RUL Prognostics Method Based on Real Time Updating of LSTM Parameters

Funa Zhou[1], Po Hu[1,*]

1. School of Computer and Information Engineering, Henan University, Kaifeng, China
E-mail: zhoufn2002@163.com, hupo4210@163.com (* Corresponding Author)

Abstract: Traditional LSTM model can not effectively use the non-life-cycle data to establish an excellent RUL Prognostics model since it can not utilize the online data reasonably. For small sampling data LSTM learning, this paper proposes an improved LSTM method with real-time parameters updating by using new online observation data to minimize the cost function. Taking NASA lithium-ion battery data as an example, the applicability of the improved LSTM model with real-time parameter updating in the field of remaining useful life Prognostics is verified.

Key Words: LSTM, RUL Prognostics, Real-time updating, Online model updating

1. INTRODUCTION

Equipment failure will bring great security risks, the RUL (Remaining Useful Life) Prognostics of device is very necessary. At present, Prognostics models for RUL can be categorized into three types: RUL Prognostics method based on empirical knowledge, RUL Prognostics method based on data-driven, RUL Prognostics method based on the mechanism model[1]. Knowledge-based approaches include expert system approach and fuzzy-based approach. Because these methods are too dependent on the prior knowledge of experts, they lead to poor generalization ability[2]. The methods based on mechanism model mainly include Markov-based method, Kalman Filter-based method, and particle filter-based method. Due to the complexity of current devices, it is difficult to establish an accurate physical model[3]. Data-driven approach is a hot research method, including statistical analysis methods and artificial intelligence methods[4]. Statistical learning methods mainly include trend extrapolation and AR methods[5]. Artificial intelligence methods mainly include ANN, SVM and deep learning methods[6]. The method of statistical learning to predict the remaining life by recursively after multi-step, it can not guarantee real-time, while the shallow artificial intelligence method has the problem of feature extraction is not accurate, so the deep learning based method is one of the most popular method about RUL prediction[7-9].

The traditional deep learning methods with stacked auto-encoder structure can extract more accurate features[10]. However, since the feature of RUL Prognostics is obviously time-dependent, the AE method can not extract the time-dependent features of time series data , recurrent neural network model (RNN) is proposed to establish the time-dependent relationship between data[11-13]. As time goes on the latter node's perceived ability of the previous node decreased, so RNN can not remember the previous information for a long time. Moreover, RNN has the problem of gradient disappearance in the process of error back propagation. In view of the above problems, long and short term memory neural network (LSTM) redesigns computing nodes by incorporating the previous information to extract auto-correlation feature, it is significant in speech recognition and other fields[14-16]. However, the literature of LSTM in equipment RUL Prognostics is very rare, especially the research of RUL Prognostics on equipment that plays a very crucial role in important fields is even less[17]. It is usually very difficult to collect the full life cycle data for those in service equipment, such as, the in service satellite. To use these precious part-life-cycle data, it is an important issue for taking the advantage of LSTM to establish an effective RUL Prognostics model [18-20].

Aiming at optimizing the network parameters,we propose an online updating mechanism of LSTM learning by minimizing the cost function. Thus, an effective RUL Prognostics model can be established without full life-cycle data of the equipment. With the increase of equipment service time, the established model parameters are gradually optimized by using the online data to update the trained LSTM based simply on history data. By comparing with many typical Prognostics models, it is verified that the proposed LSTM model with real-time updating has higher RUL prognostics accuracy in the case of only part-life-cycle is available.

The remaining part of this paper is organized as follows: Section 2 introduces LSTM theories and techniques; Section 3 introduces the proposed LSTM Prognostics model and its parameters real-time update method; Section 4 presents the experimental results and analysis of their results; Sections 5 is the conclusion.

2. LSTM THEORY AND PROGNOSTICS METHODS

RNN is an improved multi-layer neural network, which consists of input layer, hidden layer and output layer, and it can solve the time-dependent problem. RNN can be

This research was supported in part by the Natural Science Fund of China (Grant No. U1604158)

thought of as multiple replications of the same neural network, with each neural network module propagating message to the next. When it is expanded in time, as shown in Fig. 1, it can be seen that RNN is a chained structure, and chained features indicate that it is related to sequences and lists. The output of current moment is related not only to the current input time but also to the previous output, which allows the previous information propagate backwards. This is the reason why RNN can solve time auto-correlated problem. The parameter training of RNN is realized by BPNN algorithm. However, one of the most obvious problems of RNN is that the gradient of training process needs to be back propagated. When the input sequence is relatively long, there will be gradient disappearance and gradient explosion problem.

