

## Research Article

# Multisensor Prognostic of RUL Based on EMD-ESN

Jixin Pei  and Jian Wang

Computer Integrated Manufacturing System Research Center, College of Electronics and Information Engineering, Tongji University, Shanghai 201804, China

Correspondence should be addressed to Jixin Pei; 1832973@tongji.edu.cn

Received 14 October 2020; Revised 11 November 2020; Accepted 16 November 2020; Published 24 November 2020

Academic Editor: Mohammad Mahdi Abaei

Copyright © 2020 Jixin Pei and Jian Wang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents a prognostic method for RUL (remaining useful life) prediction based on EMD (empirical mode decomposition)-ESN (echo state network). The combination method adopts EMD to decompose the multisensor time series into a bunch of IMFs (intrinsic mode functions), which are then predicted by ESNs, and the outputs of each ESN are summarized to obtain the final prediction value. The EMD can decompose the original data into simpler portions and during the decomposition process, much noise is filtered out and the subsequent prediction is much easier. The ESN is a relatively new type of RNN (recurrent neural network), which substitutes the hidden layers with a reservoir remaining unchanged during the training phase. The characteristic makes the training time of ESN is much shorter than traditional RNN. The proposed method is applied to the turbofan engine datasets and is compared with LSTM (Long Short-Term Memory) and ESN. Extensive experimental results show that the prediction performance and efficiency are much improved by the proposed method.

## 1. Introduction

Prognostics is an engineering discipline with regard to predictive diagnostics, including calculation of the remaining useful life based on the observed system condition [1]. As machines are getting increasingly complicated, prognostics for complex systems has attracted more significant interest [2]. RUL prediction enables identifying problems at an early stage, which makes it beneficial for industries to make effective maintenance planning and reduce maintenance cost [1, 3]. Research techniques of RUL have been adopted in various industrial objects, including lithium-ion battery [4] and turbofan engine [5].

In general, the RUL prediction methods are typically classified as physics-model based approaches, data-driven approaches, and hybrid approaches, and data-driven methods can be divided into statistical model based approaches and AI approaches further [6, 7]. Among them, approaches based on the physical model [8, 9] generate an explicit mathematical model of the degradation processes of machinery. The modeling process tends to be difficult because in-depth knowledge and rich experiment are required

[7]. Thus, the methods are less used [6]. Compared with physics-model based approaches, data-driven approaches are easier to be implemented. They often extract features reflecting the failure through mining the historical data. Statistical model based approaches such as AR models [10] and random coefficient models [11] describe the degradation process by statistical or stochastic models. The methods can depict the uncertainty of the degradation process and its influence on RUL prediction [6]. However, they are not good at modeling for nonlinear systems, and AI approaches such as ANNs [12, 13], NF systems [14], and SVM [15] can deal with the issue. They operate RUL prediction through feature extracting and data training without building the specific model. In recent years, research about AI approaches for RUL prediction has gained more and more attention. Hybrid approaches combine physics-model based approaches and data-driven approaches to take advantage of different approaches.

RNN (recurrent neural network) is a kind of AI approaches for RUL prediction, and it performs well for forecasting tasks of time series data [16]. But the use of RNN is limited due to the “fading memory” problem caused by

gradient explosion or gradient dispersion. A variant of RNN called LSTM (long short-term memory) was proposed [17, 18]. Lots of research work based on LSTM has been carried out, and good results are acquired [19–22].

Nevertheless, the traditional RNNs possess disadvantages of computational burdens and being time consuming because numerous parameters need to be trained. ESN (echo state networks) can provide a solution to deal with the problem [16]. ESN was first proposed in 2001 [23] and it is a relatively new type of RNN. ESN has a randomly generated reservoir, which remains unchanged during the training phase and only a readout is trained [24]. The characteristic makes ESN consumes much less time for the training process compared with traditional RNN such as LSTM. In-depth research for RUL prediction based on ESN still needs to be carried out [16]. Most of the existing research concentrates on optimization of the architecture and parameters of ESNs to obtain a better prediction result [25, 26].

However, existing research lacks attention to the analysis and mining of the original data. To deal with this, decomposing the raw time series into subsequence is an effective solution [27–30]. Reference [28] proposed an SDA (secondary decomposition algorithm) to decompose the original wind speed data into detailed components twice. The decomposition algorithms include WPD (wavelet packet decomposition) and FEEMD (fast ensemble EMD). Then an Elman neural network performed the prediction. In [29], to mitigate the problem that the model cannot handle environmental factors, a hybrid EMD-based prediction model was proposed for wind speed and solar irradiation forecasts. In [30], for chaotic time series prediction, PE (permutation entropy) was adopted to analyze the complexity of the IMFs decomposed by EEMD (ensemble EMD). The decomposed IMFs with similar complexities were combined, so fewer inputs for prediction were gained. Then, the ESNs performed the prediction and the ultimate results were assembled.

Although the combination of signal decomposition and ESN have performed well for forecasting problems, in recent years, few related studies have been conducted. As far as the authors know, they are not yet applied to RUL prediction. In most cases, the prediction of RUL is based on multisensor data and a large amount of noise needs to be filtered out. Signal decomposition methods can handle noise well exactly. Inspired by this, this paper proposes a novel combination method for RUL prediction based on EMD and ESN. EMD can smooth the noisy data and decompose the time series into a bunch of IMFs, which are then predicted by ESNs. The outputs of the ESNs are summarized as the final RUL prediction results. Besides, data processing methods including GBDT feature selection and Kalman filtering are also adopted. The proposed method is verified through a case study on turbofan engines of aircraft. The relevant data of turbofan engines is collected from multiple sensors under variable operating conditions. The proposed method is compared with ESN and LSTM [19]. The results demonstrate the superiority of our proposed method. The contributions of this paper are as follows: (1) We introduce the idea of signal decomposition to multisensor data processing for

RUL prediction. The adoption of EMD has a good effect on noise disposal, which provides a reference solution for reducing the noise of the RUL prediction problem. (2) We propose a combination method of EMD and ESN for RUL prediction, which provides a new idea for RUL prediction methods based on ESN. (3) We conduct a lot of experiments on C-MAPSS datasets. During the process, some data processing skills such as GBDT feature selection and Kalman filtering are adopted. The results show that our proposed method is well performed and effective for RUL prediction.

The rest of this paper is organized as follows. Section 2 elaborates the methodology and presents our method. Section 3 demonstrates our experimental process. Meanwhile, a comparison of models and results analysis is also given. In Section 4, conclusion and highlights of future work are drawn.

## 2. Methodology

**2.1. Empirical Mode Decomposition.** The EMD [31] is the core component of the Hilbert–Huang transform. It is an adaptive time-frequency analysis method for processing nonlinear and nonstationary time series. EMD can decompose time series into a group of IMFs without analysis in advance. An IMF should satisfy two conditions: (1) Throughout the data segment, the number of the extreme points and that of the zero crossing points should be equal or the difference is no more than 1; (2) The mean of the upper envelope formed by the local maximum points and the lower envelope formed by the local minimum points is zero. The detailed steps of the EMD decomposition process are described below.

- (1) Assume the time series to be analyzed is  $x(t)$ . Extract all the local maxima and minima.
- (2) Fit the maximum value points and minimum value points by cubic spline interpolation, respectively, to form the maximum value envelope  $e_{\max}(t)$  and the minimum value envelope  $e_{\min}(t)$ . The mean value of the extreme envelopes  $m(t)$  can be obtained:

$$m(t) = \frac{(e_{\max}(t) + e_{\min}(t))}{2}. \quad (1)$$

- (3) Subtract the mean value  $m(t)$  of the envelope from the original time series  $x(t)$ , and record the result as  $y_1(t)$ .

$$y_1(t) = x(t) - m(t). \quad (2)$$

- (4) Determine whether  $y_1(t)$  is an IMF component according to the above-mentioned judgement criteria of the two conditions. If  $y_1(t)$  does not meet the conditions, treat  $y_1(t)$  as the original data and repeat the above steps until  $y_1(t)$  satisfies IMF conditions. At this time, take  $c_1(t) = y_1(t)$  and  $c_1(t)$  as the first IMF component of the time series  $x(t)$ .

