Aeroengine Performance Parameter Prediction Based on Improved Regularization Extreme Learning Machine

*CAO Yuyuan**, *ZHANG Bowen*, *WANG Huawei*

College of Civil Aviation, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, P. R. China

(Received 13 May 2020; revised 27 July 2020; accepted 6 December 2020)

Abstract: Performance parameter prediction technology is the core research content of aeroengine health management, and more and more machine learning algorithms have been applied in the field. Regularized extreme learning machine (RELM) is one of them. However, the regularization parameter determination of RELM consumes computational resources, which makes it unsuitable in the field of aeroengine performance parameter prediction with a large amount of data. This paper uses the forward and backward segmentation (FBS) algorithms to improve the RELM performance, and introduces an adaptive step size determination method and an improved solution mechanism to obtain a new machine learning algorithm. While maintaining good generalization, the new algorithm is not sensitive to regularization parameters, which greatly saves computing resources. The experimental results on the public data sets prove the above conclusions. Finally, the new algorithm is applied to the prediction of aero-engine performance parameters, and the excellent prediction performance is achieved.

Key words: extreme learning machine; aeroengine; performance parameter prediction; forward and backward segmentation algorithms

CLC number: V263.6; TH17; TP277 **Document code:** A **Article ID:** 1005-1120(2021)04-0545-15

0 Introduction

Due to the harsh working environment, large number of parts and complicated internal structure of aeroengines, performance degradation inevitably occurs during operation. Therefore, it is important to take some measures to get these signs of degradation in advance to get the rest of the engine's life, hence the engine performance parameter prediction has been the research highlights. The physical model of engines and the correlation between several parts are complicated, so it is very difficult to establish the model to predict the parameters. In recent years, with the development of machine learning methods, data-driven prediction methods have attracted more and more researchers' attention. Data-based methods do not require complex research models, and the accuracy of their predictions depends heavily on historical data. In the field of engine health detection, more and more data-based methods have been applied[1-3], and these studies also prove that the correct use of data-based methods can effectively improve the accuracy of diagnosis.

Artificial neural network (ANN)[4] has excellent nonlinear mapping function and is intensely suitable for complex fault diagnosis, so it has been widely studied. Yuan et al.[5] used the long short-term memory (LSTM) network for remaining useful life (RUL) prediction. Janssens et al.[6] adopt convolutional neural networks for fault diagnosis of rotating machinery. Qu et al.[7] used a stacked denoising auto-encoder (SDA) to solve aero-engine sensor fault diagnosis.

The extreme learning machine (ELM) is a new neural network training approach represented by Huang et al.[8] Due to its rapid learning speed

*Corresponding author, E-mail address: caoyuyuan@nuaa.edu.cn.

How to cite this article: CAO Yuyuan, ZHANG Bowen, WANG Huawei. Aeroengine performance parameter prediction based on improved regularization extreme learning machine[J]. Transactions of Nanjing University of Aeronautics and Astronautics, 2021, 38(4): 545-559.

http://dx.doi.org/10.16356/j.1005-1120.2021.04.002

and better performance, it has been widely utilized in classification, regression and other fields. Ye et al.[9] proposed to combine QR decomposition into an incremental ELM (IELM) to obtain QR-IELM in 2015. In 2017, the Gram-Schmidt process was introduced into IELM to obtain GSI-ELM[10]. Cao et al.[11] applied the entropy theory into ELM to obtain ATELM. Zhao et al.[12] also provided to suggest householder transformation and gave spins into ELM to accelerate its solution process. In addition to the improvement for the solution process, some other scholars combined ELM with other models to make improved methods suitable for certain problems. Nobrega et al.[13] combined the Kalman filter with ELM for regression problems and achieved positive results. Pacheco et al.[14] made certain breakthroughs by combining the restricted Boltzmann machine with ELM for classification problems. Anwesha et al.[15] combined the autoencoder (AE) with ELM to get a network with excellent performance.

Another group of scholars utilized the improved ELM in the field of engine fault diagnosis and achieved favorable effect. Jiang et al.[16] adopt multi-class Bayesian ELM (BELM) for engine gas-path fault diagnosis. Feng et al.[17] used multi-layer kernel ELM (KELM) for the aero-engine fault diagnosis. Lu et al.[18] employed the distributed ELM for engine fault diagnosis and achieved positive outcome. Similarly, Zhao et al.[19] employed a soft ELM for engine fault diagnosis. And more applications for ELM are utilized to estimate the RUL of complex machines like aeroengines[20-22].

Among these improved methods, the regularized ELM (RELM), provided by Deng et al.[23] in 2009, performs superior to ELM in many problems. But RELM has poor prediction performance on the time series prediction problem of SINC function in experiment, which makes RELM not applicable in some practical scenarios. To address this issue, many scholars have studied in the construction of regularization terms and training algorithms. In construction of regularization terms, Luo et al.[24] proposed a L_1-L_2 mixed regularization in 2016. Later, Li et al.[25] provided the Laplacian twin ELM, Yi et al.[26] represented a linear combination of several regularization terms and obtained an adaptive regularization term, and Inaba et al.[27] constructed a distributed regularized ELM in 2018. These RELM improvements have achieved excellent performance. In terms of training approaches, Ma et al.[28] utilized the Lagrangian algorithm to train ELM in 2019. Mahmooda et al.[29] used forward-backward splitting algorithm for training L_1-RELM for the first time. In 2019, Song et al.[30] utilized the alternating direction method of multipliers (ADMMs) for L_1-RELM training, and obtained the online ELM.

This paper contributes the forward-backward segmentation (FBS) algorithm[31] for the training of ELM with L_2 regularization (L_2-RELM), adopts a new calculation method to determine the step size, and obtains an algorithm with fewer number of training iterations. On this basis, this paper further improves the solution mechanism so that the number of training iterations is reduced again and the accuracy is reduced in an acceptable range. This study successfully overcomes the shortcomings of RELM in SINC data set and other time series prediction problems. The two represented algorithms are used for engine performance parameter prediction. Both algorithms have achieved better prediction performance compared with RELM.

1 Related Work

1.1 ELM

Assume the existing data set to be trained is $(\boldsymbol{x}, \boldsymbol{t})$ with the dimension of $n \times (m+N)$, where $\boldsymbol{x}$ represents the sample input data, $\boldsymbol{t}$ the sample output data, n the sample number, m the number of input features, N the dimension of the output data, and M the number of neurons in the hidden layer. Besides, the connection weight of the input layer to the hidden layer of the ELM represents $\boldsymbol{W}$ with the dimension of $M \times m$. The threshold represents $\boldsymbol{B}$ with the dimension of $M \times 1$. The hidden layer activation function is expressed as $g(\boldsymbol{x})$, and the hidden layer output can be expressed as

$$\boldsymbol{H} = g(\boldsymbol{W}\boldsymbol{x}^{\mathrm{T}} + \boldsymbol{B}) \tag{1}$$

Next, the connection weight $\boldsymbol{\beta}$ of the hidden layer to the output layer is calculated. In the network training phase, the network output is known, that means, T is known, $\boldsymbol{\beta}$ is to be calculated, and solving $\boldsymbol{\beta}$ is equivalent to solving the optimal loss function as follows

$$\min_{\beta}\|T-\boldsymbol{t}\|_2^2=\min_{\beta}\|\boldsymbol{H}\boldsymbol{\beta}-\boldsymbol{t}\|_2^2 \tag{2}$$

The final solution of $\boldsymbol{\beta}$ is not difficult to obtain by least squares method, shown as

$$\boldsymbol{\beta}=\boldsymbol{H}^{\dagger}T \tag{3}$$

where $\boldsymbol{H}^{\dagger}$ represents the Moore-Penrose generalized inverse of $\boldsymbol{H}$.

1.2 L$_2$-RELM

From Eq.(2), the solution of ELM is a process of minimizing empirical risk, but this solution may report over-fitting problems in some cases. Deng et al.[23] combined structural risk terms to avoid over-fitting problems, and updated the loss function of ELM as follows

$$\min_{\beta}\frac{1}{2}\|\boldsymbol{H}\boldsymbol{\beta}-\boldsymbol{t}\|_2^2+\frac{C}{2}\|\boldsymbol{\beta}\|_2^2 \tag{4}$$

where C is a regularization parameter, and the final solution of Eq.(4) is obtained according to the least squares method, shown as

$$\boldsymbol{\beta}=(H^{\mathrm{T}}H+IC)^{-1}H^{\mathrm{T}}\boldsymbol{t} \tag{5}$$

where $\boldsymbol{I}$ is an identity matrix whose dimension is the number of neurons in the hidden layer. Generally, the solution obtained by Eq.(5) is sparse and has better performance than the solution obtained by Eq.(4). L$_2$-RELM does not increase too much in computational complexity, so it is a perfect choice in practical applications.