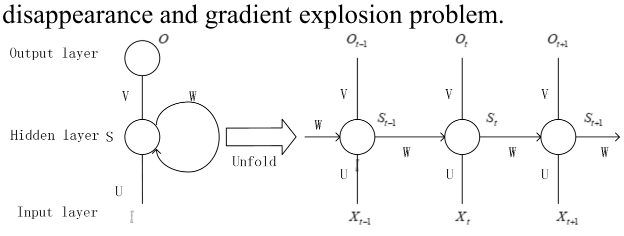


Fig.1 Unfolding of single RNN hidden layer unit

LSTM can solve the above problems of RNN by introducing a group of memory cells, allowing the network to learn when to forget the historical information and when to update the memory cells with new information. LSTM and RNN have the same chained structure, except that the hidden layer in the standard RNN has only a simple tanh layer, while there are four interactive layers in LSTM. Structure of LSTM is shown in Fig. 2.

The state of the cell is the most critical component of LSTM. The state of the cells resembles the carousel, and information can circulate throughout the chain. LSTM has the ability to remove or increase information to the state of the cell due to its well-designed gate structure. The gate provide a selection way in information propagation. The key of the gate structure is a sigmoid neural network layer and a pointwise multiply operation. The output of the sigmoid layer is between 0 and 1, 0 means that all the previous information not pass the gate, and 1 means all pass. 4 gates of LSTM, forgotten gate, input gate, output gate,can be used to control the status of each cell.

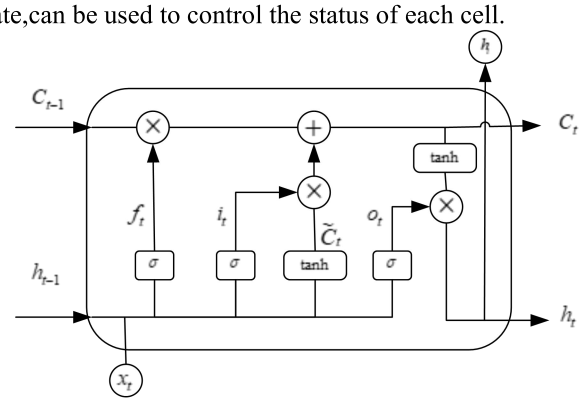


Fig.2 Structure of LSTM unit

LSTM network training process can be described as follows:

Step1: Preprocess training data(normalization and other operations).
Step2: Initialize the weights and offsets of the network, and the hyper paragraphs of the network.
Step3: Calculate the LSTM forward operation process shown in formula 5 to obtain the LSTM Prognostics value.
Step4: Compute the error between the predicted value and the actual value.
Step5: To determine whether the training process reach the error threshold or the maximum number of iterations. If exit conditions reached, exit the training, or use error back propagation algorithm to update the network parameters, and then go train to Step3. LSTM work flow is shown in Figure3.