(5) Take the remaining data  $r1(t) = x(t) - c1(t)$  as the original time series and decompose it for  $k$  times with the same principles as above to obtain the  $k$ -order IMF and a residual. The stopping criterion includes the following: (i) The residual  $r(t)$  is a monotone function; (ii) the  $k$ th IMF or the residual  $r(t)$  is smaller than the specified threshold; (iii) the number of zero crossing points and extrema values differs by 1 at most.

(6) Finally,  $x(t)$  can be described as follows:

$$\mathbf{x}(t) = \sum_{k=1}^n c_k(t) + r(t), \quad (3)$$

where  $c_k(t)$ ,  $r(t)$  and  $n$  represent the  $k$ th IMF, the residual, and the number of IMFs, respectively.

## 2.2. Echo State Network

**2.2.1. Basic Structure of ESN.** ESN is a relatively new type of RNN, which replaces a reservoir for the hidden layers of RNN [23]. It is able to catch the system's dynamic behaviours and has intrinsic memory properties [16]. ESN uses the reservoir to map input space to feature space, in which neurons connect with each other randomly and sparsely. Input, feedback, and connection in the reservoir matrix are generated at random and remain unchanged during the training process, so the only variable that needs to be trained is the output matrix, which realizes computational savings.

As shown in Figure 1, ESN consists of a  $K$  units' input layer, a  $N$  internal units' reservoir, and a  $M$  units' output layer. The directed arrows represent the weight connections of neurons. The solid arrows indicate required connections, while the dashed arrows denote that connections are possible but not required. At time step  $t$ , variables of the input units are  $u(t) = (u_1(t), u_2(t), \dots, u_K(t))^T$ , of the internal units are  $x(t) = (x_1(t), x_2(t), \dots, x_N(t))^T$ , and of the output units are  $y(t) = (y_1(t), y_2(t), \dots, y_L(t))^T$ . The input weight matrix  $W^{\text{in}} \in R^{N \times K}$  represents the weights from input units to internal units, and the internal weight matrix  $W \in R^{N \times N}$  represents the weights between the internal units and the feedback weight matrix  $W^{\text{back}} \in R^{N \times L}$  represents the weights from the output units to the internal units. Activation functions such as sigmoid function and hyperbolic tangent function are adopted in the reservoir and the output layer. And at time step  $t$  the reservoir state can be acquired as follows:

$$x(t) = f(W^{\text{in}}u(t) + Wx(t-1) + W^{\text{back}}y(t-1)), \quad (4)$$

where  $f = \{f_1, f_2, \dots, f_N\}$  are activation functions of the internal units. The ESN's output equation can be obtained as follows:

$$y(t) = f^{\text{out}}(W^{\text{inout}}u(t) + W^{\text{out}}x(t) + W^{\text{outout}}y(t-1)), \quad (5)$$

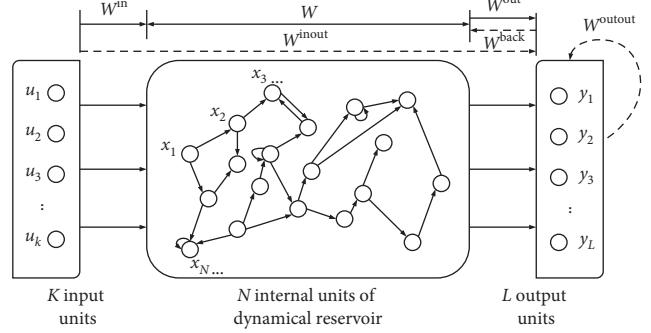


FIGURE 1: Basic structure of ESN.

and if we use  $W^{\text{out\_gen}} = \{W^{\text{inout}}, W^{\text{out}}, W^{\text{outout}}\}$  to indicate all the connections to the output neurons, then the above formula can be given as follows:

$$y(t) = f^{\text{out}}(W^{\text{out\_gen}}[u(t); x(t); y(t-1)]), \quad (6)$$

where  $f^{\text{out}} = \{f_1^{\text{out}}, f_2^{\text{out}}, \dots, f_M^{\text{out}}\}$  are also activation functions for the output layer which are often chosen to be linear. Moreover,  $[u(t); x(t); y(t-1)]$  is a concatenation of the input state, internal state, and the output state of the last time step. The dimension of the output weights  $W^{\text{out\_gen}}$  is  $M \times (K + N + M)$ .

Thus, we can use  $\text{ESN} = \{W^{\text{in}}, W^{\text{inout}}, W^r, W^{\text{out}}, W^{\text{outout}}, W^{\text{back}}\}$  to depict the ESN and the weights  $W^{\text{inout}}$ ,  $W^{\text{outout}}$ , and  $W^{\text{back}}$  are optional. All the weights except the output weights  $W^{\text{out\_gen}}$  are initialized randomly and remain unchanged during the training process. Only the weights  $W^{\text{out\_gen}}$  need to be trained.

## 2.2.2. Key Parameters of ESN

- (1) Reservoir size ( $N$ ): The reservoir size  $N$  represents the number of neuron units in the reservoir, which is very important and has a great influence on the performance of ESN [32]. If  $N$  is chosen to be too small, the network may not fit the expected output fully, while a too large reservoir may cause data overfitting and calculation consumption. The choice of  $N$  should take the complexity and effectiveness of the network into account to meet specific task requirements.
- (2) Spectral radius ( $\rho(W^r)$  or SR). The spectral radius represents the maximum absolute eigenvalue of the internal weight matrix ( $W$ ) of the reservoir. It decides the memory capacity of the ESN. If it is too small, the previous input has little effect on the present output, and the reservoir has a poor memory. If it is too large, the state of the reservoir may be unstable during the iteration process. It is not sufficient but necessary that the setting  $\rho(W^r) < 1$  can guarantee ESP (echo state property) in most cases [33]. ESP can ensure the network state is uniquely decided by the input history [26].

- (3) Sparsity ( $D$ ): Different from most neuron networks, the neurons in the reservoir are not fully connected and the sparsity denotes the sparse degree of the neurons in the reservoir. In a general way, the  $D$  is selected around 0.1 to guarantee the reservoir has enough dynamic properties [34]. When  $D$  is selected as 1, the neurons in the reservoir is fully connected and the ESN has evolved into a traditional RNN at this time.
- (4) Input scaling ( $\alpha_{W^{\text{in}}}$ ): The input scaling factor  $\alpha_{W^{\text{in}}}$  is a parameter that can realize a scaling transform to the input weight matrix  $W^{\text{in}}$  and it influences the degree of linearity of the responses of the reservoir units [35]. The larger  $\alpha_{W^{\text{in}}}$  is, the more nonlinear the response is. It is usually chosen to be within the interval  $[0, 1]$ .

**2.2.3. Training Steps of ESN.** Assume that a training sample is  $(u(t), y(t))$ , which contains  $t$  time steps. The general training steps of ESN are depicted below.

- (1) Initialize the parameters of ESN. The ESN weights  $\{W^{\text{in}}, W^r, W^{\text{out}}\}$  are randomly initialized first. To satisfy the requirement of ESP (echo state property), the  $W^r$  is scaled by the scaling factor  $\alpha_{W^r}$  to meet  $\rho(W^r) < 1$ . The state variables  $\mathbf{x}(0)$  also need to be initialized.
- (2) Update and collect the internal states of the reservoir. The internal states can be updated according to equation (4) driven by input signals  $\mathbf{u}(t)$  and the internal state of the last time step  $\mathbf{x}(t-1)$ . The state variables before the time step  $k_{\min} - 1$  should be abandoned to eliminate the influence of initial states on the network performance, which is called the washout phase. The state variables after  $k_{\min}$  are collected.
- (3) Compute the output weights. The output weights  $W^{\text{out}}$  are computed according to equation (5) to minimize the target function.