2 Improvement of ELM

2.1 ELM combined with FBS

The FBS algorithm was developed by in 2016, which is applicable for the extremum problem of separable convex functions. FBS is generally adopted to calculate the following problems

$$\min h(\boldsymbol{x})=f(\boldsymbol{x})+g(\boldsymbol{x}) \tag{6}$$

where $f(\boldsymbol{x})$, $g(\boldsymbol{x})$ are all convex functions. It is not difficult to get Algorithm 1 to calculate Eq.(6).

Algorithm 1 Forward-backward splitting

While not converged do

$$\hat{\boldsymbol{x}}_{k+1}=\boldsymbol{x}_k-\tau_k\nabla f(\boldsymbol{x}_k)$$

$$\boldsymbol{x}_{k+1}=\mathrm{pro}x_g(\hat{\boldsymbol{x}}_{k+1},\tau_k)=\arg\min\tau_k g(\boldsymbol{x})+\frac{1}{2}\|\boldsymbol{x}-\hat{\boldsymbol{x}}_{k+1}\|_2^2$$

End

In Algorithm 1, $\boldsymbol{x}$ is the solution to be calculated and τ the step factor. In general, in order to ensure the final convergence, the value of the step factor needs to meet the following condition[31]

$$\tau<\frac{2}{L(\nabla f)} \tag{7}$$

where $L(\nabla f)$ is the spectral radius of $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$, such as when $f=1/2\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_2^2$.

This study adopts the FBS algorithm to solve L$_2$-RELM. At this time, the problem that FBS needs to solve is transformed to Eq.(4), which is split into the form suitable for FBS solution, shown as

$$f(\boldsymbol{\beta})=\frac{1}{2}\|H\boldsymbol{\beta}-\boldsymbol{t}\|_2^2 \tag{8}$$

$$g(\boldsymbol{\beta})=\frac{C}{2}\|\boldsymbol{\beta}\|_2^2 \tag{9}$$

Thus, the forward step in Algorithm 1 is calculated as follows

$$\hat{\boldsymbol{\beta}}_{k+1}=\boldsymbol{\beta}_k-\tau_k\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{H}\boldsymbol{\beta}_k-\boldsymbol{t}) \tag{10}$$

The backward step is calculated as follows

$$\boldsymbol{\beta}_{k+1}=\arg\min\frac{\tau_k C}{2}\|\boldsymbol{\beta}\|_2^2+\frac{1}{2}\|\boldsymbol{\beta}-\hat{\boldsymbol{\beta}}_{k+1}\|_2^2 \tag{11}$$

Eq.(11) can be calculated using the least squares method, shown as

$$\boldsymbol{\beta}_{k+1}=[\mathrm{diag}(\tau_k C)+\boldsymbol{I}]^{-1}[\boldsymbol{\beta}_k-\tau_k\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{H}\boldsymbol{\beta}_\kappa-\tau)] \tag{12}$$

This study solves L$_2$-RELM using FBS (called FELM), which is summarized in Algorithm 2.

Algorithm 2 FELM

Initialize C, $\boldsymbol{\beta}$, τ, $k=1$, and set the maximum number of iterations K

While $k<K$ or not converged do

$$\hat{\boldsymbol{\beta}}_{k+1}=\boldsymbol{\beta}_k-\tau_k H^{\mathrm{T}}(\boldsymbol{H}\boldsymbol{\beta}_k-\boldsymbol{t})$$

$$\boldsymbol{\beta}_{k+1}=[\mathrm{diag}(\tau_k C)+\boldsymbol{I}]^{-1}\hat{\boldsymbol{\beta}}_{k+1}$$

$$k=k+1$$

End

FELM performs better on some problems than L$_2$-RELM. But on some time series problems, such as the shared bicycle time series data set in the UCI machine learning library[32], the number of iterations of FELM is close to 5×10^3, and the time spent is close to 60 s. This is obviously not conducive to practical use.

2.2 FELM combined with adaptive step size determination method

In order to reduce the number of iterations of the FELM algorithm, this section introduces an adaptive step size determination method[33] in FELM (called AFELM). The step size can be determined according to Eq.(7) in the first iteration, from the second iteration, the kth step size is calculated as follows

$$\Delta\boldsymbol{\beta}_k=\boldsymbol{\beta}_k-\boldsymbol{\beta}_{k-1} \tag{13}$$

$$\Delta\boldsymbol{F}_k=\nabla f(\boldsymbol{\beta}_k)-\nabla f(\boldsymbol{\beta}_{k-1}) \tag{14}$$

$$\tau_k^s=\frac{<\Delta\boldsymbol{\beta}_k,\Delta\boldsymbol{\beta}_k>}{<\Delta\boldsymbol{\beta}_k,\Delta\boldsymbol{F}_k>} \tag{15}$$

$$\tau_k^m=\frac{<\Delta\boldsymbol{\beta}_k,\Delta\boldsymbol{F}_k>}{<\Delta\boldsymbol{F}_k,\Delta\boldsymbol{F}_k>} \tag{16}$$

Through the above calculation process, two steps of Eqs.(15, 16) are obtained, and the adaptive step size can be determined by

$$\tau_k=\begin{cases}\tau_k^m & \tau_k^m/\tau_k^s>1/2\\ \tau_k^s-0.5\tau_k^m & \text{Otherwise}\end{cases} \tag{17}$$

AFELM is summarized in Algorithm 3.

Algorithm 3 AFELM

Initialize C, $\boldsymbol{\beta}$, τ, $k=1$, and set the maximum number of iterations K

While $k<K$ or not converged do

$\hat{\boldsymbol{\beta}}_{k+1}=\boldsymbol{\beta}_k-\tau_k\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{H}\boldsymbol{\beta}_k-\boldsymbol{t})$

$\boldsymbol{\beta}_{k+1}=[\operatorname{diag}(\tau_kC)+\boldsymbol{I}]^{-1}\hat{\boldsymbol{\beta}}_{k+1}$

$\Delta\boldsymbol{\beta}_{k+1}=\boldsymbol{\beta}_{k+1}-\boldsymbol{\beta}_k$

$\Delta\boldsymbol{F}_{k+1}=\nabla f(\boldsymbol{\beta}_{k+1})-\nabla f(\boldsymbol{\beta}_k)$

$\tau_{k+1}^s=<\Delta\boldsymbol{\beta}_{k+1},\Delta\boldsymbol{\beta}_{k+1}>/<\Delta\boldsymbol{\beta}_{k+1},\Delta\boldsymbol{F}_{k+1}>$

$\tau_{k+1}^m=<\Delta\boldsymbol{\beta}_{k+1},\Delta\boldsymbol{F}_{k+1}>/<\Delta\boldsymbol{F}_{k+1},\Delta\boldsymbol{F}_{k+1}>$

if $\tau_{k+1}^m/\tau_{k+1}^s>1/2$

$\tau_{k+1}=\tau_{k+1}^m$

else

$\tau_{k+1}=\tau_{k+1}^s-0.5\tau_{k+1}^m$

$k=k+1$

End

Compared with the FBS algorithm, the adaptive FBS algorithm reduces the number of iterations significantly, and the solution accuracy can remain unchanged or superior. The performance comparison between the adaptive FBS algorithm and the FBS algorithm is extremely detailed in Ref.[31].

2.3 Improved AFELM

It is found that the core guaranteeing the convergence of the FBS algorithm is the forward and backward steps, which mainly play animportant role in solving the current optimal solution. However, in FBS or the adaptive FBS, the forward step is just to provide a reference point for the backward step, and the backward step finds a closest advantage solution to the reference point.

This study considers more about the effect of the forward step on the final solution in each iteration, assuming that the solution obtained in the forward step is $\hat{\boldsymbol{\beta}}_{k+1}$ and the solution obtained in the backward step is $\widetilde{\boldsymbol{\beta}}_{k+1}$. Combine a parameter α with the solution of the forward step and the backward step, shown as

$$\boldsymbol{\beta}_{k+1}=\alpha\widetilde{\boldsymbol{\beta}}_{k+1}+(1-\alpha)\widetilde{\boldsymbol{\beta}}_{k+1} \tag{18}$$

AFELM is improved by adding Eq.(18) to Algorithm 3. The improved one is called IAFELM, and is summarized in Algorithm 4.

