. The performance of Deep LSTM[8] and BiLSTM[10] on FD001 and FD003 are not bad, while the performance on FD002 and FD004 are not satisfactory when compared with LSTM Fusion[25] and the proposed approach. Because there are six operating conditions in FD002 and FD004, the latter two approaches normalized the data according to different operating conditions and the first two didn't. There is also no operating conditions normalization in DLSTM[23] and BHLSTM[14] and the scores are not given. When the proposed approach is compared with LSTM fusion, the scores on FD001 and FD003 are close, while LSTM fusion had much better performance on FD003 than any other algorithm and quietly poor performance on FD004. And the LSTM fusion algorithm used two 6-layer LSTM layers with 100 or 50 nodes of each layer which makes the structure is much more complicated than the proposed approach.

The results of PHM08 are evaluated by uploading to NASA's prognostic data repository (https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/) and the score is 738 which is the same as the third-place score in the competition.[16] The actual RULs of PHM08 test set are not revealed and the scores are calculated by NASA's website. Hence, the results of PHM08 are more convincing.

Prospects

Although the scoring function proposed by NASA is a fairly well and universal evaluation criterion for aero-engine RUL estimation, the authors believe that the scoring function can be further improved. The scores shall be a little lower when the actual RULs are longer. There are two reasons:

1. When the actual RULs are long that means the aero-engines have an adequate safety margin, so the accuracy is not as important as the accuracy at shorter actual RULs.
2. In the beginning, engines are completely healthy and the sensor data will not present degradation trends, while the provided actual RULs still decrease with time. Consequently, the accuracy will be low and that is also the reason for introducing the Piece-wise RUL function.

Given this, the authors believe that making a differential treatment with the errors at different actual RUL periods will give a better evaluation criterion for aero-engine RUL estimation. An improved scoring function defined as equation (28) would make a difference

$$S(d_i,t_{RUL}) = \begin{cases} \sum_{i=1}^{n} \frac{(e^{-\frac{d_i}{13}} - 1)}{t_{RUL} + C}, & \textbf{for } d_i < 0 \\ \sum_{i=1}^{n} \frac{(e^{\frac{d_i}{10}} - 1)}{t_{RUL} + C}, & \textbf{for } d_i \geq 0 \end{cases} \tag{28}$$

where $C > 0$ is constant.

The thought of scoring loss function is a specific method for calculating gradients for the sake of early prediction. The thought can be applied to any other loss-based algorithm. In the future work, the scoring loss function will be applied to CNN-LSTM to validate its effectiveness.

Conclusions

Three special data preparation approaches with improvements and a novel Scoring LSTM for the aero-engine RUL estimation are proposed. The proposed approaches realize their function well and achieve excellent results compared with the original LSTM and other reported approaches. Some conclusions are followed by:

1. The OCN approach solved the problem that there is no clear trend showing the degradation process at multiple conditions. The OCN can improve the performance by 28% and 43% on FD002 and FD004 datasets which have six conditions.

2. A systematic and valid feature selection approach based on LDM is provided for the degradation data, and eight of the features with obvious degradation trends are selected with the LDM.
3. The introduction of the piece-wise linear RUL target function can ease the problem that the actual RULs mismatch the sensor data before the aero-engine degradation. A novel method to identify the RUL limitation of the target function is proposed. The identified RUL limitations can further improve the performance of the models with OCN when compared with constant RUL limitations.
4. The error analysis indicates that the scoring loss function makes the model have an obvious preference for early prediction when compared with the original LSTM. The results on CMAPSS and PHM08 data sets indicate that the proposed Scoring LSTM with the improved data preparations has better performance on RUL estimation than some other approaches.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This research was supported by the Fundamental Research Funds for the Central Universities under grant no. NS2022027.