**2.3. Combination Model Based on EMD-ESN.** A combination model based on EMD and ESN is adopted in this paper, comprising decomposition of EMD, prediction of the ESNs, and summation of the separate outputs. The process is demonstrated in Figure 2.

First, the raw time series has been denoised to obtain cleaner results, which are then taken as inputs to be decomposed by EMD into IMFs adaptively. The IMFs are arranged from high to low frequency and the complexity of the IMFs is reduced greatly; thus the analysis and modeling of each IMF are much easier.

Second, we use multiple ESNs to train and predict decomposed IMFs. The training steps are illustrated in Section 2.2.3. A sampling window is used to construct samples of the training and testing datasets, which are taken as inputs to train the ESNs. Each ESN gives an output.

Finally, the outputs of the ESNs are assembled to obtain the final prediction result of the RUL.

### 3. Experimental Study

#### 3.1. Experimental Setup

**3.1.1. C-MAPSS Dataset.** The dataset used to support the findings in this paper have been deposited in the “Turbofan Engine Degradation Simulation Data Set,” NASA Ames Prognostics Data repository [36], which contains simulated degradation data for turbofan engine [37]. There are 4 sub-datasets in the C-MAPSS dataset denoted as FD001, FD002, FD003, and FD004. Each dataset contains multivariate time series data under different kinds of operational conditions and fault modes. It is also separated into a training set and a testing set, as shown in Table 1. Each row is a snapshot of the run-to-failure records taken within a single time cycle. It contains 26 columns, which represent an engine ID, a current operational cycle number, 3 operational settings, and 21 sensor values, respectively. Each engine unit’s initial state is unknown but is considered to be healthy. As the operational cycle number increases, the engine units begin to degrade at some point, which is also unknown. For the training set, the engines run till a failure occurs and the whole snapshots of degradation data can be acquired. While for the test set, the degradation ends sometime prior to a failure. The goal is to predict the number of remaining operational cycles (RUL) for the test set to verify our proposed method.

#### 3.1.2. Data Preprocessing

**(1) Sensor Data Selection and Normalization.** There are 21 sensor measurements contained in the C-MAPSS dataset. However, not all variables are closely related to the RUL prediction. We analyze the correlation between sensor data and the RUL values to select important features using GBDT (gradient boost decision tree). GBDT is a kind of ensemble learning algorithm with the decision tree as its basic estimator. It performs well when used for feature selection and can output the relative importance of the features, which is shown in Figure 3. The results are ranked according to relevance from high to low, and the top 10 sensor measurements are selected as the data to be analyzed next.

As different ranges of sensor values are generated, min-max normalization has been used to normalize the original data and limit it within the interval  $[0, 1]$ .

$$x_{i,j}'(t) = \frac{x_{i,j}(t) - \min(x_{i,j})}{\max(x_{i,j}) - \min(x_{i,j})}, \quad (7)$$

where  $x_{i,j}(t)$  is the original sensor measurement of the  $j$ th sensor of the unit  $i$ .  $\min(x_{i,j})$  and  $\max(x_{i,j})$  represent the minimum value and the maximum value of the unit  $i$ .  $x_{i,j}'(t)$  is the normalized value.

**(2) Samples Constructed through Time Window.** In order to construct training samples of the ESNs, a time window is used, sliding along the processed sensor data. As shown in

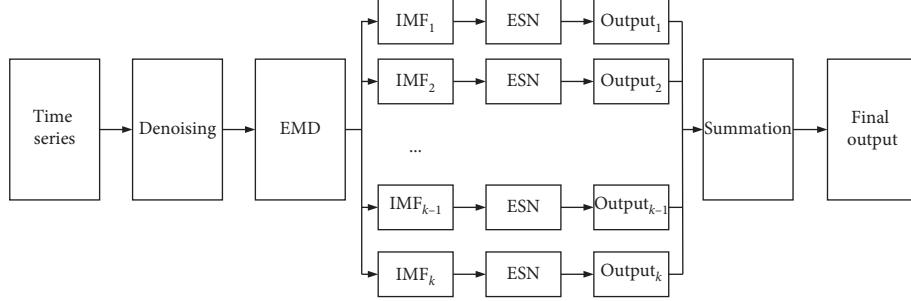


FIGURE 2: Flowchart of the combination model based on EMD-ESN.

TABLE 1: Information of the C-MAPSS dataset.

C-MAPSS data set	FD001	FD002	FD003	FD004
Engine units for training	100	260	100	249
Engine units for testing	100	259	100	248
Operating conditions	1	6	1	6
Fault modes	1	1	2	2

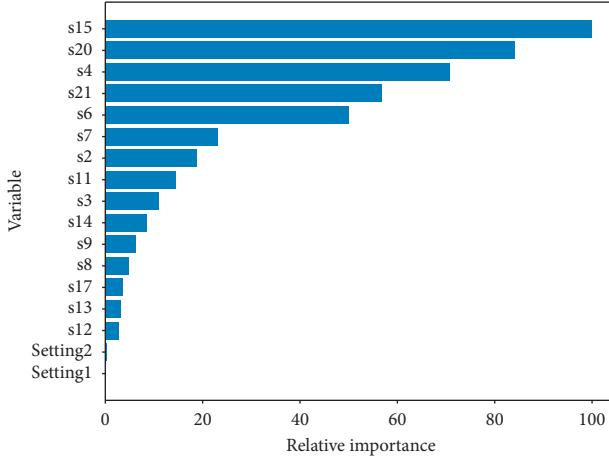


FIGURE 3: Relative importance ranking of the features in train\_FD001 by GBDT.

Figure 4, the length of the time window is time\_step. Each time the time window slides forward for one time-unit, an overlap exists in the two adjacent samples. The RUL value of the last time step is taken as the label of the training sample. Assume that the maximum life cycle of the equipment is  $L$  and the length of the time window is  $l$ , then the number of training samples we can obtain is  $L - l + 1$ .

**3.1.3. Performance Metrics.** To evaluate the performance of our proposed method, a score function and RMSE are used at the same time in this paper. Equation (8) denotes the definition of the scoring function.

$$\begin{cases} \sum_{i=1}^n \left( e^{-(h_i/13)} - 1 \right), & \text{when } h_i < 0, \\ \sum_{i=1}^n \left( e^{-(h_i/10)} - 1 \right), & \text{when } h_i \geq 0, \end{cases} \quad (8)$$

where  $n$  is the total number of the testing data samples, and  $h_i = \overline{\text{RUL}}_i - \text{RUL}_i$  (for the  $i$ th data point estimated RUL – true RUL). More penalty is given to overestimation than early predictions by the function since in a real industrial scene, late maintenance may result in more severe consequences than untimely maintenance.

Root Mean Square Error (RMSE) is defined as equation (9), which is a widely used evaluation metric for RUL estimation. RMSE gives an equal penalty to both early and late predictions. The difference between the scoring function and the RMSE function can be seen in Figure 5.

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n h_i^2}. \quad (9)$$

**3.1.4. RUL Target Function.** In practical prognostic applications, the accurate RUL values of the turbofan engines are unknown, which is different from our simulation experiment data as it contains the specific RUL value of each cycle [22]. In a system, the degradation of the equipment for the initial period of time is negligible until it runs for a certain time. So, the initial state of the engines is often taken as healthy and a piecewise linear RUL target function is adopted for the prediction. As shown in Figure 6, the maximum RUL value is limited to a constant value of 130 cycles [19, 38], which represents that we neglect degradation in the initial cycles and the linear decline of the RUL value starts from the time cycle of the number 130.

**3.1.5. Prognostic Procedure.** The prognostic procedure can be depicted in Figure 7. First, the raw data is preprocessed including Kalman filtering, data normalization, and sensor data selection to obtain cleaner data. Then, the processed data is selected through time window sliding and training samples are constructed. Next, the time series is decomposed by EMD and multiple IMFs and a RES are obtained, which are then trained by ESNs of the same number. Finally, the output of each ESN is added up to get the final prediction result.