Algorithm 4 IAFELM

Initialize C, $\boldsymbol{\beta}$, τ, $k=1$, and set the maximum number of iterations K

While $k<K$ or not converged do

…

$\hat{\boldsymbol{\beta}}_{k+1}=\boldsymbol{\beta}_k-\tau_k\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{H}\boldsymbol{\beta}_k-\boldsymbol{t})$

$\widetilde{\boldsymbol{\beta}}_{k+1}=(\tau_kC+\boldsymbol{I})^{-1}\hat{\boldsymbol{\beta}}_{k+1}$

$\boldsymbol{\beta}_{k+1}=\alpha\widetilde{\boldsymbol{\beta}}_{k+1}+(1-\alpha)\hat{\boldsymbol{\beta}}_{k+1}$

$\Delta\boldsymbol{\beta}_{k+1}=\boldsymbol{\beta}_{k+1}-\boldsymbol{\beta}_k$

…

End

The above omitted steps are the same as steps in Algorithm 3.

When α is taken as 1, Algorithm 3 is obtained. And the smaller the value of α is, the faster the convergence speed is. But the accuracy of the experiment is found to decrease, so the value of α is important.

In fact, the improvement made the ELM algorithm similar to using the gradient descent method to replace the least square method. This paper uses an improved FBS algorithm to solve the ELM problem. The FBS algorithm is essentially a type of gradient descent algorithm. The loss function of ELM is a typical convex function, so when the step size is selected appropriately, the gradient descent algorithm can guarantee the convergency. The selection of the step size has been given in Eq.(7), and the selection basis is from Ref.[31], which has a detailed derivation process on convergence, so this paper does not repeat it.

Similarly, the adaptive step size used by AFELM is essentially two mature step sizes: The steepest descent and the minimum residual, which are used more in various gradient algorithms. But this paper introduces them to the ELM solution for the first time in the process, so the convergence of AFELM can also be guaranteed. The question about the convergence of the adaptive step size is also very detailed in Ref.[33].

As for the IAFELM algorithm proposed in this paper, it is not difficult to see that there are not many changes in the selection of gradients, and the focus is on improving the solution of the weights of the actual output layer. AFELM and FELM can be regarded as the two limits of IAFELM, so their convergence problems can be guaranteed as IAFELM is a form between the two learning machines.

2.4 Determination of relevant parameters

The determination of the convergence condition also has a great influence on the performance of the algorithm. The convergence conditions of Algorithms 2—4 determine whether the difference between the solutions obtained by the two iterations is less than a preset threshold. The conditions are judged as follows

$$(\boldsymbol{\beta}_{k+1}-\boldsymbol{\beta}_k)^{\mathrm{T}}(\boldsymbol{\beta}_{k+1}-\boldsymbol{\beta}_k)/M<\varepsilon \tag{19}$$

where M represents the dimension of the solution and ε a preset threshold.

Regarding the value of the parameter α, it is found from many experiments that α takes the step factor of each step to obtain superior performance and can also reduce the iteration time to some extent, shown as

$$\alpha_k=\tau_k \tag{20}$$

2.5 Time series prediction problem

The engine performance parameter prediction problem is actually a time series prediction problem. The time series prediction problem can be summarized as a given time series $S=\{s_1, s_2, \cdots, s_n\}$. If we only consider single-step prediction, it is equivalent to constructe an m-dimensional vector $\boldsymbol{x}_i=\{s_{i+1}, s_{i+2}, \cdots, s_{i+m}\}$ and adopts this vector as the input to predict the next data, where m represents the size of the time window. Multi-step prediction means that the outputting more than one datum is the extrapolation of single-step prediction. It can be seen that the time series prediction problem is a special kind of regression problem, or it can be considered as a function fitting problem.

3 Numerical Experiments on Public Data Sets

In order to verify the performance of the proposed algorithm, this section conducts comparative experiments on several time series data sets. The experimental objects include ELM, L_2-RELM, AFELM and IAFELM. Since the training time of the algorithm FELM is too long, it is not considered for performance on the dataset. The performance comparison between the adaptive FBS algorithm and the FBS algorithm is described in great detail in Ref.[31]. The relevant information of several selected data sets is illustrated in Table 1.

Table 1 Data set settings

Data set	Size	Training
SINC(A)	1 000	600
GSK(B)	1 060	636
JNJ(C)	1 067	640
Dow Jones Industrial Average (D)	1 225	735
US Standard & Poor's 500 Index (E)	1 112	667
Mackey-glass(F)	1 005	603

The data set A is a set of data generated by the SINC function, data sets B—E are classic stock time series data sets selected from Yahoo Finance[34], and F is a commonly used data set in the field of time series prediction. All experiments were performed on computer configuration information as follows: Intel (R) Core (TM) i5-4210U CPU@

1.70 GHz/2.40 GHz; RAM 4.00 GB; Windows 10 64-bit, and the platform adopt for the experiment is MATLAB R2014a.

The number of neurons in the input layer of the neural network is set to the number of time windows m ($m=5$), and the number of neurons in the output layer is set to the number of prediction steps. If single-step prediction is performed, it is set to 1, and set to the number of steps when performing multi-step prediction. The evaluation indicators are root mean squares error (RMSE) and mean absolute error (MAE), shown as

$$\mathrm{RMSE}=\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\tilde{t}_i-t_i)^2} \tag{21}$$

$$\mathrm{MAE}=\frac{1}{n}\sum_{i=1}^{n}|\tilde{t}_i-t_i| \tag{22}$$

where n is the number of test samples, t the actual output value, and $\tilde{t}$ the model prediction value.

The stop threshold ε is taken as 1×10^{-10}. Equal considering structural risks and empirical risks, the regularization parameter C is taken as 1, the activation function takes the Sigmoid function, the input weight and the threshold are randomly determined, and all the data are normalized to $[-1, 1]$ in experiments. The number of hidden layer neurons is taken as 50 and the results are shown in Tables 2, 3.

Table 2 RMSE and MAE for all experiments

Data set	RMSE				MAE			
	ELM	L_2-RELM	AFELM	IAFELM	ELM	L_2-RELM	AFELM	IAFELM
A	0.097 7	0.139 1	**0.045 7**	0.049 6	0.070 7	0.120 1	**0.037 3**	0.040 0
B	0.142 3	0.150 1	**0.134 4**	0.142 2	0.102 1	0.114 1	**0.102 4**	0.108 7
C	0.154 3	0.159 4	**0.142 3**	0.147 7	0.095 1	0.114 8	**0.099 4**	0.104 4
D	0.110 8	0.127 1	**0.102 5**	0.104 3	0.073 3	0.094 1	**0.076 0**	0.077 3
E	0.107 9	0.116 2	0.094 3	**0.091 1**	0.066 9	0.086 2	0.070 2	**0.067 9**
F	0.213 6	0.171 7	**0.131 2**	0.167 3	0.079 2	0.127 9	**0.094 5**	0.112 0

Note: Bold indicates the best performance on the current data set.

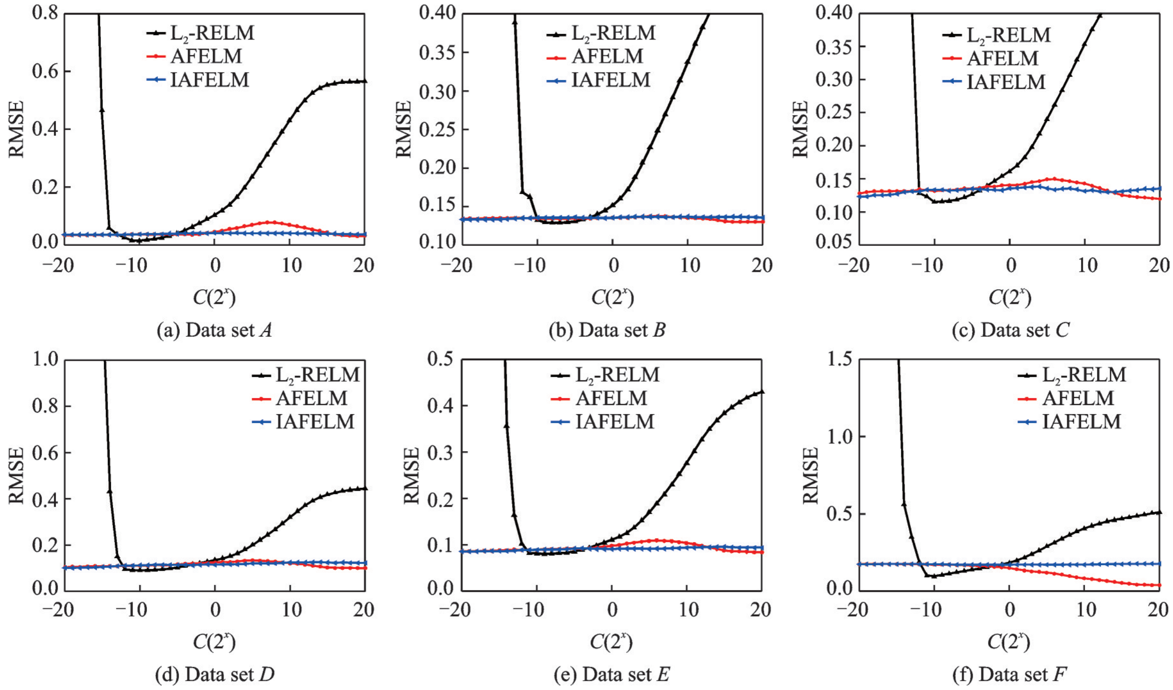
Table 3 Training time and number of iterations for all experiments