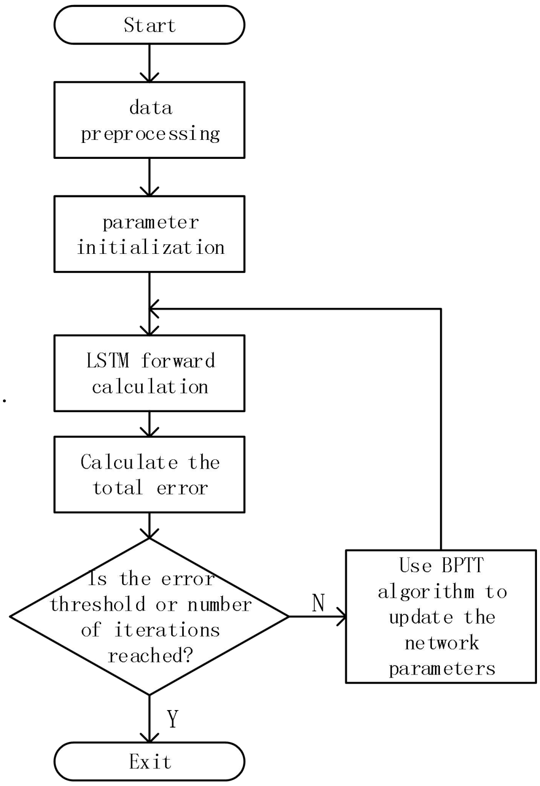


Fig.3 The Training Process of LSTM

3. LSTM PROGNOSTICS MODEL WITH REALTIME UPDATING OF PARAMETERS

Based on practical problems, this paper update the network parameters using the online data to establish a more effective RUL Prognostics model in the case when only part-life-cycle observation data in available. First of all, the LSTM Prognostics model should be established by making use of the known historical data reasonably. Secondly, when the actual online data is collected, the predicted value can be obtained by using the established Prognostics model. New observation data of the next sample time is used as the true value of the previous sample time. The error between the predicted value and the real value is added to the overall sample error. Then, the model parameters are iteration updated by using the error minimization method. By updating the parameters in this way, the model can be more and more accurate with more and more online data used. This modeling idea is more conducive to actual system online monitoring as well as failure warning when the predicted value reaches the point of failure.

The first step in the ILSTM(improved LSTM) is to decide what to discard from the state of the cell. This is done by the forgetting gate, which reads the h_{t-1} and x_t and then outputs a value between 0 and 1 after sigmoid activation function, each of the numbers in the cell state is multiplied by this number so that the purpose of selectively discarding information from the cell state is achieved. The second step is to determine what information can be stored in the cell state, including two parts here. The first is the sigmoid entry gate that determines what value we are going to update. The second is to use the hyperbolic tangent transfer function to create a new Candidate value $\widetilde{C}_t$. The process is to confirm the updated information. The third step is to update the old cell state, that is, C_{t-1} is updated to C_t, first of all, the old state C_{t-1} is multiplied by the output of the forgotten gate f_t, discarding the unwanted information, then add the new candidate value $i_t * \widetilde{C}_t$. The final step is to determine what value to output, first through a sigmoid layer (output gate) to determine which parts of the cell state will be output. Then, the cell state is processed by hyperbolic tangent transfer function and multiplied by the output of the output gate o_t to determine which parts are output. In general, ILSTM is updated as shown in formula(1) at time t.

$$\begin{cases} f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\ i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ \widetilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \\ C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t \\ o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\ h_t = o_t * \tanh(C_t) \end{cases} \tag{1}$$

Denoting the actual time series is $X = (x_1, x_2, x_3 ...... x_n)$, the improved LSTM steps are as follows:

Step 1: Expand X to $\begin{bmatrix} x_1, x_2, x_3 \cdots x_{n-k+1} \\ x_2, x_3, x_4 \cdots x_{n-k+2} \\ x_3, x_4, x_5 \cdots x_{n-k+3} \\ \vdots \\ x_k, x_{k+1}, \cdots\cdots x_n \end{bmatrix}$,Where n is the time series length, k is the sample dimension, the number of samples is n-k+1.

The training data is labeled as $y = (x_k, x_{k+1}, \cdots x_n)$, and X is normalized via Equation 2.

$$X = \frac{x_i}{\sqrt{x_i^2 + x_{i+1}^2 + \cdots x_{i+k-1}^2}} (i = 1, 2 \cdots n-k+1) \tag{2}$$

Step 2: Initialize network parameters and set hyper-parameters.

$$\begin{cases} W_f = rand(L, N) \\ b_f = rand(1, N) \\ \vdots \\ Max\_inter = M_1 \\ Error\_Cost = M_2 \end{cases} \tag{3}$$

Where M_1 M_2 represents error threshold Error_Cost and the maximum number of iterations Max_iter respectively, LSTM unit number is L, the number of neurons is N.

Forget gate weight W_f and bias b_f. Similarly, there are input gate, output gate and so on.

Step 3: Calculate what information of the cell state needs to be forgotten.

$$\hat{f}_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) * C_{t-1} \tag{4}$$

Calculate the output of the forgotten gate, and then multiply the forgotten gate's output by the cell state of the previous moment.

Step 4: Calculate which input information can stay in the cell state at time t.

$$\hat{i}_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) * \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{5}$$

This includes two parts, one is the output of the input gate i_t that determines which values we will update. The second is to use the hyperbolic tangent transfer function to create a new candidate vector $\widetilde{C}_t$. Then multiplies to the output of the input gate with the candidate vector.

Step 5: Calculate the cell state C_t

$$C_t = \hat{i}_t + \hat{f}_t \tag{6}$$

The state of the cell is the result of a combination of forgotten and input gates on the state of the cell.

