

Article citation info:

Iv Y, Zhang Q, Chen A, Wen Z, Interval Prediction of Remaining Useful Life based on Convolutional Auto-Encode and Lower Upper Bound Estimation Eksploracja i Niezawodność – Maintenance and Reliability 2023; 25(2) <http://doi.org/10.17531/ein/165811>

## Interval Prediction of Remaining Useful Life based on Convolutional Auto-Encode and Lower Upper Bound Estimation

Indexed by:  
 Web of Science Group

Yi Lyu<sup>a,b,\*</sup>, Qichen Zhang<sup>b</sup>, Aiguo Chen<sup>b</sup>, Zhenfei Wen<sup>b</sup>

<sup>a</sup> Department of Computer, University of Electronic Science and Technology of China, Zhongshan Institute, China

<sup>b</sup> School of Computer Science and Engineering, University of Electronic Science and Technology of China, China

### Highlights


- A novel interval prediction model for RUL based on deep learning is proposed.
- The interval prediction model combines the LSTM and LUBE to make full use of the timing information in the degradation data.
- Experimental results show the RUL interval prediction performance has been significantly improved by the proposed method.

### Abstract

Deep learning is widely used in remaining useful life (RUL) prediction because it does not require prior knowledge and has strong nonlinear fitting ability. However, most of the existing prediction methods are point prediction. In practical engineering applications, confidence interval of RUL prediction is more important for maintenance strategies. This paper proposes an interval prediction model based on Long Short-Term Memory (LSTM) and lower upper bound estimation (LUBE) for RUL prediction. First, convolutional auto-encode network is used to encode the multi-dimensional sensor data into one-dimensional features, which can well represent the main degradation trend. Then, the features are input into the prediction framework composed of LSTM and LUBE for RUL interval prediction, which effectively solves the defect that the traditional LUBE network cannot analyze the internal time dependence of time series. In the experiment section, a case study is conducted using the turbofan engine data set CMAPSS, and the advantage is validated by carrying out a comparison with other methods.

### Keywords

remaining useful life, lower upper bound estimation, Long Short-Term Memory, prediction interval

This is an open access article under the CC BY license (<https://creativecommons.org/licenses/by/4.0/>) 

### 1. Introduction

Remaining useful life (RUL) prediction of important mechanical parts is particularly important for important equipment such as aerospace, large-scale production lines, ships, railway and highway transportation. Accurate RUL prediction provides assurance for formulating appropriate preventive maintenance and replacement strategies, which can not only ensure the operation of mechanical equipment under high reliability conditions, but also avoid waste caused by over frequency maintenance. Therefore, the RUL prediction of mechanical parts has been paid more and more attention by

scholars [17, 18]. With the development of material technology, the structure of various mechanical equipment is becoming more and more complex, and it is difficult to obtain accurate RUL through model driven method. Deep learning is widely used in data driven RUL prediction because of its strong nonlinear fitting ability and no prior knowledge [16]. Sateesh et al. [22] first applied the convolutional neural network (CNN) to predict the RUL of equipment, making full use of the feature extraction ability of CNN, and achieved good prediction results. Zhang et al. [30] took advantage of the feature that Long Short-

(\*) Corresponding author.

E-mail addresses:

Y. Lyu (ORCID: 0000-0001-5150-3482) [lvyyi913001@163.com](mailto:lvyyi913001@163.com), Q. Zhang (ORCID: 0009-0005-3906-5172) [202021080223@std.uestc.edu.cn](mailto:202021080223@std.uestc.edu.cn), A. Chen (ORCID: 0000-0002-3712-2349) [aigchen@uestc.edu.cn](mailto:aigchen@uestc.edu.cn), Z. Wen (ORCID: 0009-0000-6304-720X) [202121080938@std.uestc.edu.cn](mailto:202121080938@std.uestc.edu.cn)

Term Memory (LSTM) network can effectively extract time series information, and applied LSTM network to RUL prediction, further improving the prediction accuracy. Kong et al. [11] combined CNN with LSTM network to more fully explore the spatial and temporal characteristics of degradation data. Li et al. [13] proposed a directed acyclic graph network combined with CNN and LSTM, and used the sliding time window method to obtain samples to better extract features. Li et al. [15] proposed a deep learning-based remaining useful life prediction approach to solve partial sensor malfunction problem.

The above methods are mainly aimed at point prediction and achieve good results. Traditional point prediction cannot deal with the uncertainty in the system operation well and reduces the reliability of life prediction. Interval prediction can describe the possible upper and lower limits of the predicted value to describe the uncertainty of service life to cope with the risk. In practical projects, it is more important to study the confidence interval of life prediction for maintenance strategies based on reliability. Chen et al. [4] proposed a prediction interval estimation method based on improved fuzzy C-means algorithm and bidirectional short-term and short-term memory network to realize offline and online RUL interval prediction. Pang et al. [20] proposed an interval prediction strategy based on fuzzy information granularity and language description for RUL prediction of lithium batteries. Zhu et al. [32] proposed a Bayesian deep-active-learning framework for RUL interval prediction. Peng et al. [21] developed a Bayesian deep-learning-based for health prognostics with uncertainty quantification. For the research on interval prediction of RUL, the current main methods are Bootstrap method [25, 27], Bayesian method [7, 10] and lower upper bound estimation (LUBE) method [1, 28]. Bootstrap, Bayesian and other methods require data to meet the premise assumption of a certain distribution. They need to calculate the mean and variance according to the form of data distribution, involving the calculation of Jacobian matrix or Hessian matrix, which requires a huge amount of calculation [3]. LUBE method is a new prediction interval estimation method developed in recent years, which has much less computation than the statistical method of neural network [8]. LUBE method is a neural network (NN) based time series prediction interval direct estimation method (NN LUBE). Different from the traditional neural network based statistical prediction interval

estimation method, LUBE does not depend on the distribution of data, and outputs the upper and lower bounds directly through the neural network, so it has received extensive research and attention. However, the parameters of the neural network model in the LUBE method cannot be adjusted through the error based loss function and forward propagation like the traditional neural network structure [5]. Instead, the network model parameters can only be obtained through heuristic algorithms. For the time information contained in the degradation data in the life prediction, the traditional LUBE method cannot effectively use it, so it cannot accurately carry out interval prediction and reliability analysis.