Data set	Training time/s				Number of iterations	
	ELM	L_2-RELM	AFELM	IAFELM	AFELM	IAFELM
A	0.284	0.276	0.494	0.305	174	33
B	0.243	0.287	0.392	0.290	60	26
C	0.249	0.227	0.299	0.291	69	30
D	0.260	0.257	0.333	0.281	65	36
E	0.230	0.226	0.330	0.315	72	38
F	0.253	0.233	0.301	0.278	49	30

From Table 2, it can be concluded that the RMSEs obtained by AFELM and IAFELM are always better than those by ELM and L_2-RELM on all datasets. And from Table 3, it can be gotten that although AFELM has obtained more predictive effects, the number of iterations and training time increase. While IAFELM is not as positive at predicting performance as AFELM, but it performs better in the number of iterations and training time, and the accuracy is reduced in an acceptable range.

First, the influence of the regularization parameter C on the prediction effect is studied, here the range of C is $[2^{-20}, 2^{20}]$, and the varying diagrams of RMSE with the range of C (2^x) are illustrated in Fig.1. From Fig.1 we can see that, with the increase of the regularization parameter C, the performance of L_2-RELM is firstly improved and then deteriorated, while AFELM and IAFELM can always maintain excellent performance and not sensitive to parameter C.

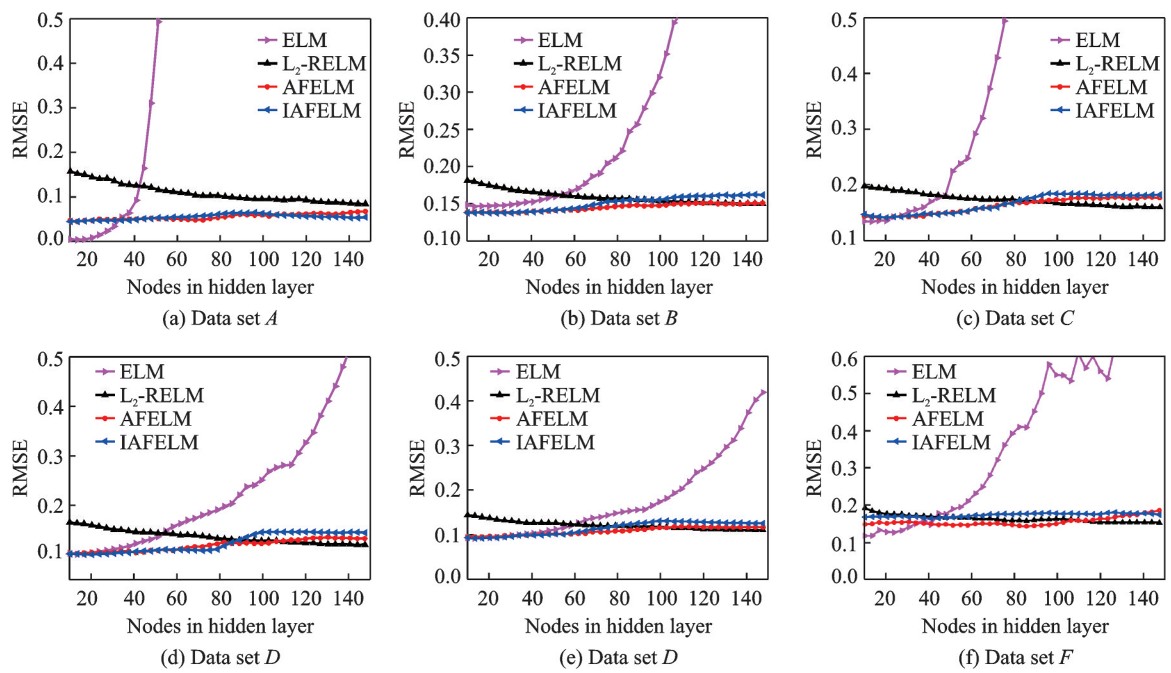
Next, the influence of the number of neurons in the hidden layer on the prediction effect is studied. The number of nodes in the hidden layer is increased from 10 to 150, and experiments on data sets A—F are performed. The varying diagrams of RMSE with the number of layer nodes are illustrated in Fig.2. It can be seen from Fig.2 that as the number of hidden layer nodes increases, the RMSEs of ELM and L_2-RELM increase and generally decrease respectively, while the RMSEs of AFELM and IAFELM have a certain increasing trend. On some data sets, when the number of hidden layer nodes is low, ELM reports the optimal prediction effect, but the stability is



(a) Data set *A* (b) Data set *B* (c) Data set *C*

(d) Data set *D* (e) Data set *E* (f) Data set *F*

Fig.1 Influence of regularization parameter *C* on RMSE



(a) Data set *A* (b) Data set *B* (c) Data set *C*

(d) Data set *D* (e) Data set *D* (f) Data set *F*

Fig.2 Influence of the number of layer nodes on RMSE

extremely poor. When the number of neurons is increasing, the prediction effect of L_2-RELM closes to that of AFELM and IAFELM. On some datasets, when the number of hidden layer nodes is large, the prediction effect of L_2-RELM is optimal, but its performance is still not as positive as that of AFELM with fewer nodes. The prediction effect of AFELM and IAFELM is always better than that of L_2-RELM when the number of neurons is small.

The above experiments consider single-step prediction. Next experiment studies the multi-step prediction, and prediction results with different numbers of steps are illustrated in Table 4.

From Table 4, we can see that with the increase of the number of prediction steps, the prediction effect is getting worse, but the prediction effects of AFELM and IAFELM are always not weaker than those of ELM and L_2-RELM.

Table 4 RMSE under multi-step prediction

Data set	Prediction step	RMSE			
		ELM	L_2-RELM	AFELM	IAFELM
A	2	0.151 4	0.133 8	**0.052 7**	0.054 6
	3	0.119 6	0.099 7	**0.055 2**	0.059 6
	4	0.138 2	0.150 4	**0.060 3**	0.062 3
	5	0.313 6	0.124 0	0.073 4	**0.065 5**
B	2	0.169 5	0.163 7	0.158 9	**0.156 0**
	3	0.195 0	0.183 2	**0.171 7**	0.175 4
	4	0.220 0	0.195 9	**0.187 4**	0.188 7
	5	0.240 9	0.205 5	0.199 8	**0.198 8**
C	2	0.175 5	0.177 3	**0.150 0**	0.157 5
	3	0.190 1	0.196 0	0.173 2	**0.168 7**
	4	0.238 8	0.212 1	0.193 7	**0.191 4**
	5	0.233 2	0.230 5	**0.207 9**	0.210 7
D	2	0.124 5	0.141 2	0.116 4	**0.120 4**
	3	0.176 8	0.145 7	**0.129 0**	0.130 4
	4	0.149 6	0.164 4	**0.144 2**	0.144 3
	5	0.157 6	0.172 7	**0.152 0**	0.155 3
E	2	0.122 2	0.122 6	0.109 6	**0.106 2**
	3	0.137 6	0.136 8	0.126 1	**0.120 7**
	4	0.150 2	0.134 6	**0.125 1**	0.128 0
	5	0.164 6	0.158 9	**0.149 1**	0.150 9
F	2	0.259 1	0.160 8	**0.130 4**	0.162 0
	3	0.229	0.203 6	**0.181 5**	0.195 6
	4	0.347 4	0.198 6	**0.159 8**	0.190 1
	5	0.409 3	0.212 0	**0.192 6**	0.204 5

Finally, the influence of the number of training samples on the prediction effect is studied. The training samples are increased from 50% to 80% of the total, and the varying diagrams of RMSE with the percentage of training samples are illustrated in Fig.3.