### 3.2. Experimental Procedure and Performance Analysis

**3.2.1. Experimental Procedure and Comparation.** In this section, we take FD001 dataset as an example to demonstrate

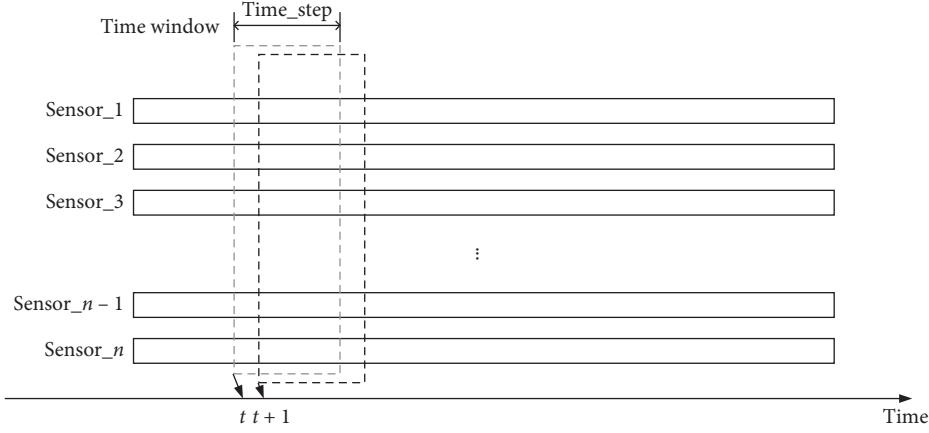


FIGURE 4: Training samples constructing through a time window.

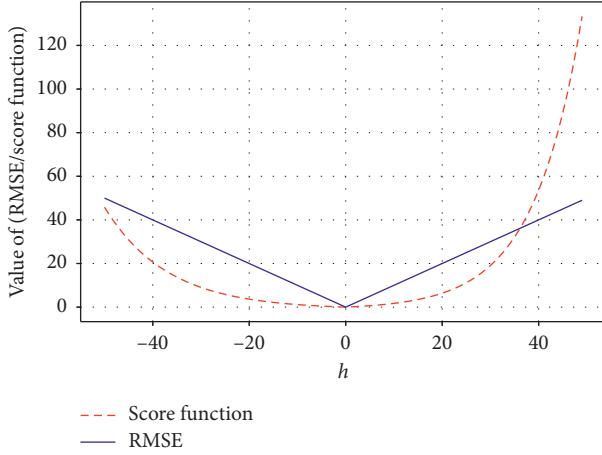


FIGURE 5: Comparison of the scoring function and RMSE.

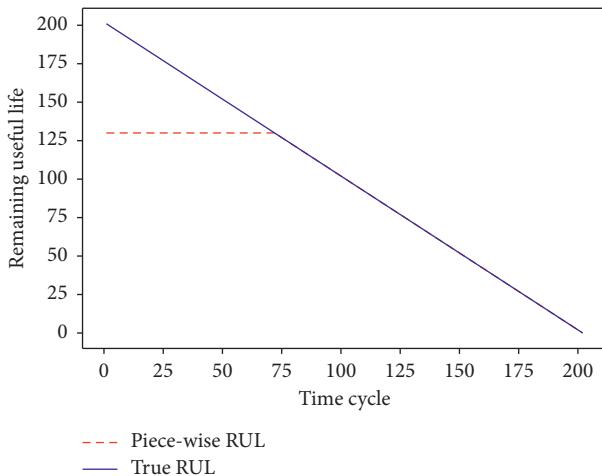


FIGURE 6: Piecewise RUL of C-MAPSS data set (the maximum RUL is 130 time cycles).

the experimental procedure. Then, the proposed method is employed on the C-MAPSS data set and is compared with LSTM [17] and basic ESN. Finally, a results analysis is given.

(1) *The Time Window Length Selection Experiment.* An experiment is carried out to select the most appropriate length of the time window to construct training samples. As we can see in Figure 8, the corresponding relations between the time step and RMSE are shown. As the time step increases, the trend of RMSE is generally downward. This can explain that to a certain degree, the more historical data training samples take, the more accurate the prediction is. The minimum cycles of the test dataset are only 31, so we choose 30 as the length of the time window in this paper. Besides, when the length grows more than 30, the decline of RMSE is not that much. In some research based on the C-MAPSS data set, the length of the time window is also chosen to be no more than 30 [19, 21, 22, 25, 26].

(2) *The EMD Decomposition.* After the process of feature extraction by GBDT and sampling by the time window, the 10 sensor measurements are then decomposed by EMD. The decomposition result of the training set of FD001 is shown in Figure 9. There are 4 IMFs and a RES generated, which have simpler information each and can be synthesized into the original time series. The decomposition results are taken as the input of the ESNs next. Much noise can be filtered out during the decomposition process, which is beneficial for prediction improvement.

(3) *The ESN Training Experiment.* As depicted in Sections 2.2.2 and 2.2.3, to train ESN, we need to set some parameters, i.e., the reservoir size  $N$  which influences dynamics of the system, the spectral radius SR which decides the ESP (echo state property) of the reservoir, the sparsity  $D$  to represent the connection degree of the neurons, and the input scaling  $\alpha_{\text{Win}}$  to scale the input signal. Besides, in this paper, the regularization coefficient  $\lambda$  is used to prevent overfitting. With the grid search strategy, the parameters of the ESN are set as the values in Table 2.

The training process of the ESN can be divided into two subprocesses, which includes preliminary training and weights calculation. The preliminary training process is for the purpose to eliminate the effects of the initial network state on the training process. The weights calculation process

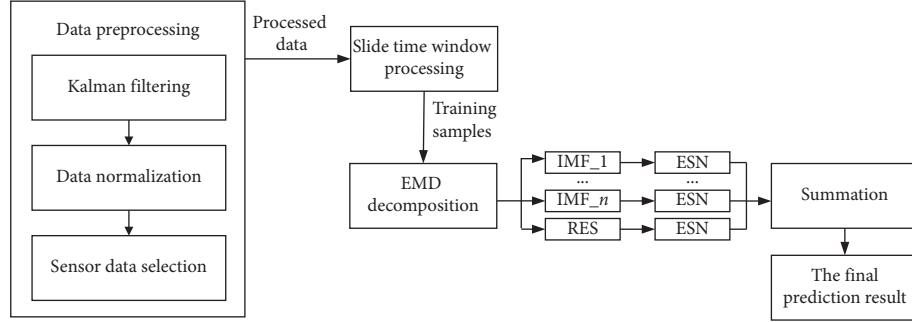


FIGURE 7: Flow chart of the prognostic procedure.

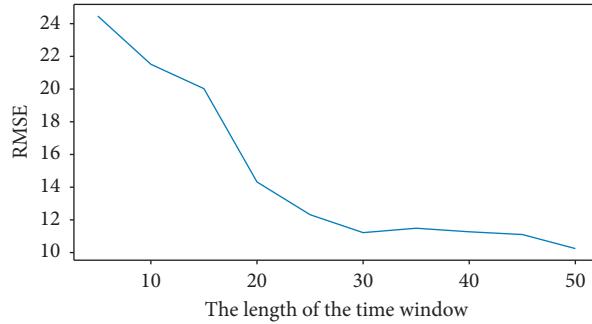


FIGURE 8: Influence of different time steps on the prediction accuracy in train\_FD001.