In order to make full use of the timing information in the degradation data and predict the interval RUL of the equipment more reliably, this paper proposes a model structure called LSTM-LUBE, uses LSTM to replace the neural network of the traditional LUBE method, and constructs the optimization model structure of the interval loss function that can be used for gradient descent. In this paper, the RUL interval prediction method is proposed. First, the multi-dimensional variables detected by multiple sensors are coded into one dimension using Convolutional Auto-Encode (CAE) network as the input characteristics of the life prediction stage. Then, the life interval prediction is carried out by using the life prediction framework combining LSTM and LUBE. The main contributions and innovation points of this article are summarized below.

1. Use the CAE network structure to effectively extract the degradation characteristics related to the remaining service life of the system, and reduce the dimension and noise.

2. A more effective life interval prediction structure using the time series information of degraded data is proposed, and a differentiable loss function is constructed for this structure.

## 2. Theory

### 2.1. Interval prediction of remaining useful life

For the prediction of the remaining service life of products, most of the relevant work is focused on point prediction. However, in practice, especially for increasingly complex and sophisticated mechanical systems, interval estimation of the remaining service life of products is more meaningful than point estimation. At present, there are three main methods for interval prediction of remaining useful life:

(1) Interval prediction based on probability. The probabilistic interval prediction model requires high sample integrity and requires a large amount of statistical data for experiments. In practical projects, the sample is missing and the data is small, which is not applicable.

(2) Interval prediction based on neural network. Such methods include Bayesian, bootstrap, Delta, Mean Variance Estimation (MVE) and other methods. Such methods usually need to assume that the data satisfies a certain distribution, such as normal distribution or Gaussian distribution, and then calculate the mean and variance according to the distribution, and calculate the prediction interval. This kind of method often needs matrix calculation, and the calculation cost is large.

(3) Direct upper and lower bound interval estimation. This method does not need to make assumptions about the data distribution, and the calculation cost is small. The LUBE method uses neural network to output the upper and lower bounds of PI respectively. This method is direct to obtain the prediction interval. The neural network of LUBE method cannot construct suitable loss function for supervised learning, and can only use heuristic algorithm to obtain model parameters.

## 2.2. Convolutional Auto-Encode

Convolution neural network consists of a neural network composed of convolution and pooling [9]. Convolution acts as a filter, while pooling is to extract invariant features. Auto-encoder is a neural network composed of input layer, hidden layer and output layer [19]. Its structure is shown in Figure 1.

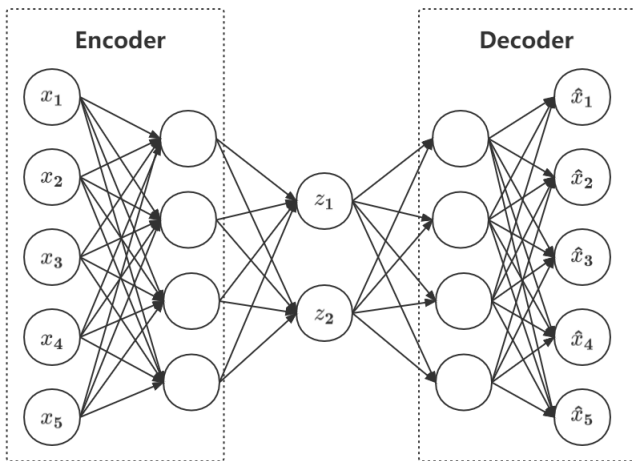


Fig.1. Structure of Auto-encoder network.

By using the mapping relationship between input layer and output layer, it realizes sample reconstruction and extracts features.

Due to the large amount of noise in the degenerated data of general mechanical components, it is difficult to extract features directly using CNN. Therefore, we use Convolutional Auto-Encode (CAE) network [2] to encode and decode through convolution networks and deconvolution networks to obtain the final features. The specific optimization formula is as follows:

$$h^k = \sigma(x \times \omega^k + b^k) \quad (1)$$

$$\hat{x} = \sigma(h^k \times \bar{\omega}^k + c) \quad (2)$$

$$E = \frac{1}{2n} \sum (x_i - y_i)^2 \quad (3)$$

EQ.1 and EQ.2 are the coding and decoding calculation processes of the convolutional self encoder respectively. Where,  $k$  represents the  $k$ th convolution kernel, and each convolution kernel is composed of parameters  $\omega^k$  and  $b^k$ , representing the feature vector extracted by the encoder.  $\hat{x}$  represents the output value after decoding and reconstruction. Finally, compare the Euclidean distance between the input samples and the final result obtained by feature reconstruction, as shown in EQ.3, and optimize the network parameters to obtain a complete CAE network.