From Fig.3, it can be concluded that AFELM and IAFELM indicate superior predictive performance and outstanding stability when other conditions are the same, and ELM and L_2-RELM sometimes illustrate better or comparable performance,

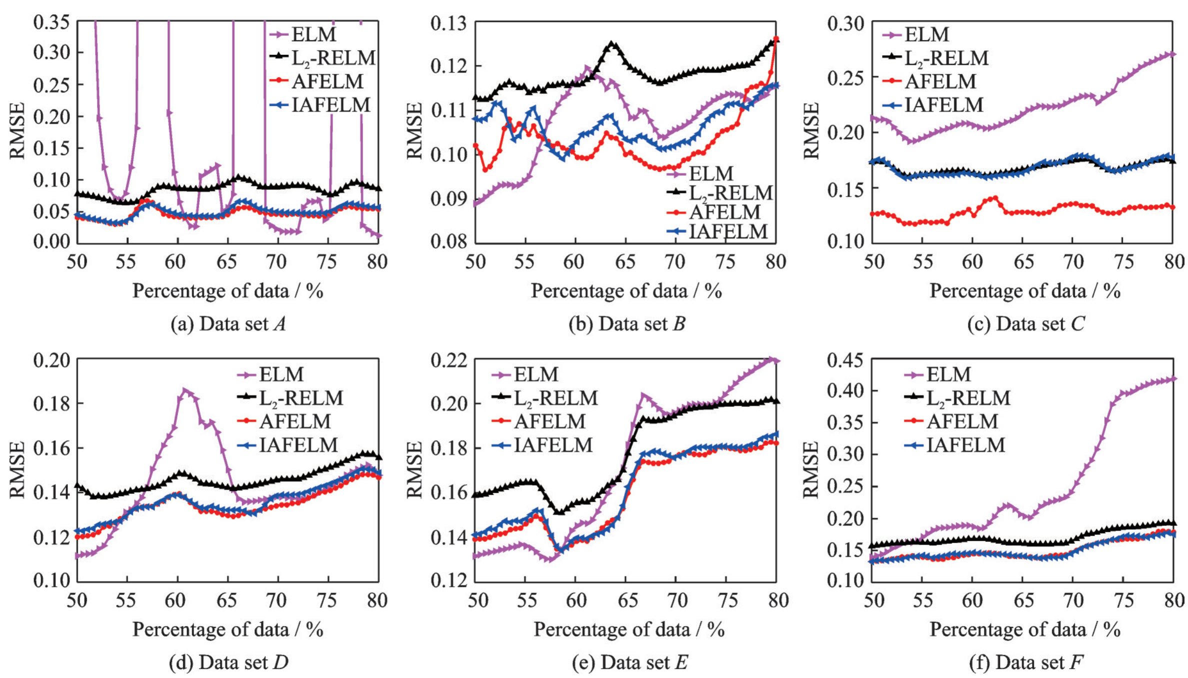


Fig.3 Influence of percentage of training samples on RMSE

but not stable enough for practical applications.

4 Aeroengine Performance Parameter Prediction

4.1 Prediction on simulation data

Aeroengines have a harsh operating environment and complex components, so performance degradation is inevitable after an increase in their service life. This degradation is irreversible and will become a fault to a certain extent, seriously jeopardizing the operational safety of the engine. And there will often be some signs on the relevant performance parameters of the engine in the early stage of performance degradation. If engineers can get this information in advance, it is intensely effective to avoid engine failure. Aeroengines typically consist of an air inlet, a fan, a low pressure compressor (LPC), a high pressure compressor (HPC), a combustor, a high pressure turbine (HPT), a low pressure turbine (LPT), bypasses, and a nozzle. The simplified diagram of the engine is shown in Fig.4, and the performance parameters are collected by sensors attached to these components. NASS' Commercial Modular Aeronautical Propulsion System Simulation (C-MAPSS)[35] generates a set of turbofan engine performance degradation data sets, each of which is a set of multivariate time series, including engine unit ID, operation cycle index, three values indicating the operational settings and 21 sensor measurements contaminated by unknown noises.

The data-driven aeroengine performance parameter prediction method is mainly divided into the following steps: (1) Preprocess the historical data of the obtained performance parameters, including normalization and smoothing, (2) use the pre-processed data to train the model and obtain a mature model, (3) input the pre-predicted parameter data into the model obtained by the last step, and the model outputs the predicted value of the performance parameter.

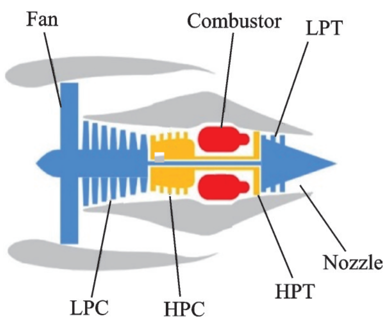


Fig.4 Simplified diagram of aircraft engine

This section uses the algorithm contributed above to predict five important performance parameters of aeroengines. This study compares the prediction results with L_2-RELM to illustrate the effect of the improved algorithm. The important parameters in the C-MAPSS data set selected for prediction are listed in Table 5[36].

Table 5 Specification of the selected measurement sensor signals based on C-MAPSS

Parameter	Description
T_{24}/°R	Total temperature at lower-pressure compressor inlet
N_f/(r•min^{-1})	Physical fan speed
P_{s30}/PSIA	Static pressure at high-pressure compressor outlet
N_{Rf}/(r•min^{-1})	Corrected fan speed
BPR	Bypass rate

FD001 is the data under a class of working conditions in the C-MAPSS data set, including the "train_FD001" data set. This data set is the time series data of performance parameters of the engine from normal to degraded under a single operating condition, and is often used for the verification of aeroengine performance prediction algorithms. Performance parameter prediction is performed using "train_FD001.txt", in which engines #1— #10 are selected as test sets, and engines #11—#100 are selected as training sets. Before the training, the data are denoised. The twenty-point moving average method[36] is used to smooth the time series raw data.

Based on the previous analysis, we can draw a conclusion that AFELM and IAFELM can have fewer hidden layers with better prediction effect, which is meaningful for saving space in practical applications. Therefore, when predicting engine performance parameters, the number of hidden layer neurons is reduced to 20, the time window m is set to 5, and the stop threshold ε is set to 1×10^{-10}. Because of the experimental results on the insensitivity of the new algorithm to regularization parameter C, we set

C to be 1, take Sigmoid as the activation function, randomly determine the input weight and the threshold, and normalize all the data to $[-1, 1]$ in experiments. Results are illustrated in Table 6.

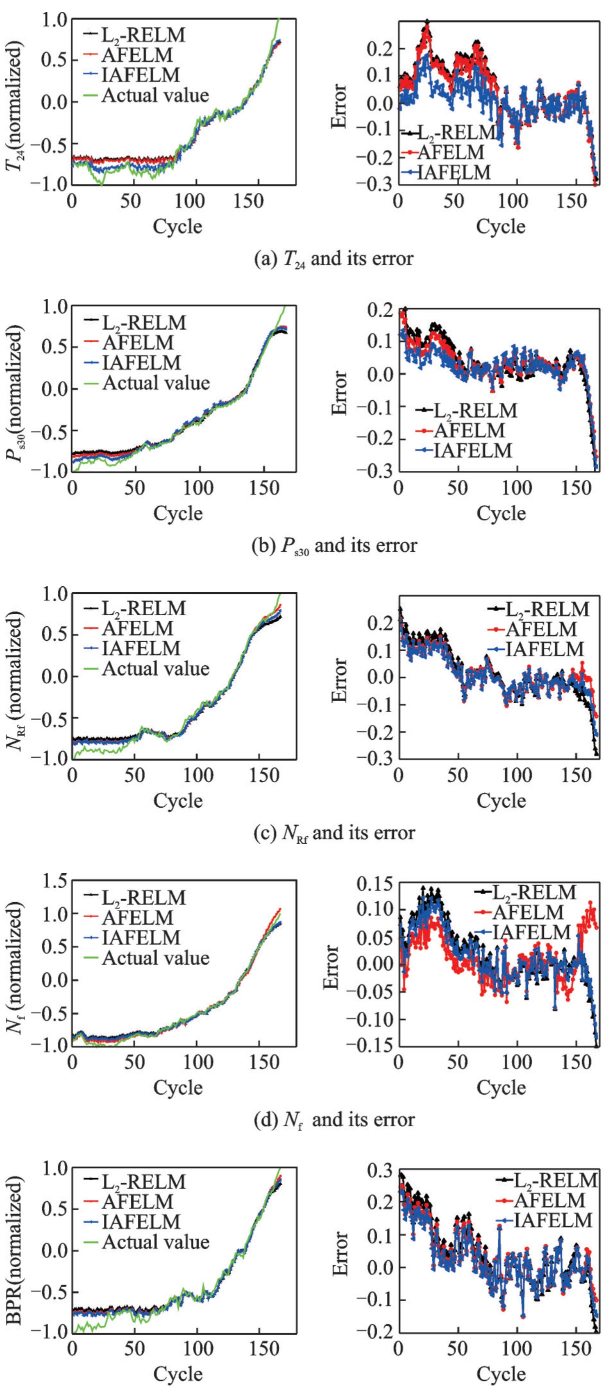
Table 6 RMSE and MAE of parameter prediction of engines #1—#10