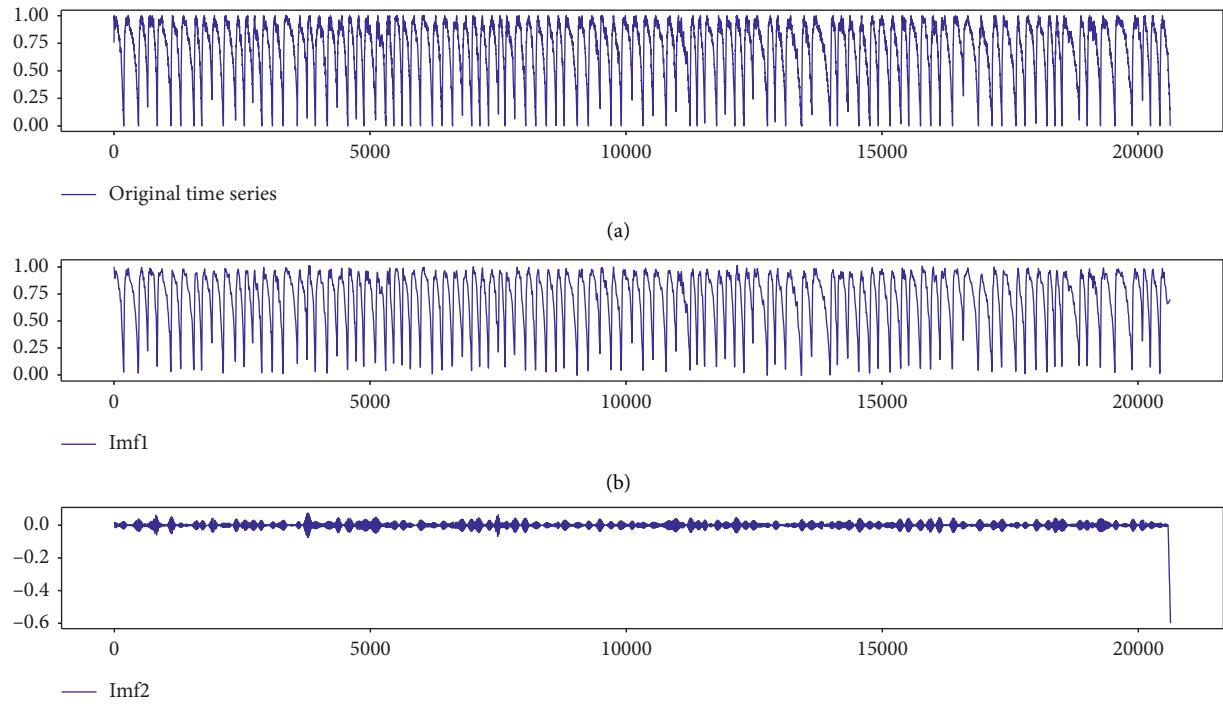


FIGURE 9: Continued.

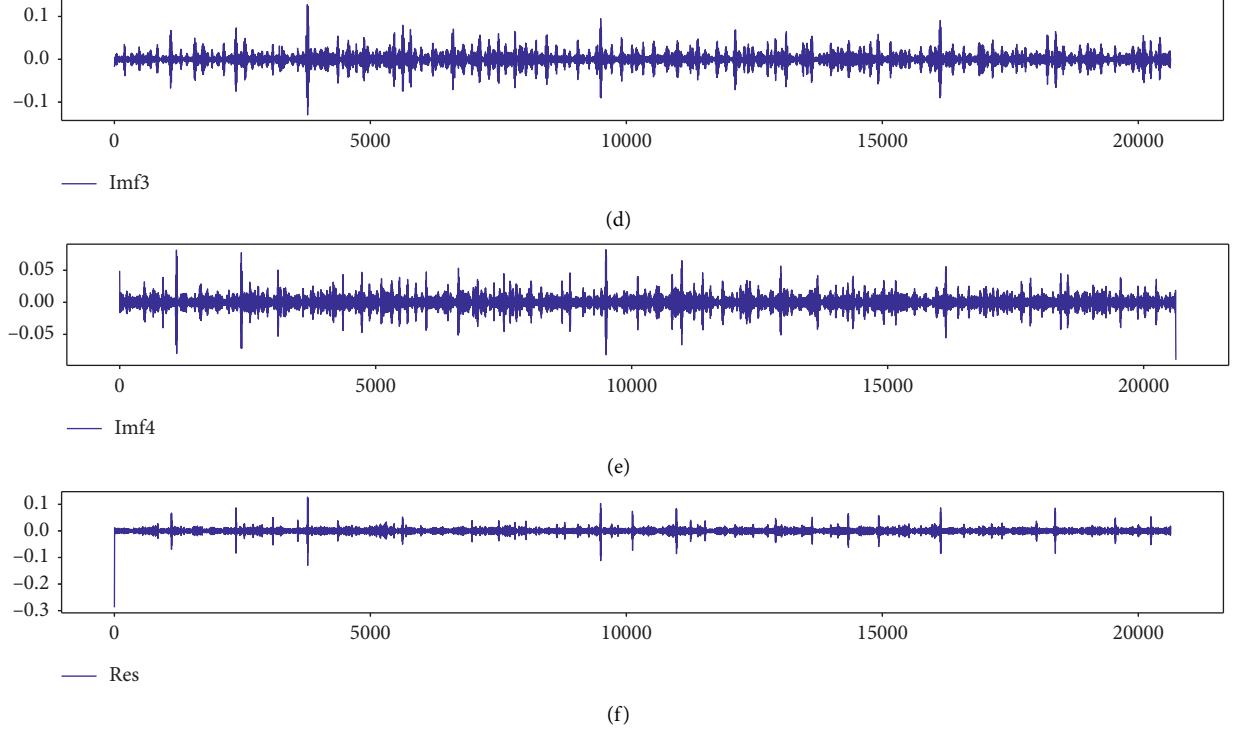


FIGURE 9: Decomposition results of the processed data in train\_FD001.

TABLE 2: Parameter setting of the ESN.

Optimal parameter	Value
Reservoir size $N$	0.9
Spectral radius SR	500
Sparsity $D$	0.2
Input scaling $\alpha_{W^m}$	0.05
Regularization coefficient $\lambda$	0.0001

is to calculate the  $W^{\text{out}}$  illustrated in Section 2.2.3, which is taken as a linear regression process. The learning algorithm adopted in this paper is RR (Ridge Regression). The hyperbolic tangent function is chosen as the activation function in this paper, which can be expressed as equation (10). The response curve is demonstrated in Figure 10. The activation can make the network have good nonlinearity.

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \quad (10)$$

The overall prediction results are shown in Figures 11 and 12, which demonstrate the prediction of ESN and EMD-ESN, respectively. As we can see in the figures, the models of ESN and EMD-ESN have both acquired good results. However, the prediction accuracy is enhanced by EMD-ESN. The circumstances of overpredicting have also been reduced, which is encouraged in the PHM (prognostics and health management) applications because delay maintenance may cause major damage and cost. Besides, it can be discovered that when the actual RUL values are relatively small, an overprediction situation is more likely to happen.

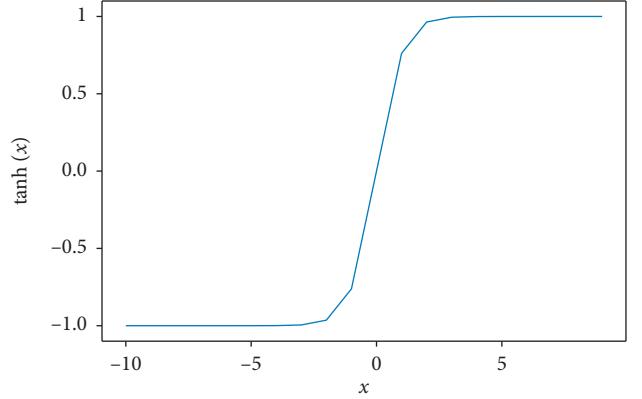


FIGURE 10: The response curve of the activation function.

One of the reasons for the problem may be that the amount of historical data is small.

**3.2.2. Results Comparison and Analysis.** In the experiment, the prediction performance and efficiency of EMD-ESN, LSTM, and ESN are compared. The experimental results and analysis are discussed below.