## 3. Our method

### 3.1. LSTM-LUBE structure

To make full use of the time series data of the turbine engine data set and more accurately carry out interval prediction, this paper uses the framework of LSTM and LUBE to carry out bearing interval prediction. The data input mode of LSTM-LUBE is a degradation data set with a length of sliding window  $W$  after preprocessing; LSTM-LUBE has two outputs in each time step, corresponding to the lower and upper bounds of the prediction interval of the next time step. The calculation formula is:

$$\hat{L}_t = W_L \times h_{t-1} + b_L \quad (4)$$

$$\hat{U}_t = W_U \times h_{t-1} + b_U \quad (5)$$

Where,  $\hat{L}_t$  and  $\hat{U}_t$  are the upper and lower bounds of life prediction at time  $t$ ,  $W_L$  and  $W_U$  are the weight matrices corresponding to the upper and lower bounds respectively, and  $b_L$  and  $b_U$  are the offset matrices corresponding to the upper and lower bounds respectively.

The specific structure is shown in Figure 2. This structure uses LSTM to replace the networks used by LUBE, which effectively solves the defect that traditional LUBE networks

cannot analyze the internal time dependence of time series, and is more suitable for interval prediction analysis of degraded data. Different from the general LUBE method, LSTM-LUBE has a complex framework and many parameters, which makes it impossible to use the heuristic algorithm for interval prediction. Therefore, the key to this structure is to select a suitable loss function and use the gradient descent algorithm to optimize the interval prediction model.

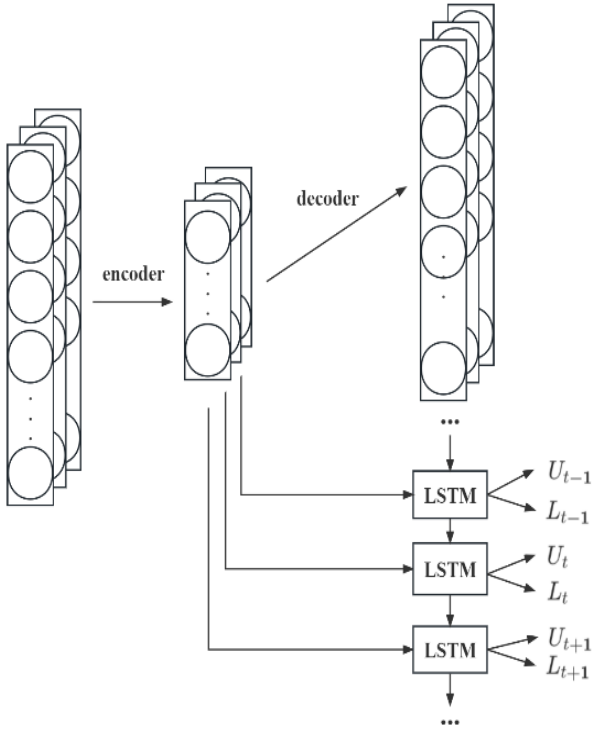


Fig.2. Network structure of LSTM-LUBE method.

### 3.2. Construction of loss function

Traditional interval prediction algorithms mostly adopt heuristic algorithms, such as particle swarm optimization algorithm [26] and genetic algorithm [24], combine SVM, ANN and other models, and use PICP, CWC and other evaluation indicators as optimization objectives to optimize the model and achieve interval prediction. However, most interval prediction evaluation indicators such as PICP and CWC are non-differentiable, and gradient descent algorithm cannot be used to optimize the model. Therefore, we need to find a differentiable loss function, and then derive the gradient descent to update the model parameters, optimize the model, and achieve interval prediction.

The construction of loss function shall meet the requirements of reliability and clarity [31]. Reliability refers to

the overall probability that the actual value falls into the prediction interval. The higher the coverage is, the more reliable the interval is; Clarity is used to measure the average width of the prediction interval. The smaller the prediction interval, the clearer the range of the prediction value. It is worth noting that these two indicators often exist in opposition. The smaller the interval width is, the harder it is to ensure that the coverage is kept at a good level, and vice versa. Therefore, the loss function constructed should comprehensively consider two requirements. In addition, to optimize the target interval towards the target value, a penalty should be added to the extent that the target value deviates from the interval. The specific loss function is as follows:

$$f(\theta) = \sum_{i=1}^n \left| y_i - \frac{U_i + L_i}{2} \right| + \gamma_i * e^{\frac{d_i}{|U_i - L_i|}} * \max(0, \mu - \sum_{i=1}^n \gamma_i) + \lambda |U_i - L_i| \quad (6)$$

$$\gamma_i = \begin{cases} 0 & y_i \in [U_i, L_i] \\ 1 & y_i \notin [U_i, L_i] \end{cases} \quad (7)$$

$$d_i = \left| y_i - \frac{U_i + L_i}{2} \right| - \frac{|U_i - L_i|}{2} \quad (8)$$

Where,  $y_i$  is the true RUL value,  $U_i$  and  $L_i$  are the upper and lower limits of the prediction interval, and  $\lambda$  is the proportional coefficients used to adjust the weights of the interval coverage and interval width.  $d$  is the distance between the interval and the true value,  $\mu$  is determined by the confidence interval. When the confidence level is 95%,  $\mu$  is 0.95. When the prediction results reach the confidence level, the loss function is mainly optimized to narrow the interval width and make the real value closer to the median of the interval. When the prediction result does not reach the confidence interval, the deviation degree is punished by exponential growth to optimize the interval towards high coverage.

The flow of our method selection is shown in Figure 3. It is mainly divided into three stages. First, data preprocessing is performed, then CAE network is used to extract features, and finally LSTM-LUBE network is used to obtain prediction interval and evaluation index. Next, the CMAPSS engine data set will be used to verify the effectiveness of this method.