Engine number	Parameter	RMSE			MAE		
		L_2-RELM	AFELM	IAFELM	L_2-RELM	AFELM	AFELM
#1	T_{24}	0.127 5	0.120 6	**0.077 6**	0.095 5	0.090 1	**0.054 1**
	N_f	0.061 0	**0.045 7**	0.052 7	0.042 1	**0.033 5**	0.036 1
	P_{s30}	0.092 8	0.073 4	**0.063 7**	0.061 1	0.050 4	**0.042 6**
	N_{Rf}	0.101 2	**0.076 2**	**0.076 2**	0.071 7	**0.054 2**	0.054 8
	BPR	0.113 6	0.097 0	**0.090 1**	0.079 6	0.067 3	**0.063 4**
#2	T_{24}	0.091 5	**0.077 9**	0.084 7	0.065 8	**0.054 2**	0.061 1
	N_f	0.074 5	**0.052 5**	0.045 7	0.048 6	0.035 0	**0.031 4**
	P_{s30}	0.084 7	0.069 5	**0.072 8**	0.060 7	**0.050 8**	0.052 4
	N_{Rf}	0.086 4	0.120 3	**0.077 9**	0.064 1	0.079 8	**0.058 5**
	BPR	0.116 9	0.091 5	**0.074 5**	0.083 2	0.063 9	**0.050 2**
#3	T_{24}	0.089 6	0.078 9	**0.074 9**	0.060 2	0.054 2	**0.052 7**
	N_f	0.111 0	**0.085 6**	0.087 0	0.084 9	**0.065 7**	0.065 8
	P_{s30}	0.112 4	0.089 6	**0.065 6**	0.076 1	0.060 0	**0.046 6**
	N_{Rf}	0.095 0	**0.076 3**	0.080 3	0.063 5	**0.052 5**	0.060 1
	BPR	0.069 6	0.069 6	**0.046 8**	0.050 6	0.049 6	**0.032 8**
#4	T_{24}	0.204 8	**0.158 1**	0.182 8	0.152 1	**0.115 1**	0.131 2
	N_f	**0.097 6**	**0.097 6**	0.099 0	**0.071 3**	0.071 8	0.072 0
	P_{s30}	0.079 7	0.079 7	**0.078 4**	**0.060 6**	0.062 1	0.060 8
	N_{Rf}	0.115 5	0.158 1	**0.103 1**	0.076 3	0.113 5	**0.068 3**
	BPR	0.071 5	0.072 9	**0.055 0**	0.050 5	0.050 4	**0.039 7**
#5	T_{24}	0.075 4	0.065 6	**0.054 1**	0.054 9	0.044 6	**0.036 3**
	N_f	0.055 8	0.049 2	**0.045 9**	0.040 7	0.036 8	**0.035 3**
	P_{s30}	0.080 4	0.073 8	**0.042 6**	0.058 1	0.052 8	**0.031 9**
	N_{Rf}	0.121 4	0.093 5	**0.067 2**	0.090 2	0.071 5	**0.048 6**
	BPR	0.049 2	0.049 2	**0.039 4**	0.035 8	0.035 4	**0.029 0**
#6	T_{24}	0.100 1	0.098 7	**0.090 5**	0.072 3	**0.062 3**	0.065 2
	N_f	0.093 2	0.074 0	**0.054 8**	0.079 2	0.062 8	**0.048 2**
	P_{s30}	0.090 5	0.078 2	**0.064 4**	0.079 8	0.071 4	**0.056 3**
	N_{Rf}	0.124 8	0.097 4	**0.067 2**	0.082 5	0.065 5	**0.049 0**
	BPR	0.109 7	0.105 6	**0.086 4**	0.076 8	0.076 1	**0.064 6**
#7	T_{24}	0.141 6	0.130 4	**0.114 3**	0.096 8	0.089 4	**0.078 8**
	N_f	0.046 7	**0.043 5**	0.045 1	0.032 8	**0.031 4**	0.032 2
	P_{s30}	0.067 6	**0.061 2**	0.064 4	0.048 8	**0.044 5**	0.046 2
	N_{Rf}	0.075 6	0.069 2	**0.064 4**	0.049 2	0.045 4	**0.040 8**
	BPR	0.077 2	**0.074 0**	0.075 6	0.059 9	**0.057 2**	0.058 8
#8	T_{24}	0.171 5	**0.165 3**	0.173 9	0.128 2	**0.120 0**	0.129 7
	N_f	0.075 9	**0.057 6**	0.066 1	0.050 1	**0.041 7**	0.046 0
	P_{s30}	0.073 5	0.064 9	**0.063 7**	0.050 7	0.047 0	**0.045 6**
	N_{Rf}	0.083 3	0.087 0	**0.073 5**	0.054 3	0.061 1	**0.049 4**
	BPR	0.088 2	**0.067 4**	0.083 3	0.060 4	**0.048 2**	0.057 4
#9	T_{24}	0.079 4	0.073 7	**0.072 3**	0.057 5	0.055 5	**0.054 3**
	N_f	0.092 2	**0.087 9**	0.095 0	0.063 2	**0.063 3**	0.068 4
	P_{s30}	0.075 1	0.048 2	**0.035 4**	0.052 8	0.035 2	**0.027 0**
	N_{Rf}	0.120 5	0.112 0	**0.096 4**	0.096 0	0.086 0	**0.073 3**
	BPR	0.065 2	0.066 6	**0.051 0**	0.046 1	0.046 0	**0.036 2**
#10	T_{24}	0.079	**0.070 0**	0.086 4	0.060 2	**0.051 7**	0.065 7
	N_f	0.105 8	**0.087 9**	0.092 4	0.080 4	**0.068 0**	0.071 0
	P_{s30}	0.071 5	0.059 6	**0.058 1**	0.049 8	0.042 4	**0.041 7**
	N_{Rf}	0.055 1	0.056 6	**0.049 2**	0.036 1	0.036 9	**0.033 0**
	BPR	0.095 4	0.087 9	**0.055 1**	0.073 9	0.067 4	**0.042 9**

Note: Bold values indicate the best performance on the current parameter.

It can be concluded from Table 6 that the two new algorithms perform better than L_2-RELM on engines #1—#10.

This paper also provides a prediction of five parameters of engine #1, shown in Fig.5. It is not difficult to find that in the first 50 cycles of performance parameters, the fluctuations are relatively large, and the irregular fluctuation is a difficult problem in the time series forecasting problem, so the algorithm will perform poorly in the first 50 cycles.



(a) T_{24} and its error

(b) P_{s30} and its error

(c) N_{Rf} and its error

(d) N_f and its error

(e) BPR and its error

Fig.5 Varying diagrams of different parameters and errors of engine #1 with cycle

Next, a simple experiment on the training and test time is carried out. The engine #8 with the smallest amount of data is selected in the test sample and the engine #2 with the largest data volume is selected to conduct the experiment. The training sample selection is unchanged, still using engines #11—#100, then test experiments are carried out on engines #8 and #2, respectively. The results on training time and test time are obtained, shown in Tables 7—9.

