Journal of Intelligent Manufacturing (2021) 32:37–49
https://doi.org/10.1007/s10845-020-01556-3

# An adaptive update model based on improved Long Short Term Memory for online prediction of vibration signal

Huixin Tian[1,2,5] · Daixu Ren[1,2] · Kun Li[3] · Zhen Zhao[4]

Received: 25 August 2019 / Accepted: 3 March 2020 / Published online: 1 April 2020

**Abstract**
In industrial production, the characteristics of compressor vibration signal change with the production environment and other external factors. Therefore, to ensure the effectiveness of the model, the vibration signal prediction model needs to be updated constantly. Due to the complex structure of Long Short Term Memory (LSTM) network, the LSTM model is difficult to adapt to the scene of online update. Therefore, the update model based on LSTM is difficult to respond quickly to data changes, which affects the accuracy of the model. To solve this problem, the online learning algorithm is introduced into prediction model, Error-LSTM (E-LSTM) model is proposed in this paper. The main idea of E-LSTM model is to improve the accuracy and efficiency of the model according to test error of the model. First, the hidden layer neurons of LSTM network are divided into blocks, and only part of the modules are activated at each time step. The number of modules activated is determined by test error. Thus, the training speed of the model is accelerated and the efficiency of the model is improved. Second, the E-LSTM model can adaptively adjust the training method according to the data distribution characteristics, so as to improve the accuracy of updated model. In experimental part, two types of datasets are used to verify the performance of the proposed model. LSTM model is used for comparative experiments, and the results showed that the updating model based on E-LSTM is better than that based on LSTM in terms of model accuracy and efficiency.

**Keywords** Vibration signal predicting · LSTM network · Test error · Model update · Online learning

## Introduction

In recent years, machine learning technology has been widely used in various fields because it is based on data and does not need to build mechanism models. In the field of machine learning, machine learning algorithm can be divided into online learning algorithm and offline learning algorithm (Ozay et al. 2016). Offline learning is to train the data independently and use the trained model to predict the task. Most of machine learning algorithms studied at present are offline learning algorithm, which have achieved excellent results in many classification and regression tasks (Ozay et al. 2016). However, the following disadvantages of offline machine algorithm limit its application: (1) the training process of the model is very inefficient, and the model training costs a lot of time and space. (2) The training process is not adapting to big data application scenarios. (3) The model cannot adapt to the dynamic changing environment, and the trained model cannot adapt to the change of the external data when the external data changes.

Compared with offline learning, online learning means that the model receives training data in sequence (Mao et al. 2017), and the current model is updated with each sample received. Online learning algorithms do not need all the training data stored in the computer, but can automatically adjust the model itself according to the change of data distribution.

✉ Kun Li
lk_neu@163.com

Huixin Tian
tianhuixin@tiangong.edu.cn

[1] School of Electrical Engineering and Automation, Tiangong University, Tianjin 300387, China

[2] Key Laboratory of Advanced Electrical Engineering and Energy Technology, Tiangong University, Tianjin 300387, China

[3] School of Economic and Management, Tiangong University, Tianjin 300387, China

[4] College of Electronic Information and Automation, Civil Aviation University of China, Tianjin 300300, China

[5] State Key Laboratory of Process Automation in Mining and Metallurgy/Beijing Key Laboratory of Process Automation in Mining and Metallurgy Research, Beijing 100020, China

These advantages make online learning more suitable for processing massive data and timely responding to the dynamic changes in the external environment. Therefore, the study of online learning algorithms is very hot in recent years. Cavalcante et al. (2016) discussed the application of online learning algorithm in the field of computational finance. In the stock trading system, trading data is generated in high frequency and real-time, and users have higher requirements for trading speed. Therefore, the online learning algorithm is very suitable for portfolio research in the financial field to deal with the problem of online portfolio selection. Malaca et al. (2019) applied online inspection system based on machine learning to fabric texture classification in the automotive industry. The application of machine learning online algorithm improves the real-time performance of inspection system, thus increasing the economic benefit. Song et al. (2016) proposed a novel large-scale, context-aware recommender system. The system relies on a novel online learning algorithm which learns online the item preferences of users based on their click behavior. Experimental results show that the proposed algorithm outperforms the state-of-the-art algorithms by over 20% in terms of click through rates. He et al. (2011) proposed a general adaptive online learning framework that is capable of learning from continuous raw data and using such knowledge to improve future learning and prediction performance.