(1) *Results Comparison of Prediction Performance.* The results of the prediction performance of different models are revealed in Table 3. The performance metrics include Score and RMSE, which are depicted in Section 3.1.3. The length of the time window is 15 for LSTM and is 30 for ESN and EMD-ESN. Although more historical training data often

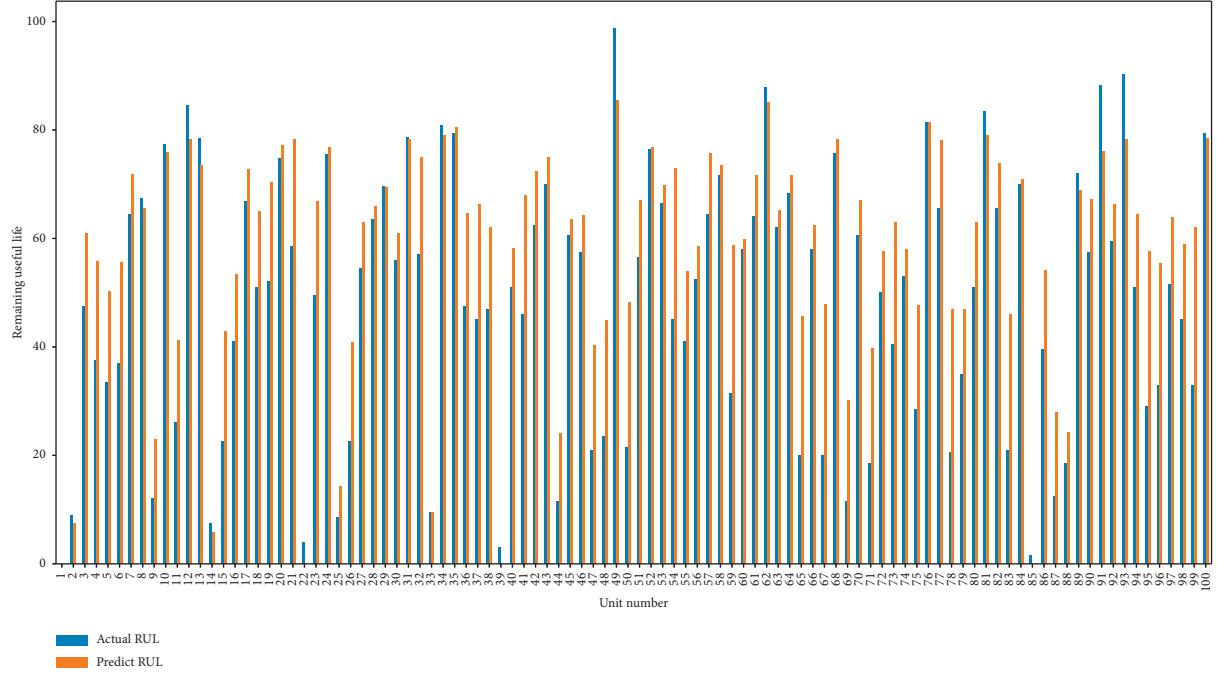


FIGURE 11: The prediction results of ESN for 100 units in test\_FD001.

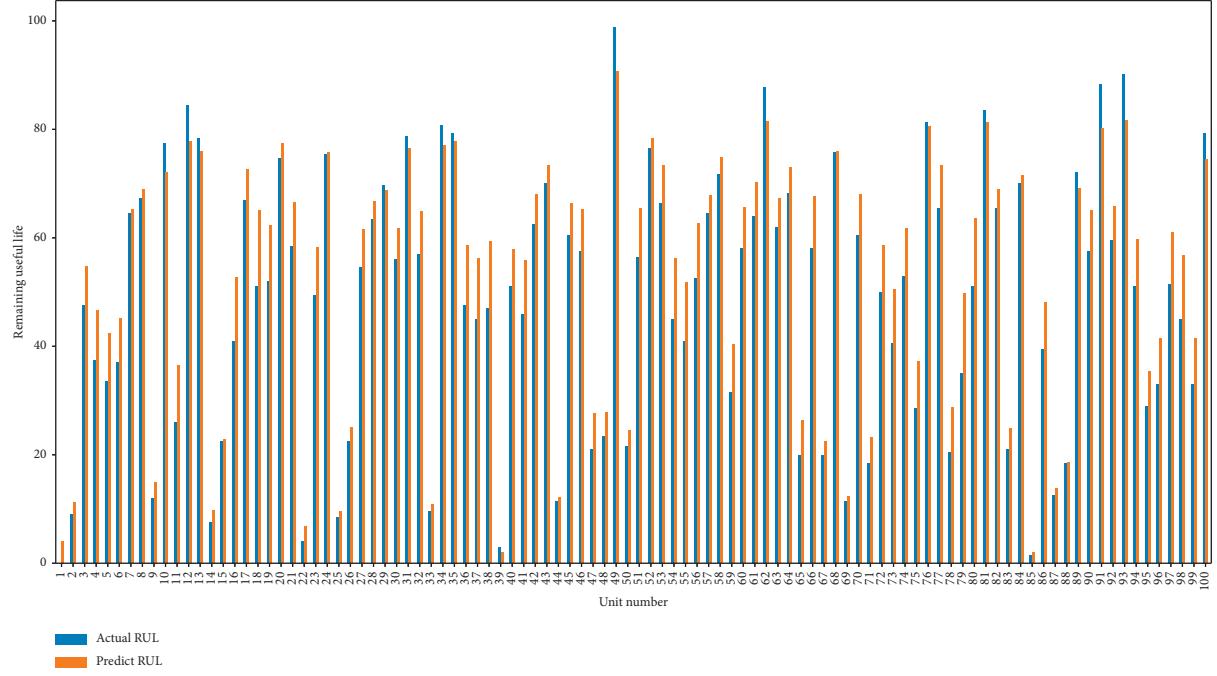


FIGURE 12: The prediction results of EMD-ESN for 100 units in test\_FD001.

TABLE 3: Prediction comparison on C-MAPSS data set. The IMP over LSTM represents an improvement of EMD-ESN over LSTM (IMP = 1 - EMD - (ESN/LSTM)), while the IMP over ESN represents improvement of EMD-ESN over ESN (IMP = 1 - EMD - (ESN/ESN)).

Data set	FD001		FD002		FD003		FD004	
	Score	RMSE	Score	RMSE	Score	RMSE	Score	RMSE
LSTM [21]	338	16.14	4450	24.49	852	16.18	5550	28.17
ESN	359	13.92	748	13.33	4971	23.72	2912	16.88
EMD-ESN	<b>184</b>	<b>10.63</b>	<b>293</b>	<b>8.85</b>	<b>299</b>	<b>11.54</b>	<b>525</b>	<b>9.87</b>
IMP over LSTM (%)	45.56	34.14	93.41	63.86	64.91	28.68	90.54	64.96
IMP over ESN (%)	48.75	23.64	60.83	33.61	93.99	51.35	81.97	41.53

TABLE 4: Training time comparison on C-MAPSS data set.

Data set	FD001 (s)	FD002 (s)	FD003 (s)	FD004 (s)
LSTM [21]	6540.53	17159.43	7649.63	15494.50
ESN	34.58	105.80	47.84	91.57
EMD-ESN	419.40	1318.06	508.97	1262.64