Table 7 Training time s

Parameter	L_2-RELM	AFELM	IAFELM
T_{24}	2.087	3.104(78)	2.209(25)
N_f	2.035	2.877 (69)	2.589(29)
P_{s30}	2.065	3.897 (150)	2.778 (35)
N_{Rf}	2.005	3.822 (132)	2.768 (33)
BPR	2.314	3.661 (123)	2.434 (35)

Note: Values in parentheses mean the number of iterations.

Table 8 Test time of engine #8 s

Parameter	L_2-RELM	AFELM	IAFELM
T_{24}	0.370	0.337	0.306
N_f	0.361	0.302	0.325
P_{s30}	0.342	0.323	0.314
N_{Rf}	0.338	0.303	0.324
BPR	0.356	0.306	0.298

Table 9 Test time of engine #2 s

Parameter	L_2-RELM	AFELM	IAFELM
T_{24}	0.375	0.348	0.273
N_f	0.369	0.339	0.338
P_{s30}	0.358	0.320	0.325
N_{Rf}	0.364	0.355	0.306
BPR	0.350	0.342	0.328

Through Table 7 we can see that: Although AFELM can obtain the higher prediction accuracy, it often consumes more computing resources during training; IAFELM has a certain reduction in training time compared with AFELM, and the accuracy is reduced within an acceptable range. From Tables 8, 9, we find out that the test time of the three algorithms is not much different, which means the three algorithms all have a faster calculation speed. It is intensely instructive to predict engine performance parameters with certain requirements on prediction accuracy and training time.

4.2 Prediction on real data

The above experiments are all performed on the simulation data. Although the algorithm performs well, the degradation of the predicted performance parameters is a exceedingly ideal state. Therefore, the performance parameters are predicted on the real data. The operating conditions in the real situation are overwhelmingly complicated, and the varieties in engine performance parameters are more complicated. Two typical performance parameters from the same model of Honeywell collected from an airline are selected, which are turbo turbine total temperature (EGT) and low pressure rotor speed N_1, and their variation with cycle is shown in Fig.6.

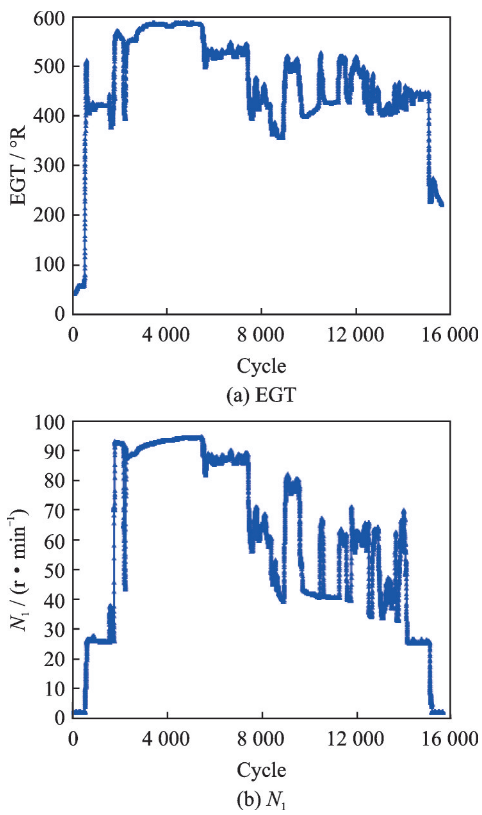


Fig.6 Performance parameter variation with cycle

A total of 15 648 flight cycle parameters are recorded, but it is not difficult to find that the parameters of the first 5 000 flight cycles are relatively stable. The parameters of the 10 648 flight cycles fluctuate greatly. This paper selects the performance parameters after 5 000 flight cycles to predict. The data of the first 7 986 flight cycles are selected as the training samples, and the parameters of the 2 662 flight cycles are predicted. After normalizing and smoothing, the experimental results are shown in Tables 10, 11 and Figs.7, 8.

Table 10 RMSE of two parameters by three algorithms

Parameter	L_2-RELM	AFELM	IAFELM
EGT/°R	0.056 3	0.031 6	**0.028 4**
N_1/(r•min^{-1})	0.054 5	**0.031 8**	0.035 3

Table 11 MAE of two parameters by three algorithms

Parameter	L_2-RELM	AFELM	IAFELM
EGT/°R	0.042 9	0.023 3	**0.021 5**
N_1/(r•min^{-1})	0.038 0	**0.020 6**	0.020 8

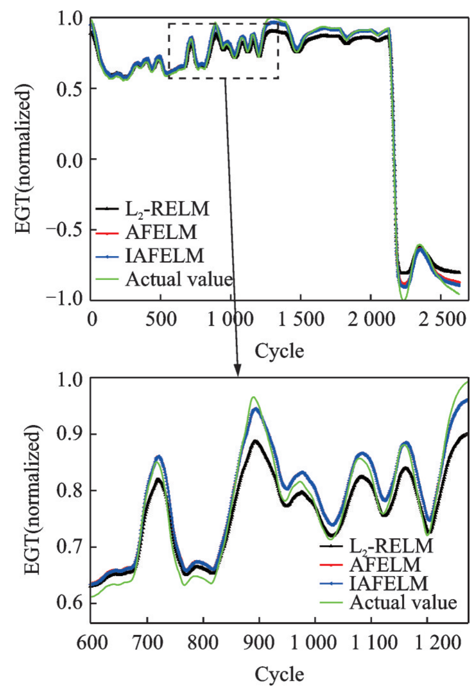


Fig.7 EGT prediction results

It can be seen from Fig.7 that within the prediction range of the test set of the entire EGT parameter, the prediction performance of the two new algorithms is better than that of L_2-RELM at all peaks. At some troughs, the prediction performance of L_2-RELM is better, but the two new algorithms can always perform similar to L_2-RELM, which means AFELM and IAFELM even perform better in troughs.

It can be seen from Fig.8 that within the prediction range of the N_1 parameter test set, the two new algorithms are better than L_2-RELM in all peaks and troughs. It is proved that AFELM and IAFELM always obtain the higher prediction accuracy.

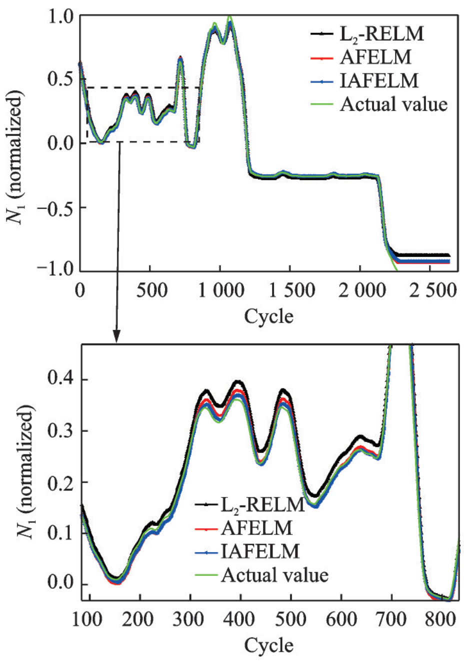


Fig.8 N_1 prediction results

5 Conclusions

In order to solve the poor performance of L_2-RELM in time series prediction under the condition that the regular parameters are not easy to be determined, this paper uses the FBS algorithm to solve L_2-RELM. But it is found that the new algorithm costs more training time and needs multiple iterations through experiments. Therefore, this paper introduces an adaptive algorithmic iterative step size determination method to obtain the AFELM algorithm. On the basis of AFELM, an improved method for the solution mechanism of the forward and backward steps is proposed to further reduce the number of iterations. During the training time, experiments are carried out on several commonly used time series prediction data sets. Results show that the newly proposed algorithm is not sensitive to regularization parameters, and can obtain better prediction performance than L_2-RELM at a faster training speed. The proposed algorithm is applied to the engine public data set for performance parameter prediction, and achieves better prediction performance than L_2-RELM. In order to further compare the performance of the algorithm, the actual engine performance parameters of an airline are predicted, and results prove that the proposed algorithm still obtains good prediction performance.

References

[1] LIU Xiaofeng, YUAN Ye, SHI Jing, et al. Adaptive modeling of aircraft engine performance degradation model based on the equilibrium manifold and expansion form[J]. Journal of Aerospace Engineering, 2014, 228(8): 1246-1272.

[2] ZHAO Ningbo, LI Shuying, YANG Jialong. A review on nanofluids: Data driven modeling of thermal physical properties and the application in automotive radiator[J]. Renewable and Sustainable Energy Reviews, 2016, 66: 596-616.