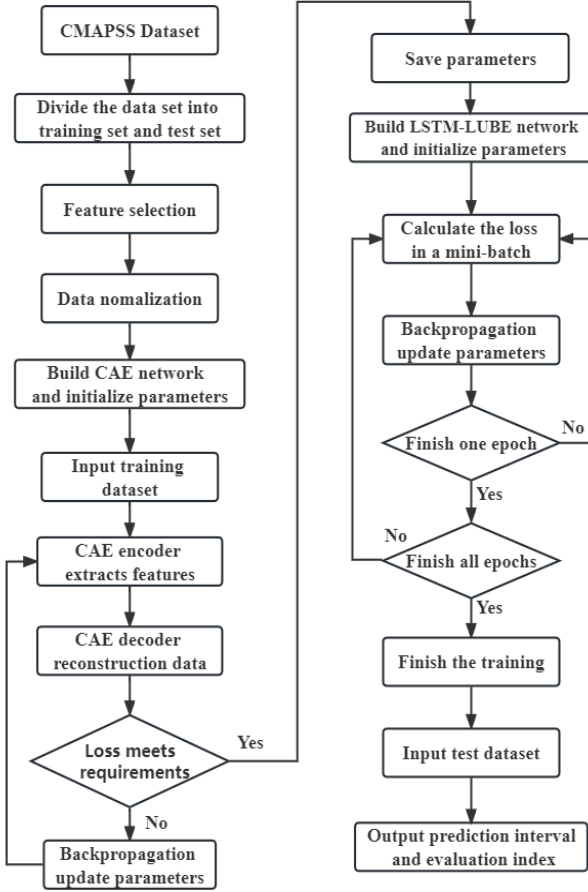


Fig.3. Flow chart of the proposed RUL interval prediction method.

## 4. Experiment

### 4.1. Dataset and preprocessing

This paper takes the turbofan engine as the research object, and adopts the public dataset of aircraft engine simulation state monitoring provided by NASA Prognostics Center of Excellence (PCoE). It simulates different engine operating environments through CMAPSS software and outputs condition monitoring data of various components in the degradation process [23]. It is widely used as benchmark data in the research field of equipment residual life prediction. The dataset has 4 subsets. The training set and test set select 21 sensor signals and three operating parameters. The training set includes the monitoring data of the entire degradation process of the turbine engine from the initial state to the first failure. The test set records the monitoring data from the initial state to a certain period before the failure. The remaining life set gives the remaining period of the test set engine from operation to failure

[14]. Specific data set information is shown in Table 1.

Table 1. Sensors of turbofan engine.

Num	Symbol	Description	Units	trend
1	T2	Total temperature at fan inlet	° R	~
2	T24	Total temperature at LPC outlet	° R	↑
3	T30	Total temperature at HPC outlet	° R	↑
4	TSO	Total temperature at LPT outlet	° 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	↓

Due to the diversity of engine performance variable selection, the data dimension collected by the sensor is not uniform. The data needs to be normalized before being input into the model, and the processed data will not change the degradation characteristics of the sensor data. This paper uses standardization to process the data of each sensor. The specific formula is as follows:

$$x'_i = \frac{x_i - \mu_i}{\sigma_i} \quad (9)$$

In EQ.9,  $\mu_i$  and  $\sigma_i$  are the average value and standard deviation of the  $i$ th characteristic signal  $x_i$  in  $x$ .

The normalized data is input into CAE network by the sliding window strategy according to the fixed length [6]. After a series of parameter adjustments, it is found that the training effect is better when the window size is 30. Therefore, the sliding window width  $T_W$  is set to 30. The dataset is divided into:

$$X_{i:i+T_W-1} = \{X_i, X_{i+1}, X_{i+2}, \dots, X_{i+T_W-1}\}$$

$$= \begin{bmatrix} x_{i,1} & x_{i,2} & \dots & x_{i,m} \\ x_{i+1,1} & x_{i+1,2} & \dots & x_{i+1,m} \\ \vdots & \vdots & & \vdots \\ x_{i+T_W-1,1} & x_{i+T_W-1,2} & \dots & x_{i+T_W-1,m} \end{bmatrix} \in R^{m \times T_W}$$

In 21 kinds of sensor data, some data remain unchanged throughout the life cycle. In order to prevent the interference of irrelevant data and reduce the training time, We select the 14

dimensional sensor data of the 2nd, 3rd, 4th, 7th, 9th, 11th, 12th, 13th, 14th, 15th, 17th, 20th, 21st as the original data input model.

#### 4.2. Feature extraction and selection

Since CAE network has the advantage of self extracting features without obtaining labels, and there is no parameter redundancy, we choose CAE network as the feature extraction module of the structure. The main structure of the module is composed of three convolutional coding layers and three deconvolution decoding layers. Since the degenerate data is one-dimensional data, our convolution code is one-dimensional convolution. The specific network structure parameters are shown in Table 2.

Table 2. Parameters of CAE network structure

Layer Type	Output size	Activation function
Convolutional Layer	(30,14,16)	RELU
Pooling layer	(15,7,16)	
Convolutional Layer	(14,6,16)	RELU
Pooling layer	(7,3,16)	
Convolutional Layer	(3,3,16)	RELU
Pooling layer	(1,1,1)	Sigmoid
Unpooling layer	(3,3,16)	
Convolutional Layer	(7,3,16)	RELU
Unpooling layer	(14,6,16)	
Convolutional Layer	(15,7,16)	RELU
Unpooling layer	(30,14,16)	
Convolutional Layer	(30,14,1)	Sigmoid

The characteristics of the output of the trained CAE network are shown in Figure 4.