ORCID iD

Yong-Ping Zhao https://orcid.org/0000-0003-3310-1329

References

1. Volponi AJ. Gas turbine engine health management: past, present, and future trends. *J Eng Gas Turbines Power* 2014; 136: 051201.
2. Schwabacher MA. *A Survey of Data-Driven Prognostics*. Isabela, PR: Infotech@Aerospace, 2005, p. 26–29.
3. An D, Choi J-H and Kim NH. Prognostics 101: a tutorial for particle filter-based prognostics algorithm using Matlab. *Reliability Eng Syst Saf* 2013; 115: 161–169.
4. Fink O, Wang Q, Svensén M, et al. Potential, challenges and future directions for deep learning in prognostics and health management applications. *Engineering Applications of Artificial Intelligence* 2020; 92: 103678.
5. Babu GS, Zhao P and Li X-L. *Deep Convolutional Neural Network Based Regression Approach for Estimation of Remaining Useful Life, International Conference on Database Systems for Advanced Applications*. Berlin: Springer, 2016, pp. 214–228.
6. Lecun Y and Bengio Y. *Convolutional Networks for Images, Speech, and Time-Series, the Handbook of Brain Theory and Neural Networks*. Cambridge, MA: MIT Press, 1995.
7. Hochreiter S and Schmidhuber J. Long short-term memory. *Neural Computation* 1997; 9: 1735–1780.
8. Zheng S, Ristovski K, Farahat A, et al. Long short-term memory network for remaining useful life estimation. In: 2017 IEEE International Conference on Prognostics and Health Management (ICPHM), 19–21 June 2017, 2017, pp. 88–95.
9. Wu Y, Yuan M, Dong S, et al. Remaining useful life estimation of engineered systems using vanilla LSTM neural networks. *Neurocomputing* 2018; 275: 167–179.
10. Wang J, Wen G, Yang S, et al. *Remaining Useful Life Estimation in Prognostics Using Deep Bidirectional LSTM Neural Network, 2018 Prognostics and System Health Management Conference (PHM-Chongqing)*. Piscataway: IEEE, 2018, pp. 1037–1042.
11. Shi Z and Chehade A. A dual-LSTM framework combining change point detection and remaining useful life prediction. *Reliability Engineering & System Safety* 2021; 205: 107257.
12. Fan D, Sun H, Yao J, et al. Well production forecasting based on ARIMA-LSTM model considering manual operations. *Energy* 2021; 220: 119708.
13. Yu W, Kim IIY and Mechefske C. An improved similarity-based prognostic algorithm for RUL estimation using an RNN autoencoder scheme. *Reliability Eng Syst Saf* 2020; 199: 106926.
14. Elsheikh A, Yacout S and Ouali M-S. Bidirectional handshaking LSTM for remaining useful life prediction. *Neurocomputing* 2019; 323: 148–156.
15. Saxena A, Goebel K, Simon D, et al. *Damage propagation modeling for aircraft engine run-to-failure simulation*. In: Proceedings of the 2008 IEEE international conference on prognostics and health management (PHM 2008), Piscataway: IEEE, 2008, pp. 1–9.
16. Saxena A and Goebel K. PHM08 challenge data set, NASA ames prognostics data repository, 2008.
17. Peel L. *Data driven prognostics using a kalman filter ensemble of neural network models*. In: 2008 International Conference on Prognostics and Health Management, Piscataway: IEEE, 2008, pp. 1–6.
18. Gebraeel N, Lawley M, LI R, et al. Residual life distributions from component degradation signals: residual life distributions from component degradation signals: a bayesian approach. *IIE Trans* 2005; 37: 543–557.
19. Gebraeel N. Sensory-updated residual life distributions for components with exponential degradation patterns. *IEEE Trans Automation Sci Eng* 2006; 3: 382–393.
20. Chakraborty S, Gebraeel N, Lawley M, et al. Residual-life estimation for components with non-symmetric priors. *IIE Trans* 2009; 41: 372–387.
21. Elwany A and Gebraeel N. Sensor-driven prognostic models for equipment replacement and spare parts inventory. *IIE Trans* 2008; 40: 629–639.
22. Heimes FO. *Recurrent neural networks for remaining useful life estimation*. In: International Conference on Prognostics and Health Management, 2008, Piscataway: IEEE, 2008, pp. 1–6.
23. Wu J, Hu K, Cheng Y, et al. Data-driven remaining useful life prediction via multiple sensor signals and deep long short-term memory neural network. *ISA Trans* 2020; 97: 241–250.
24. Kingma D and Ba J. Adam: a method for stochastic optimization, international conference on learning representations, 2014.
25. Zhang Y, Hutchinson P, Lieven NAJ, et al. Remaining useful life estimation using long short-term memory neural networks and deep fusion. *IEEE Access* 2020; 8: 19033–19045.