Step 6: Calculate the network output at time t.

$$h_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) * \tanh(C_t) \tag{7}$$

First calculate the output of the output gate o_t .The output of the current moment is obtained by multiplying the output of the output gate by the cell state of the current moment. h_t is the current moment of the predictive value. Repeat steps 3 to 6 to calculate the predicted values for all training samples.

Step 7: Calculate the error between the predicted h and the true values y of all samples.

$$J_{(\theta)}(y, h; W, b) = \frac{1}{2} \| y - h \|^2 \tag{8}$$

If error <Error_Cost or the current number of iterations iter> Max_iter, then quit the training loop. Otherwise use BPTT algorithm to update the network parameters, the number of iterations also plus one and then go to step three until it reaches the error threshold or the maximum number of iterations. The trained network parameters can be calculated as formula (9).

$$\theta_0 = (W_f, W_i, W_C, W_o, C, h, b_f, b_i, b_C, b_o) \qquad (9)$$

Step 8: New sample $X_{n+1}(x_{n-k+2},\cdots x_{n+1})$ with θ_0 Perform the forward operation of LSTM shown in Step 3-Step 6,and obtain the output predicted value of the new sample h_{n+1}.

When next data $X_{n+2}(x_{n-k+3},\cdots x_{n+2})$ is collected x_{n+2} is the true value of predicted value h_{n+1}. Calculate the total error.

$$error = error + \frac{1}{2}(h_{n+1} - x_{n+2})^{\wedge 2} \qquad (10)$$

Then the BP algorithm is used to update the model parameters.

$$\theta_1 = (W_f - \lambda * \Delta W_f, \cdots b_f - \lambda * \Delta b_f) \qquad (11)$$

λ is the learning rate, ΔW_f and Δb_f are the gradient matrices and vectors of the neuron's weights and offsets respectively. Since the parameter initialization is the global optimal solution of the historical sample when adding a new sample, the global optimal solution under the new sample can be achieved with a few loop steps .

Step 9: When the predicted value at the next sample time reaches the point of failure, the system will initiate a fault warning and react promptly to prevent any further loss or even a safety accident.

The ILSTM flowchart is shown in Fig. 4.