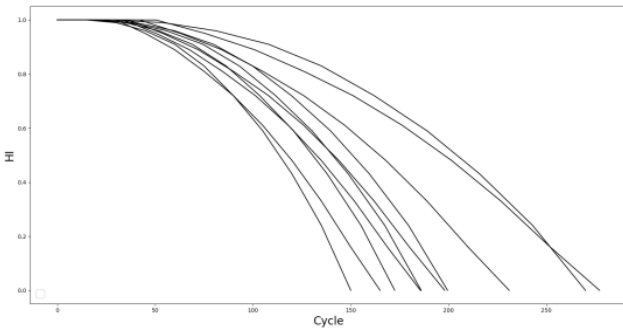


Fig.4. Features extracted from training set.

It can be seen that, in the early stage of the training cycle, the characteristics do not change significantly, but only drop sharply when approaching the failure period. This is consistent with the degradation trend of aero-engine characteristics that we recognize. That is to say, the extracted feature vector can be used as the basic feature of RUL prediction.

#### 4.3. RUL interval prediction

We use the LSTM-LUBE method mentioned above to predict the RUL interval. The structure consists of three layers of LSTM and four layers of full connection layer. The LSTM input is convolutional encoder output, the output vector size is set to 64, and the number of neurons in the four layers of full connection layer is 32, 16, 8, and 2 respectively. The loss function is shown in EQ.6. In order to balance the interval width and interval coverage, the proportional coefficient  $\lambda$  set to 0.5, use the random gradient descent algorithm to update the model parameters, and set the weight update batch to 64. Figure 5 shows the loss descent in the training process. It can be seen that the loss function tends to be stable after 90 iterations. Therefore, in order to fully train the model, we set the total number of iterations to 120.

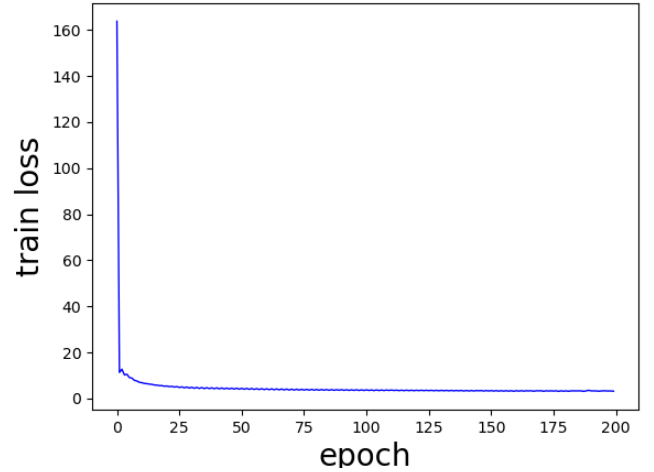


Fig.5. Loss of training data set.

#### 4.4 evaluation index

In this experiment, the commonly used interval evaluation indicators PICP, NPIW and CWC are used as evaluation indicators for model comparison. The relevant evaluation index formula is as follows.

The Prediction Interval Coverage Probability (PICP) indicator is used to evaluate the reliability of interval prediction. It reflects the overall probability level that the actual prediction value falls within the prediction interval.

$$\left\{ \begin{array}{l} PICP = \frac{1}{n} \sum_{i=1}^n c_i \\ c_i = \begin{cases} 0 & y_i \notin [L_i, U_i] \\ 1 & y_i \in [L_i, U_i] \end{cases} \end{array} \right. \quad (10)$$

Where,  $L_i$  and  $U_i$  is the lower bound and upper bound of the  $i$ th PI. PICP is an index for measuring PI calibration with preset

confidence. Theoretically, PI can be trusted when  $PICP \geq (1 - \alpha)\%$ .

Prediction Interval Normalized Average Width (PINAW) is used to evaluate the clarity of the prediction interval. Interval width is also an important indicator to evaluate the quality of an interval.

$$PINAW = \frac{1}{nR} \sum_{i=1}^n (U_i - L_i) \quad (11)$$

Where,  $n$  is the number of samples,  $U_i$  and  $L_i$  represent the upper and lower bounds of prediction, and  $R$  is the difference between the maximum value and the minimum value in the prediction samples, which is used to normalize the indicators.

Coverage Width Criterion (CWC) is an evaluation index of prediction interval proposed by Khosravi et al. [1]. It combines the PICP index of coverage evaluation with the PINAW index of interval width evaluation to construct a comprehensive evaluation index of prediction interval.

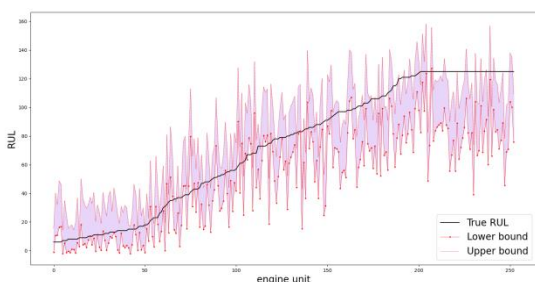
$$CWC = \begin{cases} PINAW & (PICP \geq \mu) \\ PINAW + \exp(-\eta(PICP - \mu)) & (PICP < \mu) \end{cases} \quad (12)$$

Where,  $\mu$  And  $\eta$  are super parameters,  $\mu$  is the preset interval prediction nominal confidence level PINC, and  $\eta$  is the penalty parameter, which is used to punish the case that the PICP index is lower than PINC.