However, online learning algorithms are rarely used in industrial production, especially in equipment fault diagnosis and prediction. In industrial production, as time goes on, the state of equipment maybe changes. At this time, if the previous generated model is adopted, there will be a large error and affect the accuracy of the model. For example, due to aging of compressor components or changes of production environment, the normal operating range of vibration signal changes. At this time, if the previous generated vibration signal prediction model is used, there will be a considerable error or even a state misjudgment (Liu et al. 2019; Carino et al. 2018). Therefore, in this paper, the idea of online learning is introduced into the compressor vibration signal prediction model to ensure the accuracy of the model. Vibration signal prediction have been widely studied in recent years (Fei 2016; Liu and Yang 2018; Wu and Lei 2019), which belongs to time series paradigm (Chen et al. 2006). Long Short Term Memory (LSTM) network (Hochreiter and Schmidhuber 1997) is the most widely used time series prediction model. LSTM network can not only deal with long time dependence in time series data (Gers et al. 2000), but also deal with nonlinear and non-stationary in vibration signal (Tian et al. 2019). Hence, LSTM network is used as the basic prediction model in this paper.

In order to ensure the continuous validity of the model, the online learning algorithm is introduced into the LSTM model. The operational efficiency of the model is an important index of online learning model. However, complex structure and too many parameters of LSTM model lead to high computational cost (Mohamed 2018). Therefore, if dataset is large, problems such as too long training time and not fast enough prediction speed may occur, so that the online update requirements cannot be met in the model update process. In literature (Prabhavalkar et al. 2016), an improved Recurrent neural network (RNN) compression model based on low-rank decomposition and linear projection is proposed to achieve accelerated training, and the loss of precision can be ignored. Tang proposed a parallel improved LSTM neural network to predict Large-Scale Computing Systems Workload (Tang 2019). This network improves the efficiency of the model through the parallel mode and the improved error back propagation method, which makes it perform well in the large-scale workload prediction. Rizk and Awad (2019) proposed a Non-iterative training algorithm to reduce the training time, mainly on feedforward artificial neural networks. The experimental results show that the training speed of this training algorithm is significantly improved compared with that of the back propagation training algorithm. However, the current literatures only consider the training speed of LSTM model in the update process, and does not consider the model accuracy. To solve this problem, online learning algorithm is introduced into LSTM model, and an update model based on Error-LSTM (E-LSTM) is proposed in this paper. The main idea of E-LSTM model is to improve the accuracy and efficiency of the model according to the test error of the model. The contribution of this paper is mainly in three respects.

(1) Online learning algorithm is introduced into vibration signal prediction model to solve the problem of model adaptability when the data distribution changes.
(2) Hidden layer neurons are divided into blocks, and only part of neurons is calculated at each time, so as to solve the efficiency problem of LSTM model in update process.
(3) Test error is used to determine the number of neurons updated at each time. The greater the error, the more neurons will be activated, which ensures the accuracy of the model.

The organization of this paper is as follows. “Methodology” section describes the proposed algorithm and relevant theoretical knowledge in detail. Then, the validity of the proposed algorithm is verified through different experiments and the experimental results are analyzed in detail in “Experiments and analysis” section. Finally, the conclusion of the paper is summarized in “Conclusion” section.