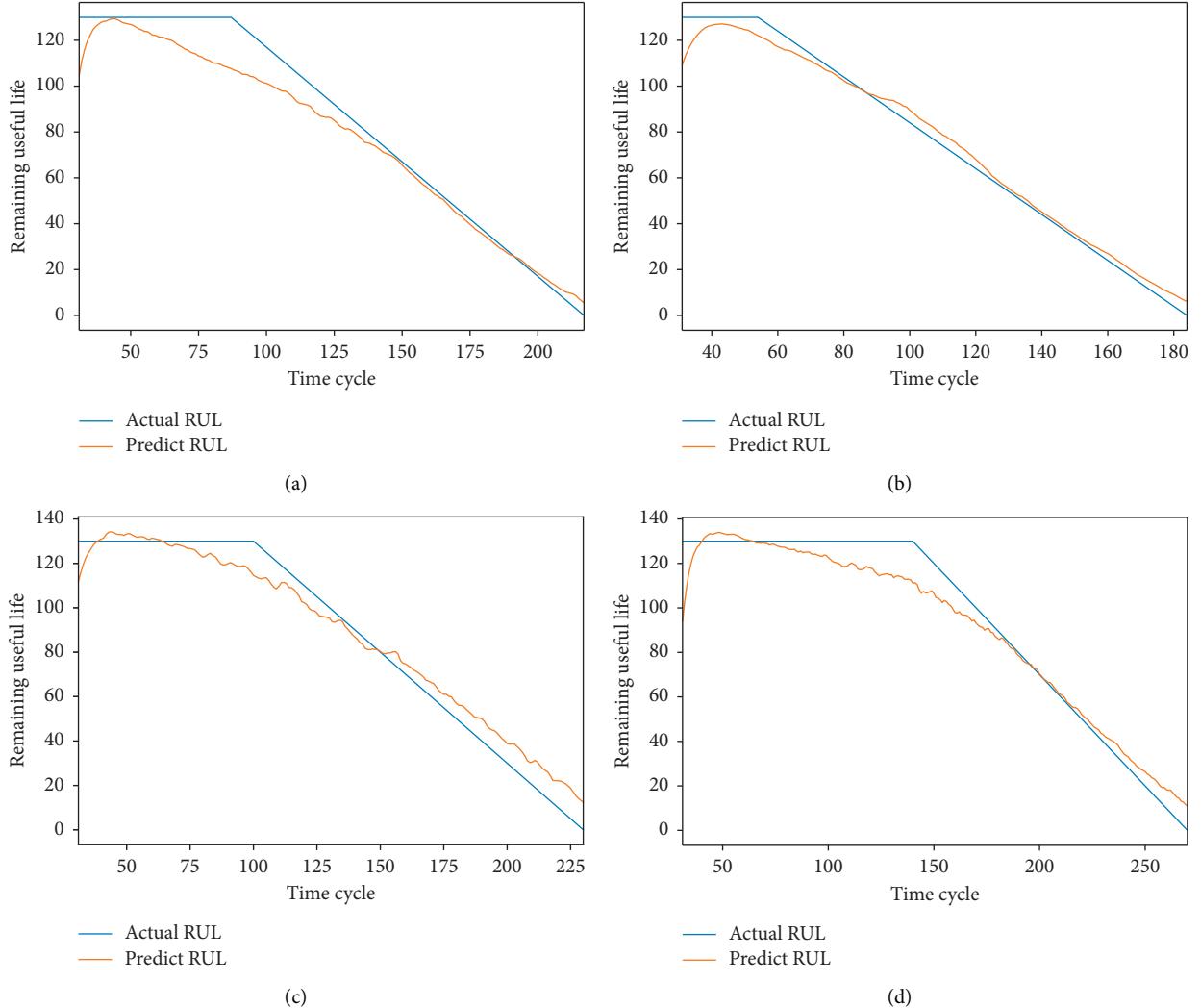


FIGURE 13: RUL prediction for engine unit examples of C-MAPSS dataset. (a) The example is unit 12 in test\_FD001. (b) The example is unit 7 in test\_FD002. (c) The example is unit 9 in test\_FD003. (d) The example is unit 12 in test\_FD004.

lead to improvement of results, we get a significant gain in prediction result of our proposed method. As we can see in Table 3, the EMD-ESN method proposed in this paper performs best on the four C-MAPSS datasets, whether it is for Score or RMSE. For Score, IMP over LSTM of our proposed method is up to 93.41% on FD002 and IMP over ESN is up to 93.99% on FD003. For RMSE, IMP over LSTM of our proposed method is up to 64.96% on FD004 and IMP over ESN is up to 51.35% on FD003. We can conclude that the proposed method can improve the prediction effect to a large extent. When we take Score as the performance metrics, the improvement effect is particularly obvious.

For RMSE, the ESN has also performed better than the LSTM on all the datasets. However, for Score the LSTM performs better than the ESN on FD001 and FD003. The situation demonstrates that overpredicting tends to be generated by the ESN, especially for FD001 and FD003, which contain just 1 operational condition. Different from the ESN, our proposed method has improved the over-predicting situation, which can also be seen by comparing Figure 12 with Figure 11. In Figure 12, the situations of overprediction are less than those in Figure 11, obviously. Besides, there is a substantial improvement for the EMD-ESN on FD004, which is the most complicated dataset

containing 6 operating conditions and 2 fault modes. This all shows that the EMD-ESN can filter out redundant signals or noise, in which EMD plays a crucial role. This inspires us that the signal decomposition technology can be adopted in the RUL prediction to deal with noise.

(2) *Results Comparison of Prediction Efficiency.* In addition to results comparison of prediction performance, the training time comparison is also performed, and the results are demonstrated in Table 4. As we can see in Table 4, the training time of the ESN and the EMD-ESN are significantly saved compared with the LSTM, which means fewer computing resources are required. Although the EMD-ESN still takes more training time than the ESN, which consumes a certain time for the EMD decomposition process, the prediction improvement is also considered, especially for data with much noise. This shows the necessity to adopt EMD to decompose the raw sensor data.

(3) *Sample Examples of RUL Prediction.* In Figure 13, the RUL prediction of a sample engine unit for each dataset is illustrated. We can see that the prediction for the four examples all perform well, with an unobvious trend for overpredicting. It can also be seen that for each unit, the preceding period prediction is generally not as good as the following period prediction. This can be explained by the fact that with the increase of the training data, the accuracy is becoming higher.

In this section, the prediction performance and efficiency are compared between LSTM, ESN, and the proposed EMD-ESN, and result analysis is performed. Through experiments, the superiority of our proposed method is verified.

## 4. Conclusions

Prognostic of RUL contributes to timely maintenance and less cost consuming. In this paper, we propose a combination method of EMD and ESN to predict the RUL. The proposed EMD-ESN method is performed on the turbofan engine multisensor time series, which demonstrates the validity, accuracy, and effectiveness of our method. The EMD decomposes the raw time series to obtain portions with simpler information, which can be easily trained by the ESN further. The output of each ESN is summarized to get the final prediction value. Compared with the LSTM and the basic ESN method, the prediction effect of the EMD-ESN method is greatly improved, especially for data with much noise. Besides, the computing time is largely reduced.

However, there is still a lot of research to be conducted and some of the directions are as follows:

- (1) Although the adoption of EMD improves the prediction effect, it also consumes more time. It is necessary to study how to improve prediction accuracy and save computing resources at the same time.
- (2) EMD is one of the signal processing methods in the time and frequency domain, and there are many other relative methods such as wavelet packet

decomposition. The adoption of relative methods into noise data processing is also worthy of further study.

## Data Availability

The dataset used to support the findings in this paper have been deposited in the "Turbofan Engine Degradation Simulation Data Set," NASA Ames Prognostics Data repository, which contains simulated degradation data for turbofan engine developed by NASA. The dataset can be found through the link <https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/>.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

## Acknowledgments

This research was funded by a major project of the National Natural Science Foundation of China, "Operation Optimization of Intelligent Factories for High-end Equipment Manufacturing under the Internet and Big Data Environment" (Project no. 71690230/71690234).