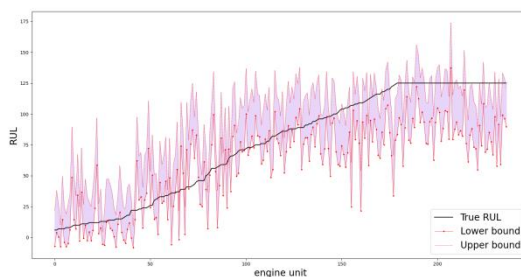
#### 4.5 Experimental design

1) *RUL interval prediction results:* In the experiment, we selected FD001 and FD003 datasets for life span prediction. To ensure the accuracy of the experiment, we conducted five experiments on each group of datasets and took the average of the prediction results. The prediction results of the two subsets are sorted according to the size of the remaining service life, as shown in Figure 6.

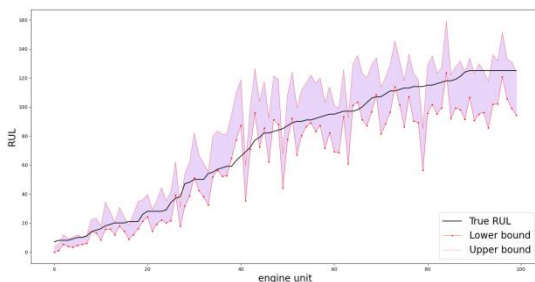
It can be seen from Figure 6 that most of the prediction intervals of our method on CMAPSS datasets contain real data, and the smaller the remaining life, the narrower the interval width, and the higher the clarity of interval prediction.



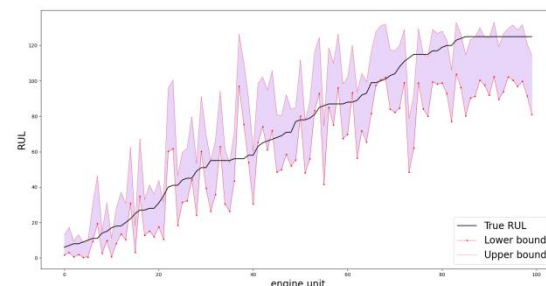
(a) Comparison of FD001 Dataset RUL prediction interval and Real remaining life



(b) Comparison of FD002 Dataset RUL prediction interval and Real remaining life



(c) Comparison of FD003 Dataset RUL prediction interval and Real remaining life



(d) Comparison of FD004 Dataset RUL prediction interval and Real remaining life

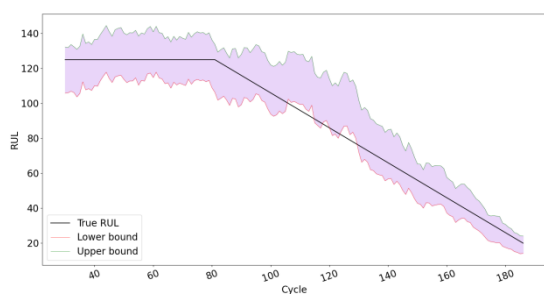
Fig.6.

Figure 7 shows the prediction results of engine units in FD001 and FD003 data sets. (a) and (b) are the prediction results of engines 24 and 100 in FD001, and (c) and (d) are the

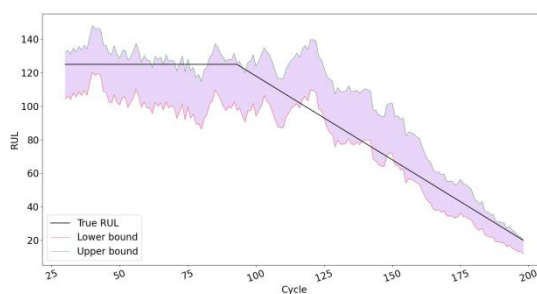
prediction results of engines 24 and 100 in FD003. It can be seen from the figure that when the cycle period is small, the width of interval prediction is large. With the increase of cycle

period, the coverage and width of interval prediction are

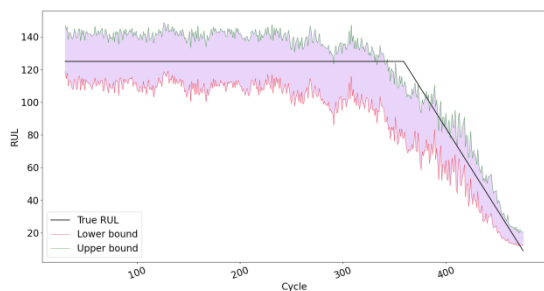
optimized towards high reliability and high definition.



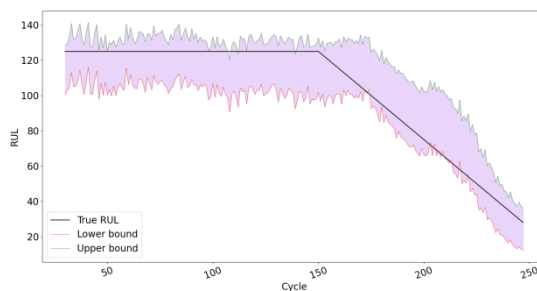
(a) The 24th engine test unit in FD001



(b) The 100th engine test unit in FD001



(c) The 24th engine test unit in FD003



(d) The 100th engine test unit in FD003

Fig.7. Comparison between actual value and prediction interval of remaining life of different engines.

2) *comparison experiment*: Evaluate and compare models based on EQ 10-12 and run times. Compare our method with

LUBE algorithm and Bootstrap method on four engine datasets, We set the confidence level  $\mu$  to 90%. The experimental index results are shown in Table 3.