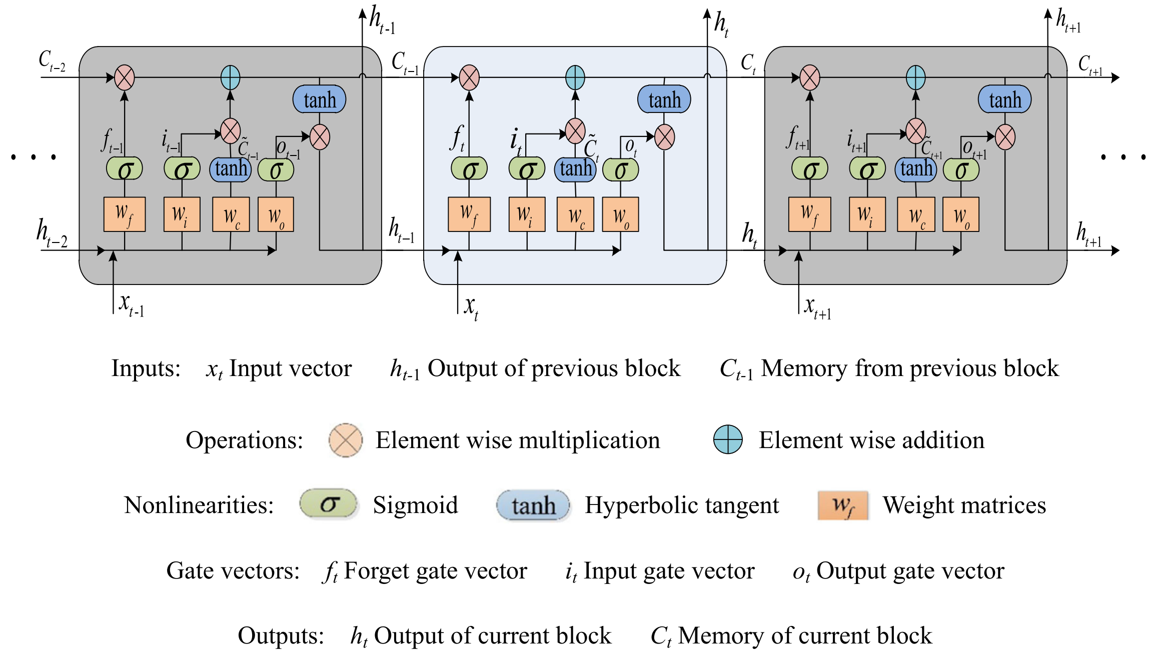

$C_{t-2}$
$h_{t-2}$
$x_{t-1}$
$h_{t-1}$
$C_{t-1}$
$f_{t-1}$
$i_{t-1}$
$\tilde{C}_{t-1}$
$o_{t-1}$
$x_t$
$h_t$
$C_t$
$f_t$
$i_t$
$\tilde{C}_t$
$o_t$
$x_{t+1}$
$h_{t+1}$
$C_{t+1}$
$f_{t+1}$
$i_{t+1}$
$\tilde{C}_{t+1}$
$o_{t+1}$
σ
tanh
$w_f$
$w_i$
$w_c$
$w_o$
Inputs: $x_t$ Input vector $h_{t-1}$ Output of previous block $C_{t-1}$ Memory from previous block
Operations: ⊗ Element wise multiplication ⊕ Element wise addition
Nonlinearities: σ Sigmoid tanh Hyperbolic tangent $w_f$ Weight matrices
Gate vectors: $f_t$ Forget gate vector $i_t$ Input gate vector $o_t$ Output gate vector
Outputs: $h_t$ Output of current block $C_t$ Memory of current block

**Fig. 1** The structure of LSTM network

## Methodology

### Long Short Term Memory (LSTM)

Compressor vibration signals is time series data and has a strong nonlinear and non-stationary (Tian et al. 2019). LSTM network is widely used in dealing with nonlinear time series problems. LSTM network is proposed by Hochreiter and Schmidhuber (1997) as a special kind of the recurrent neural networks (RNNs). LSTM network can alleviate the problem of gradient vanishing encountered by RNN when solving long time dependence tasks (Hochreiter and Schmidhuber 1997; Gers et al. 2000). Therefore, LSTM network is used as the based forecasting model of vibration signal. The structure of LSTM network is shown in Fig. 1, such a memory block consists of a cell, an input gate, an output gate, and a forget gate. The calculation formula of each gate and the update of cell state can be expressed as follows:

$$f_t = \sigma(w_f[h_{t-1}, x_t] + b_f) \tag{1}$$

$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i) \tag{2}$$

$$\tilde{c}_t = \tanh(w_c[h_{t-1}, x_t] + b_c) \tag{3}$$

$$o_t = \sigma(w_o[h_{t-1}, x_t] + b_o) \tag{4}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{5}$$

$$h_t = o_t \odot \tanh(c_t) \tag{6}$$

where $i_t$, $f_t$, $o_t$, $c_t$ and $h_t$ are outputs of input gate, forget gate, output gate, cell and memory block at time step $t$. $x_t$ is input vector of memory block at time step $t$. $h_{t-1}$ is output vector of memory block at time step $t-1$. $\tilde{c}_t$ denotes candidate information of input gate. $\sigma$ and tanh denote activation function. $\odot$ denotes the dot product of the vectors. Additionally, $w_f$, $w_i$, $w_c$ and $w_o$ are the weight to be learned; $b_f$, $b_i$, $b_c$ and $b_o$ are the corresponding bias vectors.

As can be seen from Fig. 1, to achieve better results than RNN model, LSTM model need to sets more parameters. Meanwhile, the LSTM model is slower than the ordinary RNN model in training, which makes it difficult to adapt to the application scenario of online updating (Mohamed 2018; Rizk and Awad 2019). To alleviate the issue, Error-LSTM model is proposed which speeds up the training speed of LSTM model and improves its model accuracy in the updating process.