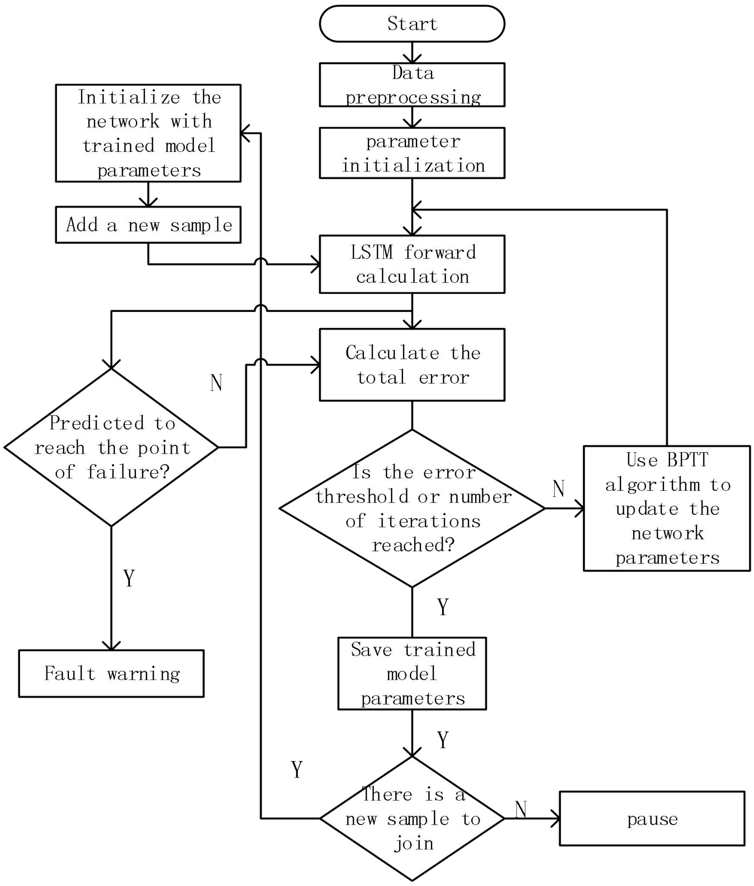


Fig.4 The Training Process of ILSTM

4. EXPERIMENT ANALYSIS

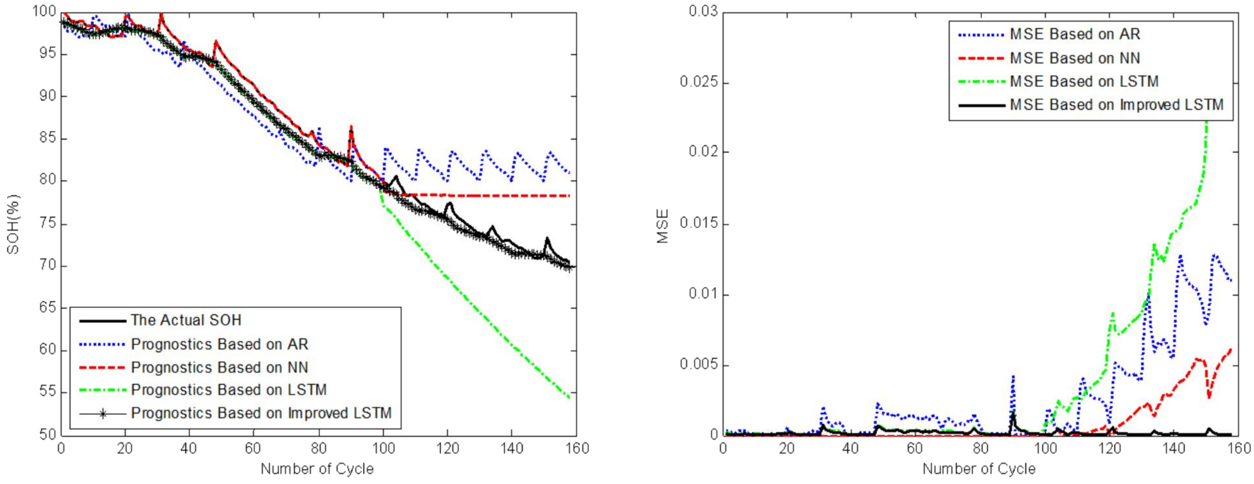
4.1 Experimental Data Description

In this paper, partial ground experimental data of NASA lithium-ion battery is used. NASA lithium-ion battery data is tested in Idaho National Laboratory, the experiment used market sales 18650 lithium-ion battery rated capacity of 2Ah[21-23]. In this paper, B5 lithium battery data is used to experiment. In order to achieve the life expectancy of lithium-ion battery, different Prognostics algorithms and ILSTM algorithm are adopted to predict the cycle of charge-discharge cycle when the capacity of lithium-ion battery declines to 70%. Battery capacity declines to 70% can be considered invalid.

4.2 Analysis of Experimental Results

The experimental results are shown in Figure 5. The solid line is the actual SOH curve, the blue dashed line is the SOH curve predicted by AR, the red dashed line is predicted by the neural network, the green dashed line is predicted by LSTM, and the black dashed line is predicted by the improved LSTM (ILSTM). From figure 5(a), it can be concluded that all the prognostics methods work well with historical data, but the prognostics of several traditional methods can not work well when historical data is not rich enough. Since more online data is used to update the parameters of ILSTM Prognostics model. The proposed method can be a good solution to this situation.

Fig.5(b) illustrates the mean square error(MSE) of the ILSTM. Efficiency comparison of different algorithm is shown in Table 1.



(a) (b)

Fig.5 Battery RUL Prognostics of ILSTM Model

5. CONCLUSION

Aiming at the problem that the traditional LSTM model can not effectively use the non-life-cycle data to establish an excellent RUL Prognostics model and the inability to utilize the online data reasonably, this paper proposes a real-time updating method based on minimizing the error to update model online in the case when small number of sample data is available. Taking NASA lithium-ion battery data as an example, the applicability of the proposed method's efficiency in the field of reliability Prognostics is verified. At the same time, it is proved that this method can be effectively used in lithium ion battery life Prognostics and has high practical value in engineering application.

In the future we will conduct further research to find more effective LSTM parameter optimization methods. In addition, non-numerical data such as image data can also be fused together to make the extracted features more comprehensive and accurate to establish the Prognostics model.

Table 1 Efficiency Comparison of Different Algorithm

The algorithm name	Training time(s)	Predict the point of fault	Mean square error
AR algorithm	4.85543	NAN	0.4017
NN algorithm	12.588388	NAN	0.1237
LSTM algorithm	25.993037	117	0.6479
Improved LSTM algorithm	4.380053	158	0.0144

REFERENCES

[1] Xiao-Sheng Si, Wenbin Wangbde, Dong-Hua Zhouc. Remaining useful life estimation – A review on the statistical data driven approaches[J]. European Journal of Operational Research, 2011, 213(1):1-14.