TABLE 3. Comparison of interval prediction methods.

	evaluation index	FD001	FD002	FD003	FD004
Bootstrap+SVR [12]	PICP(%)	85.01	87.56	83.25	86.32
	PINAW(%)	<b>16.67</b>	<b>14.25</b>	<b>17.97</b>	<b>18.23</b>
	CWC	8.94	9.58	4.38	5.64
	Time(s)	87.68	90.29	75.26	86.39
LUBE+CSS [29]	PICP(%)	90.07	91.26	89.75	90.68
	PINAW(%)	30.02	32.56	28.14	29.77
	CWC	4.32	5.29	5.26	5.14
	Time(s)	<b>65.42</b>	<b>71.22</b>	<b>67.23</b>	<b>68.97</b>
Our proposed	PICP(%)	<b>93.32</b>	<b>92.45</b>	<b>92.68</b>	<b>93.11</b>
	PINAW(%)	21.16	20.31	18.54	19.69
	CWC	<b>1.03</b>	<b>1.48</b>	<b>0.83</b>	<b>2.14</b>
	Time(s)	98.26	97.15	89.59	85.24

It can be seen from the data in Table III that our method has better reliability and clarity than the LUBE method using the heuristic algorithm. The PICP average is about 92.82%,

effectively ensuring the reliability of the prediction interval. At the same time, the PINAW average is about 20.14, which is also better than the traditional LUBE algorithm. Compared with



Bootstrap, although Bootstrap has smaller PINAW, its interval coverage rate (PICP) is far less than our method, and the reliability of the prediction interval is difficult to guarantee. By comparing the comprehensive evaluation index CWC, we can also see that the CWC index of our method is smaller, and the fluctuation is more stable, which is generally better than the Bootstrap and LUBE+CSS methods.

## 5. Conclusion

- (1) The neural network structure based on convolutional self encoder can adaptively extract the feature vector of the engine through encoding and decoding, and effectively reduce the dimension of the input multi-dimensional time data, reducing the training time.
- (2) The residual life interval prediction model based on LSTM-

## Acknowledgements

This research was funded by the Major Special Projects of Zhongshan (200824103628344) and the Guangdong Science and Technology Program (2021A0101180005).

## References

1. Akbari M, Kabir H M D, Khosravi A, Nasirzadeh F. Ann-based lube model for interval prediction of compressive strength of concrete. *Iranian Journal of Science and Technology, Transactions of Civil Engineering* 2021; 1-11, <https://doi.org/10.1007/s40996-021-00684-x>.
2. Badrinarayanan V, Kendall A, Cipolla R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence* 2017; 39(12): 2481-2495, <https://doi.org/10.1109/TPAMI.2016.2644615>.
3. Chen C, Lewis FL, Li B. Homotopic policy iteration-based learning design for unknown linear continuous-time systems. *Automatica* 2022; 138:110153, <https://doi.org/10.1016/j.automatica.2021.110153>.
4. Chen C, Lu N, Jiang B, Zhu Z H. Prediction interval estimation of aeroengine remaining useful life based on bidirectional long short-term memory network. *IEEE Transactions on Instrumentation and Measurement* 2021; 70: 1-13, <https://doi.org/10.1109/TIM.2021.3126006>.
5. Chen C, Xie L, Jiang Y, Xie K, Xie S. Robust Output Regulation and Reinforcement Learning-Based Output Tracking Design for Unknown Linear Discrete-Time Systems. *IEEE Transactions on Automatic Control* 2023; 68:2391–2398, <https://doi.org/10.1109/TAC.2022.3172590>.
6. Chen X, Xiao H, Guo Y, Kang Q. A multivariate grey RBF hybrid model for residual useful life prediction of industrial equipment based on state data. *International Journal of Wireless and Mobile Computing* 2016; 10(1): 90-96, <https://doi.org/10.1504/IJWMC.2016.075230>.
7. Dong L, Zhou W, Zhang P, Liu G, Li W. Short-term photovoltaic output forecast based on dynamic bayesian network theory. *Proceedings of the CSEE* 2013; 33(S1): 38-45.
8. Khosravi A, Nahavandi S, Creighton D, Atiya A F. Lower upper bound estimation method for construction of neural network-based prediction intervals. *IEEE transactions on neural networks* 2010; 22(3): 337-346, <https://doi.org/10.1109/TNN.2010.2096824>.
9. Kim Y. Convolutional neural networks for sentence classification. *Eprint Arxiv* 2014, <https://doi.org/10.3115/v1/D14-1181>.
10. Kobayashi K, Kaito K, Lethanh N, A bayesian estimation method to improve deterioration prediction for infrastructure system with markov chain model. *International Journal of Architecture, Engineering and Construction* 2012; 1(1): 1-13, <https://doi.org/10.7492/IJAEC.2012.001>.
11. Kong Z, Cui Y, Xia Z, Lv H. Convolution and long short-term memory hybrid deep neural networks for remaining useful life prognostics. *Applied Sciences* 2019; 9(19): 4156, <https://doi.org/10.3390/app9194156>.
12. Lee B H. Bootstrap Prediction Intervals of Temporal Disaggregation. *Stats* 2022; 5(1): 190-202, <https://doi.org/10.3390/stats5010013>.
13. Li J, Li X, He D. A directed acyclic graph network combined with cnn and lstm for remaining useful life prediction. *IEEE Access* 2019; 7: 75464-75475, <https://doi.org/10.1109/ACCESS.2019.2919566>.

LUBE can make full use of the time series information in the degraded data, and improve the prediction accuracy by constructing a differentiable interval prediction loss function and updating the model parameters with gradient descent.