### Update model based on Error-LSTM

In industrial production, the state of compressor will change with the aging of equipment components, which will lead to changes in the characteristics of vibration signals collected (Ye and Dai 2018). Hence, to ensure the effectiveness of the model, the model needs to be constantly updated. In order to meet the demand of online updating, the prediction model needs to have higher computational efficiency. Due to the complex structure of LSTM model, it is difficult to adapt to the scene of online update. Therefore, E-LSTM model is proposed to improve the efficiency and accuracy

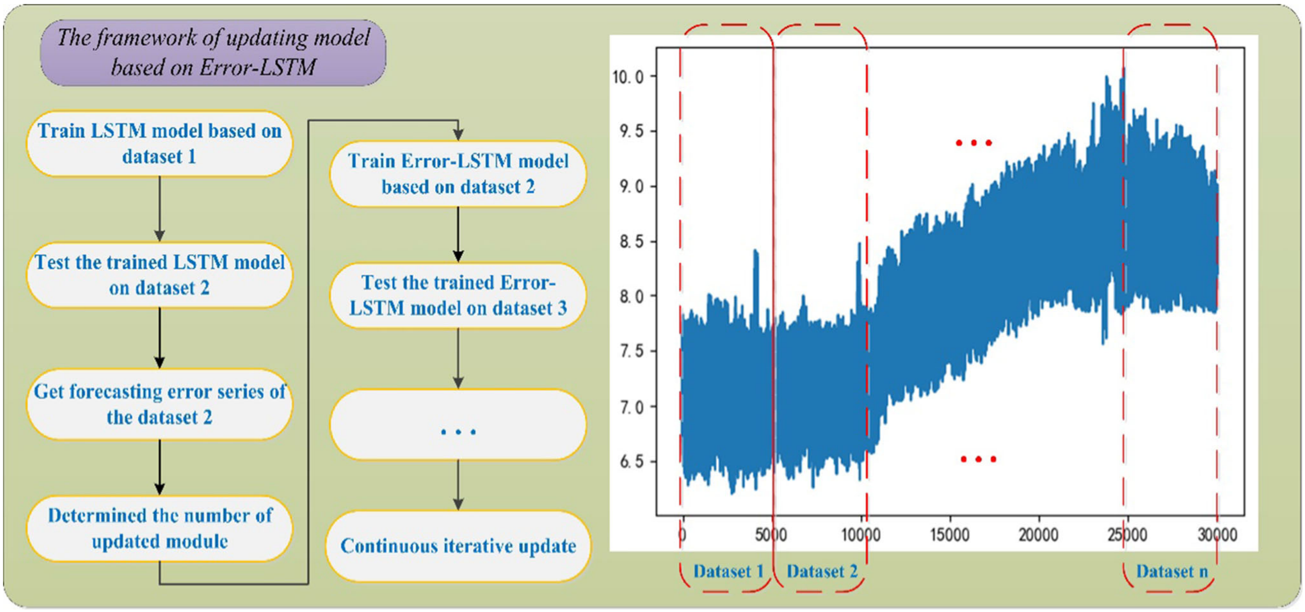

The framework of updating model based on Error-LSTM
Train LSTM model based on dataset 1
Test the trained LSTM model on dataset 2
Get forecasting error series of the dataset 2
Determined the number of updated module
Train Error-LSTM model based on dataset 2
Test the trained Error-LSTM model on dataset 3
. . .
Continuous iterative update
10. 0
9. 5
9. 0
8. 5
8. 0
7. 5
7. 0
6. 5
0
5000
10000
15000
20000
25000
30000
• • •
• • •
Dataset 1
Dataset 2
Dataset n