[3] ZHOU Dengji, YU Ziqiang, ZHANG Huisheng, et al. A novel grey prognostic model based on Markov process and grey incidence analysis for energy conversion equipment degradation[J]. Energy, 2016, 109: 420-429.

[4] ZHONG Shisheng, XIE Xiaolong, LIN Lin, et al. Genetic algorithm optimized double-reservoir echo state network for multi-regime time series prediction[J]. Neurocomputing, 2017, 238: 191-204.

[5] YUAN M, WU Y, LIN L. Fault diagnosis and remaining useful life estimation of aero engine using LSTM neural network[C]//Proceedings of IEEE International Conference on Aircraft Utility Systems. USA: IEEE, 2016: 135-140.

[6] JANSSENS O, SLAVKOVIKJ V, VERVISCH B, et al. Convolutional neural network based fault detection for rotating machinery[J]. Journal of Sound and Vibration, 2016, 377: 331-345.

[7] BING Y, QU W. Aero-engine sensor fault diagnosis based on stacked denoising autoencoders[C]//Proceedings of Control Conference. USA: IEEE, 2016: 6542-6546.

[8] HUANG G B, ZHU Q, SIEW C. Extreme learning machine: Theory and applications[J]. Neurocomputing, 2006, 70(1): 489-501.

[9] YE Yibin, YANG Qin. QR factorization based incremental extreme learning machine with growth of hidden nodes[J]. Pattern Recognition Letters, 2015, 65:

177-183.

[10] ZHAO Y P, LI Z Q, XIA P P. Gram-Schmidt process based incremental extreme learning machine[J]. Neurocomputing, 2017, 241: 1-17.

[11] CAO J W, ZHANG K, YONG H W. Extreme learning machine with affine transformation inputs in an activation function[J]. IEEE Transactions on Neural Networks and Learning Systems, 2019(30): 2093-2107.

[12] ZHAO Y P, XI P P, LI B. Sparse kernel minimum squared error using Householder transformation and givens rotation[J]. Applied Intelligence, 2018(48): 390-415.

[13] NOBREGA J P, OLIVEIRA A L. A sequential learning method with Kalman filter and extreme learning machine for regression and time series forecasting[J]. Neurocomputing, 2019, 337: 235-250.

[14] ANDRE G C, PACHECO A, RENATO A, et al. Restricted Boltzmann machine to determine the input weights for extreme learning machines[J]. Expert Systems with Applications, 2018(9): 77-85.

[15] ANWESHA L, ASHISH G. Multi-label classification using a cascade of stacked autoencoder and extreme learning machines[J]. Neurocomputing, 2019, 358: 222-234.

[16] LU F, JIANG J P, HUANG J Q. Gas turbine engine gas-path fault diagnosis based on improved SBELM architecture[J]. International Journal of Turbo & Jet-Engines, 2018, 35(4): 351-363.

[17] FENG L, JIANG J P, HUANG J Q. Dual reduced kernel extreme learning machine for aero-engine fault diagnosis[J]. Aerospace Science and Technology, 2017, 71: 742-750.

[18] LU J J, HUANG J Q, LU F. Distributed kernel extreme learning machines for aircraft engine failure diagnostics[J]. Applied Sciences-Basel, 2019, 9(8): 1707.

[19] ZHAO Y P, HUANG G, HUA Q K. Soft extreme learning machine for fault detection of aircraft engine[J]. Aerospace Science and Technology, 2019, 91: 70-81.

[20] LU F, WU J D, HUANG J Q. Aircraft engine degradation prognostics based on logistic regression and novel OS-ELM algorithm[J]. Aerospace Science and Technology, 2019, 84: 661-671.

[21] MA Y Y, SHEN D X, WU L F. The remaining useful life estimation of lithiumion Batteries based on the HKA-ML-ELM algorithm[J]. International Journal of Electrochemical Science, 2019,14(8): 7737-7757.

[22] YANG J, ZHEN P, WANG H G. The remaining useful life estimation of lithiumion battery based on improved extreme learning machine algorithm[J]. International Journal of Electrochemical Science, 2018, 13(5): 4991-5004.

[23] DENG W, ZHENG Q, CHEN L. Regularized extreme learning machine[C]//Proceedings of IEEE Symposium on Computational Intelligence and Data Mining. Nashville: IEEE, 2009: 389-395.

[24] LUO X, CHANG X H, BAN X J. Regression and classification using extreme learning machine based on L1-norm and L2-norm[J]. Neurocomputing, 2016, 174: 179-186.

[25] LI S, SONG S J, WAN Y H. Laplacian twin extreme learning machine for semi-supervised classification[J]. Neurocomputing, 2018, 321: 17-27.

[26] YI Y G, QIAO S J, ZHOU W. Adaptive multiple graph regularized semi-supervised extreme learning machine[J]. Soft Computing, 2018(22): 3545-3562.

[27] INABA F K, TEATINI-SALLES E O, PERRON S. DGR-ELM-distributed generalized regularized ELM for classification[J]. Neurocomputing, 2018, 275: 1522-1530.

[28] MA J, WEN Y K, YANG L M. Lagrangian supervised and semi-supervised extreme learning machine[J]. Applied Intelligence, 2019, 49: 303-318.

[29] MAHMOODA S F, MARHABANB M H, ROKHANIA F Z. FASTA-ELM: A fast adaptive shrinkage/thresholding algorithm for extreme learning machine and its application to gender recognition[J]. Neurocomputing, 2017, 219: 312-322.

[30] SONG T H, LI D Z, LIU Z Y. Online ADMM-based extreme learning machine for sparse supervised learning[J]. IEEE Access, 2019(7): 64533-64544.

[31] GOLDSTEIN T O M, STUDER C, BARANIUK R. A field guide to forward-backward splitting with a FASTA implementation[EB/OL]. (2014-11-17)[2020-05-10]. http://arxiv.org/abs/1411.3406.

[32] FRANK A, ASUNCION A. UCI machine learning repository[EB/OL]. (2013-11-06) [2020-05-10]. http://archive.ics.uci.edu/ml.

[33] ZHOU B, GAO L, DAI Y H. Gradient methods with adaptive step-sizes[J]. Computational Optimization and Applications, 2006, 35(1): 69-86.

[34] Yahoo finance[EB/OL]. (2014-05) [2020-05-10]. http://finance.yahoo.com/.

[35] OZA N. Turbofan engine degradation simulation data set[EB/OL]. (2010-09-30)[2020-05-10]. https://c3.nasa.gov/dashlink/resources/139/.

[36] JAVED K, GOURIVEAU R, ZERHOUNI N. A

new multivariate approach for prognostics based on extreme learning machine and fuzzy clustering[J]. IEEE Trans Cybern, 2015, 45(12): 2626.

[37] LU F, JU H F, HUANG J Q. An improved extended Kalman filter with inequality constraints for gas turbine engine health monitoring[J]. Aerospace Science Technology, 2016, 58: 36-47.

Author Mr. **CAO Yuyuan** received the B.S. degree from the College of Civil Aviation, Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China. In 2009, he obtained the M.S. degree in aviation safety management jointly issued by the French National University of Civil Aviation and the French University of Aviation. He is currently a researcher in the College of Civil Aviation, NUAA. His current research interests include aeroengine fault diagnosis and machine learning.

Author contributions Mr. **CAO Yuyuan** proposed the idea and designed the experiment. Mr. **ZHANG Bowen** wrote the manuscript. Dr. **WANG Huawei** reviewed previous research. All authors commented on the manuscript draft and approved the submission.

Competing interests The authors declare no competing interests.

(Production Editor: ZHANG Huangqun)

基于改进正则化极限学习机的航空发动机性能参数预测

曹愈远，张博文，王华伟

（南京航空航天大学民航学院，南京 211106，中国）

摘要：性能参数预测技术是航空发动机健康管理的核心研究内容，越来越多的机器学习算法被应用于该领域，正则化极限学习机(Regularized extreme learning machine, RELM)是其中之一。但RELM的正则化参数确定非常消耗计算资源，使其在数据量较大的航空发动机性能参数预测领域表现出不适应性。本文使用前向和后向分割(Forward and backward segmentation, FBS)算法提升RELM性能，并引入自适应步长确定方法和一种改进的求解机制获得新的机器学习算法。新算法在保持良好泛化的同时，对正则化参数不敏感，极大地节省了计算资源。在公开数据集上的实验结果证明了新算法的优异性能。最后将新算法应用于航空发动机性能参数的预测，取得了优异的预测性能。

关键词：极限学习机；航空发动机；性能参数预测；前向和后向分割算法