[2] Nøhr C, Boye N. Towards computer supported clinical activity: A roadmap based on empirical knowledge and some theoretical reflections[J]. Human Social & Organizational Aspects of Health Information Systems, 2008.

[3] Li Y, Shi J, Wang G, et al. A data-driven prognostics approach for RUL based on principle component and instance learning[C]// IEEE International Conference on Prognostics and Health Management. IEEE, 2016:1-7.

[4] Silvio Simani, Cesare Fantuzzi, Ron J. Patton. Model-based Fault Diagnosis in Dynamic Systems Using Identification Techniques[J]. Advances in Industrial Control, 2003:1-54.

[5] Dick van Dijk, Timo Teräsvirta, Philip Hans Franses. Smooth transition autoregressive models - A survey of recent developments[J]. Sse/efi Working Paper, 2002, 21(1):1-47.

[6] Wen-Chuan Wang, Kwok-Wing Chau, Chun-Tian Cheng, et al. A comparison of performance of several artificial intelligence methods for forecasting monthly discharge time series[J]. Journal of Hydrology, 2009, 374(3):294-306.

[7] Shahin M A. A review of artificial intelligence applications in shallow foundations[J]. International Journal of Geotechnical Engineering, 2015, 9(1):49-60.

[8] Hinton G, Deng L, Yu D, et al. Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups[J]. IEEE Signal Processing Magazine, 2012, 29(6):82-97.

[9] Collobert, Ronan, Weston, et al. A unified architecture for natural language processing: deep neural networks with multitask learning[J]. Journal of Parallel and Distributed Computing, 2008.
[10] Tan J, Lu W, An J, et al. Fault diagnosis method study in roller bearing based on wavelet transform and stacked auto-encoder[C]// Control and Decision Conference. IEEE, 2015:4608-4613.
[11] Angeline PJ, Saunders GM, Pollack JB. An evolutionary algorithm that constructs recurrent neural networks.[J]. Neural Networks, IEEE Transactions on, 1994, 5(1):54-65.
[12] Santini S, Bimbo A D, Jain R. Block-structured recurrent neural networks[C]// Neural Networks. 1995:579--596.
[13] Frasconi P, Gori M, Soda G. Recurrent neural networks and prior knowledge for sequence processing: a constrained nondeterministic approach[J]. Knowledge-Based Systems, 1995, 8(6):313-332.
[14] Graves A. Long Short-Term Memory[M]// Supervised Sequence Labelling with Recurrent Neural Networks. Springer Berlin Heidelberg, 2012:1735-1780.
[15] Sepp Hochreiter, Jürgen Schmidhuber. Long Short-Term Memory[J]. Neural Computation, 1997, 9(8):1735-1780.
[16] Tai K S, Socher R, Manning C D. Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks[J]. Computer Science, 2015, 5(1):: 36.
[17] Ge C, Zhu Y, Di Y. Equipment remaining useful life Prognostics oriented symbiotic simulation driven by real-time degradation data[J]. 2017.
[18] Wu Y, Yuan M, Dong S, et al. Remaining Useful Life Estimation of Engineered Systems using vanilla LSTM Neural Networks[J]. Neurocomputing, 2017.
[19] Gers F A, Schmidhuber J, Cummins F. Learning to Forget: Continual Prognostics with LSTM. Neural Computation 12(10): 2451-2471[J]. Neural Computation, 2000, 12(10):2451-2471.
[20] Gers F A, Schmidhuber J, Cummins F. Learning to Forget: Continual Prognostics with LSTM. Neural Computation 12(10): 2451-2471[J]. Neural Computation, 2000, 12(10):2451-2471.
[21] Perrone D, Stefano S D. Survey of lithium-ion battery performance for potential use in NASA missions[C]// Energy Conversion Engineering Conference, 1997. Iecec-97. Proceedings of the, Intersociety. IEEE, 1997:39-41 vol.1.
[22] Erbacher J K, Vukson S P, Marsh R A. Joint DoD/NASA Lithium Ion Battery Development Program[C]// Intersociety Energy Conversion Engineering Conference. 1999.
[23] Liu D, Pang J, Zhou J, et al. Prognostics for state of health estimation of lithium-ion batteries based on combination Gaussian process functional regression[J]. Microelectronics Reliability, 2013, 53(6):832-839.