(3) The LSTM-LUBE method is tested on the CMAPSS dataset and compared with other interval prediction methods. The results show that the proposed method is effective.

## 6. Feature work

The future research work can further optimize the loss function, and only calculate the interval width of the actual value in the interval when optimizing the interval width. And further verify the effectiveness and universality of the method in more datasets.

14. Li X, Ding Q, Sun J Q. Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliability Engineering & System Safety* 2018; 172: 1-11, <https://doi.org/10.1016/j.ress.2017.11.021>.
15. Li X, Xu Y, Li N, Lei Y. Remaining useful life prediction with partial sensor malfunctions using deep adversarial networks. *IEEE/CAA Journal of Automatica Sinica* 2022; 10(1): 121-134, <https://doi.org/10.1109/JAS.2022.105935>.
16. Lyu Y, Gao J, Chen C, Jiang Y, Li H, Chen K, Zhang Y. Joint model for residual life estimation based on Long-Short Term Memory network. *Neurocomputing* 2020; 410: 284-294, <https://doi.org/10.1016/j.neucom.2020.06.052>.
17. Lyu Y, Jiang Y, Zhang Q, Chen C. Remaining useful life prediction with insufficient degradation data based on deep learning approach. *Eksploracja i Niezawodność* 2021; 23(4): 745-756, <https://doi.org/10.17531/ein.2021.4.17>.
18. Lyu Y, Zhang Q, Wen Z, Chen A. Remaining Useful Life Prediction Based on Multi-Representation Domain Adaptation. *Mathematics* 2022; 10(24): 4647, <https://doi.org/10.3390/math10244647>.
19. Michelucci U. An introduction to autoencoders. 2022, [https://doi.org/10.1007/978-1-4842-8020-1\\_9](https://doi.org/10.1007/978-1-4842-8020-1_9).
20. Pang X, Zhao Z, Wen J, Jia J, Shi Y, Zeng J, Dong Y. An interval prediction approach based on fuzzy information granulation and linguistic description for remaining useful life of lithium-ion batteries. *Journal of Power Sources* 2022; 542: 231750, <https://doi.org/10.1016/j.jpowsour.2022.231750>.
21. Peng W, Ye Z S, Chen N. Bayesian deep-learning-based health prognostics towards prognostics uncertainty. *IEEE Transactions on Industrial Electronics* 2020; 67: 2283-2293, <https://doi.org/10.1109/TIE.2019.2907440>.
22. Sateesh B G, Zhao P, Li X L. Deep convolutional neural network based regression approach for estimation of remaining useful life. in *International conference on database systems for advanced applications*. Springer 2016; pp. 214–228, [https://doi.org/10.1007/978-3-319-32025-0\\_14](https://doi.org/10.1007/978-3-319-32025-0_14).
23. Saxena A, Goebel K, Simon D, Eklund N. Damage propagation modeling for aircraft engine run-to-failure simulation. in *2008 International Conference on Prognostics and Health Management* 2008; pp.1–9, <https://doi.org/10.1109/PHM.2008.4711414>.
24. Sharghi E, Paknezhad N J, Najafi H. Assessing the effect of emotional unit of emotional ANN (EANN) in estimation of the prediction intervals of suspended sediment load modeling. *Earth Science Informatics* 2021; 14(1): 201-213, <https://doi.org/10.1007/s12145-020-00567-1>.
25. Sheng C, Zhao J, Wang W, Leung H. Prediction intervals for a noisy nonlinear time series based on a bootstrapping reservoir computing network ensemble. *IEEE Transactions on neural networks and learning systems* 2013; 24(7): 1036-1048, <https://doi.org/10.1109/TNNLS.2013.2250299>.
26. Shrivastava N A, Khosravi A, Panigrahi B K. Prediction interval estimation of electricity prices using PSO-tuned support vector machines. *IEEE Transactions on Industrial Informatics* 2015; 11(2): 322-331, <https://doi.org/10.1109/TII.2015.2389625>.
27. Wang X, Ghidaoui M S, Lin J. Confidence interval localization of pipeline leakage via the bootstrap method. *Mechanical Systems and Signal Processing* 2022; 167: 108580, <https://doi.org/10.1016/j.ymssp.2021.108580>.
28. Wang Y, Tang H, Wen T, Ma J. A hybrid intelligent approach for constructing landslide displacement prediction intervals. *Applied Soft Computing* 2019; 81: 105506, <https://doi.org/10.1016/j.asoc.2019.105506>.
29. Wu Y K, Su P E, Wu T Y, Hong J S, Hassan M Y. Probabilistic wind-power forecasting using weather ensemble models. *IEEE Transactions on Industry Applications* 2018; 54(6): 5609-5620, <https://doi.org/10.1109/TIA.2018.2858183>.
30. Zhang Y, Xiong R, He H, Pecht M G. Long short-term memory recurrent neural network for remaining useful life prediction of lithiumion batteries. *IEEE Transactions on Vehicular Technology* 2018; 67(7): 5695-5705, <https://doi.org/10.1109/TVT.2018.2805189>.
31. Zhou M, Wang B, Guo S, Watada J. Multi-objective prediction intervals for wind power forecast based on deep neural networks. *Information Sciences* 2021; 550: 207-220, <https://doi.org/10.1016/j.ins.2020.10.034>.
32. Zhu R, Chen Y, Peng W, Ye Z S. Bayesian deep-learning for RUL prediction: An active learning perspective. *Reliability Engineering & System Safety* 2022; 228: 108758, <https://doi.org/10.1016/j.ress.2022.108758>.