## References

- [1] J. Z. Sikorska, M. Hodkiewicz, and L. Ma, "Prognostic modelling options for remaining useful life estimation by industry," *Mechanical Systems and Signal Processing*, vol. 25, no. 5, pp. 1803–1836, 2011.
- [2] M. S. Kan, A. C. C. Tan, and J. Mathew, "A review on prognostic techniques for non-stationary and nonlinear rotating systems," *Mechanical Systems and Signal Processing*, vol. 62–63, pp. 1–20, 2015.
- [3] M. Compare, L. Bellani, and E. Zio, "Availability model of a PHM-equipped component," *IEEE Transactions on Reliability*, vol. 66, no. 2, pp. 487–501, 2017.
- [4] Y. Zhang, R. Xiong, H. He, and M. G. Pecht, "Lithium-ion battery remaining useful life prediction with box-cox transformation and Monte Carlo simulation," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 2, pp. 1585–1597, 2019.
- [5] L. Yongxiang, S. Jianming, W. Gong, and Z. Mengying, "An ensemble model for engineered systems prognostics combining health index synthesis approach and particle filtering," *Quality and Reliability Engineering International*, vol. 33, no. 8, pp. 2711–2725, 2017.
- [6] Y. Lei, N. Li, L. Guo, N. Li, T. Yan, and J. Lin, "Machinery health prognostics: a systematic review from data acquisition to RUL prediction," *Mechanical Systems and Signal Processing*, vol. 104, pp. 799–834, 2018.
- [7] M. S. K. Kopuru, S. Rahimi, and K. Baghaei, "Recent approaches in prognostics: state of the art," in *Proceedings of the International Conference on Artificial Intelligence (ICAI)*, pp. 358–365, Xuzhou, China, August 2019.
- [8] C. H. Oppenheimer and K. A. Loparo, "Physically based diagnosis and prognosis of cracked rotor shafts," in *Component and Systems Diagnostics, Prognostics, and Health Management II*, P. K. Willett and T. Kirubarajan, Eds., pp. 122–132, SPIE, Bellingham, WA, USA, 2002.

- [9] J. Sun, H. Zuo, W. Wang, and M. G. Pecht, "Prognostics uncertainty reduction by fusing on-line monitoring data based on a state-space-based degradation model," *Mechanical Systems and Signal Processing*, vol. 45, no. 2, pp. 396–407, 2014.
- [10] Y. Qian, R. Yan, and S. Hu, "Bearing degradation evaluation using recurrence quantification analysis and kalman filter," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 11, pp. 2599–2610, 2014.
- [11] R. Zhou, N. Gebraeel, and N. Serban, "Degradation modeling and monitoring of truncated degradation signals," *IIE Transactions*, vol. 44, no. 9, pp. 793–803, 2012.
- [12] L. Xiao, X. Chen, X. Zhang, and M. Liu, "A novel approach for bearing remaining useful life estimation under neither failure nor suspension histories condition," *Journal of Intelligent Manufacturing*, vol. 28, no. 8, pp. 1893–1914, 2017.
- [13] F. O. Heimes, "Recurrent neural networks for remaining useful life estimation," in *Proceedings of the 2008 International Conference on Prognostics and Health Management*, pp. 1–6, IEEE, Denver, CO, USA, October 2008.
- [14] D. Zurita, J. A. Carino, M. Delgado et al., "Distributed neuro-fuzzy feature forecasting approach for condition monitoring," in *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*, pp. 1–8, IEEE, Barcelona, Spain, September 2014.
- [15] X. Si, T. Li, Q. Zhang, and X. Hu, "An optimal condition-based replacement method for systems with observed degradation signals," *IEEE Transactions on Reliability*, vol. 67, no. 3, pp. 1281–1293, 2018.
- [16] M. Rigamonti, P. Baraldi, E. Zio, I. Roychoudhury, K. Goebel, and S. Poll, "Ensemble of optimized echo state networks for remaining useful life prediction," *Neurocomputing*, vol. 281, pp. 121–138, 2018.
- [17] J. Zhang, P. Wang, R. Yan, and R. X. Gao, "Long short-term memory for machine remaining life prediction," *Journal of Manufacturing Systems*, vol. 48, pp. 78–86, 2018.
- [18] S. J. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [19] S. Zheng, K. Ristovski, A. Farahat, and C. Gupta, "Long short-term memory network for remaining useful life estimation," in *Proceedings of the 2017 IEEE International Conference on Prognostics and Health Management (ICPHM)*, pp. 88–95, IEEE, Dallas, TX, USA, June 2017.
- [20] H. Miao, B. Li, C. Sun, and J. Liu, "Joint learning of degradation assessment and RUL prediction for aeroengines via dual-task deep LSTM networks," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 9, pp. 5023–5032, 2019.
- [21] C.-G. Huang, H.-Z. Huang, and Y.-F. Li, "A bidirectional LSTM prognostics method under multiple operational conditions," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 11, pp. 8792–8802, 2019.
- [22] A. Al-Dulaimi, S. Zabihi, A. Asif, and A. Mohammadi, "A multimodal and hybrid deep neural network model for remaining useful life estimation," *Computers in Industry*, vol. 108, pp. 186–196, 2019.
- [23] H. Jaeger, "The ‘echo state’ approach to analysing and training recurrent neural networks," vol. 148, German National Research Center for Information Technology, Sankt Augustin, Germany, 2001, Technical Report GMD Report 148.
- [24] M. Lukoševičius and H. Jaeger, "Reservoir computing approaches to recurrent neural network training," *Computer Science Review*, vol. 3, no. 3, pp. 127–149, 2009.
- [25] C. Liu, R. Yao, and L. e. a. Zhang, "Attention based echo state network: A novel approach for fault prognosis," in *Proceedings of the 2019 11th International Conference on Machine Learning and Computing*, pp. 489–493, Zhuhai, China, February 2019.
- [26] S. Zhong, X. Xie, L. Lin, and F. Wang, "Genetic algorithm optimized double-reservoir echo state network for multi-regime time series prediction," *Neurocomputing*, vol. 238, pp. 191–204, 2017.
- [27] Z. Tan, J. Zhang, J. Wang, and J. Xu, "Day-ahead electricity price forecasting using wavelet transform combined with ARIMA and GARCH models," *Applied Energy*, vol. 87, no. 11, pp. 3606–3610, 2010.
- [28] H. Liu, H.-q. Tian, X.-f. Liang, and Y.-f. Li, "Wind speed forecasting approach using secondary decomposition algorithm and Elman neural networks," *Applied Energy*, vol. 157, pp. 183–194, 2015.
- [29] Y. Wang and L. Wu, "On practical challenges of decomposition-based hybrid forecasting algorithms for wind speed and solar irradiation," *Energy*, vol. 112, pp. 208–220, 2016.
- [30] X. Xu and W. Ren, "A combination model based on EEMD-PE and echo state network for chaotic time series prediction," in *Proceedings of the 2019 Eleventh International Conference on Advanced Computational Intelligence (ICACI)*, pp. 290–295, Guilin, China, June 2019.
- [31] N. E. Huang, Z. Shen, S. R. Long et al., "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis," *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 454, no. 1971, pp. 903–995, 1998.
- [32] J. Qiao, S. Li, and W. Li, "Mutual information based weight initialization method for sigmoidal feedforward neural networks," *Neurocomputing*, vol. 207, pp. 676–683, 2016.
- [33] M. Lukosevicius, "A practical guide to applying echo state networks," in *Neural Networks: Tricks of the Trade*, pp. 659–686, Springer, Berlin, Germany, 2012.
- [34] H. Jaeger, M. Lukoševičius, D. Popović, and U. Siewert, "Optimization and applications of echo state networks with leaky-integrator neurons," *Neural Networks*, vol. 20, no. 3, pp. 335–352, 2007.
- [35] A. Bala, I. Ismail, R. Ibrahim, and S. M. Sait, "Applications of metaheuristics in reservoir computing techniques: a review," *IEEE Access*, vol. 6, pp. 58012–58029, 2018.
- [36] A. Saxena and K. Goebel, "Turbofan engine degradation simulation data set," *NASA Ames Prognostics Data Repository*, <http://ti.arc.nasa.gov/project/prognostic-data-repository>, NASA Ames Research Center, Moffett Field, CA, USA, 2008, <http://ti.arc.nasa.gov/project/prognostic-data-repository>.
- [37] A. Saxena, K. Goebel, D. Simon, and N. H. W. Eklund, "Damage propagation modeling for aircraft engine run-to-failure simulation," in *Proceedings of the IEEE International Conference on Prognostics and Health Management*, pp. 1–9, Denver, CO, USA, October 2008.
- [38] F. O. Heimes, "Recurrent neural networks for remaining useful life estimation," in *Proceedings of the 2008 International Conference on Prognostics and Health Management*, pp. 1–6, IEEE, Denver, CO, USA, October 2008.