**Fig. 2** The framework of updating model based on Error-LSTM

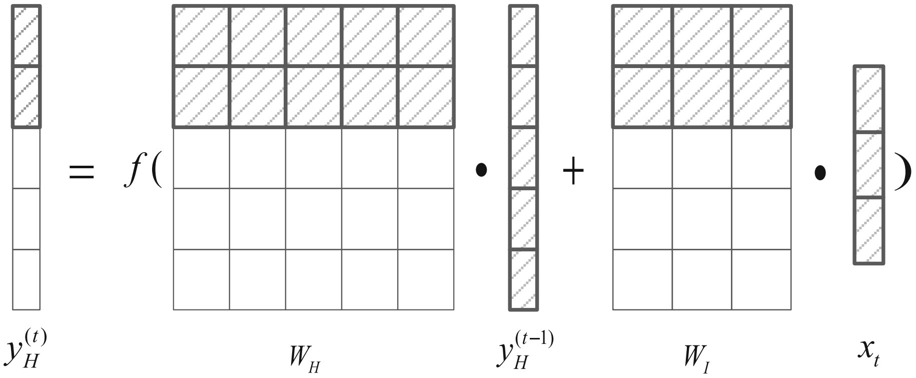


**Fig. 3** Calculation of the hidden layer output gate

of the model. The framework of updating model based on Error-LSTM is demonstrated in Fig. 2. The main idea of this model is simply to improve the performance of the model according to the test error. Next, we will show in detail how E-LSTM improves the efficiency and accuracy of the model.

### Improvement of model efficiency

E-LSTM like LSTM, contains input layer, hidden layer and output layer. There are forward connections from the input to hidden layer, and from the hidden to output layer. However, unlike LSTM, E-LSTM hidden layer neurons are divided into $g$ modules of size $k$ (If the number of hidden layer nodes is $m$, then $m = k * g$). To speed up the training of the model, only part of the module is executed at each time step, and the other modules retain their output values from the previous time step. Suppose two modules are activated at time step $t$, calculation of the hidden layer output gate is illustrated in Fig. 3.

The backward pass of the error propagation is similar to LSTM as well. The only difference is that the error propagates only from modules that were executed at time step $t$. The error of non-activated modules gets copied back in time (similarly to copying the activations of nodes not activated at the time step $t$ during the corresponding forward pass), where it is added to the back-propagated error.

The number of modules activated at each time step is determined by the test error. The basic idea is that if the model generated of the previous time period performs well on the dataset of this time period, it indicates that the dataset of this time period is similar to the dataset of the previous time period. Therefore, when the model is updated during this time period, the number of modules activated can be correspondingly reduced, and vice versa. For clearer expression, let's take the update process from time period $T1$ to $T2$ as an example, the algorithm process is described as follows:

```
Algorithm: The update algorithm based on E-LSTM

Input: dataset1 $data_1$ of time $T1$, dataset2 $data_2$ of time $T2$
       model $M1$ based on $data_1$, number of hidden layer neuron modules: $g$
Output: model $M2$ based on $data_2$
1 :  Calculate the test error series $Err$ of model $M1$ on $data_{T2}$. The length of error series $Err$
     is the same as the length of $data_{T2}$.
2 :  The test error series $Err$ is divided into $n$ error blocks of size $L$. In other words, each error
     block contains $L$ time information.
3 :  Calculate the average error of each error block, namely $Err\_mean$.
4 :  Calculate the maximum value of $Err\_mean$.
     $Emax = \max(Err\_mean)$
5 :  Determine the number of updated modules per error block.
     $Num = \text{int}\ (E\_mean\ /\ Emax * g)$
      The number of updated modules each time step within the error block is the same.
6 :   Determine the hidden layer connection mode, and training the model to get $M2$.
```

It should be noted that since there was no test error in training model $M1$, we used the mode in which all neurons were activated to train the model $M1$. It is important to note that when calculating the number of update modules, we consider the error block series $Err\_mean$ rather than the single error series $Err$. This is because if there are abnormal points in the data, the error of the model at the abnormal point will be large. At this time, if only the error at the current moment is considered, the number of updated modules will be more, which means that the model will focus on training the abnormal points. It is obviously wrong. However, if the average error of $L$ time step is considered (the size of $L$ will be tuned in the experimental section), the influence of abnormal points will be weakened, so as to improve the accuracy of the model.

## Improvement of model accuracy

As we know, when the data distribution changes, it makes more sense to update the model (Razavi-Far et al. 2019). Figure 4 shows the vibration signal data of compressor running for a long time. The data is divided into three sub-datasets, which represent different distribution conditions. If the model is not updated and the model trained in dataset1 is used as the final vibration signal prediction model, the model may perform poorly in dataset2 and dataset3. The update model based on LSTM adopts the incremental learning method to continuously update the model based on new data. However, the update model based on LSTM does not consider the influence of data distribution on model accuracy, which leads to low model accuracy when data distribution changes. Update model based on E-LSTM adopts different update methods for data with different distribution according to test error, so as to improve the accuracy of the updated model.

To explain the improvement in model accuracy of the update model based on E-LSTM, dataset2 is divided into two parts: dataset21 and dataset22, as shown in Fig. 5. It can be seen from Fig. 5 that dataset21 has a similar distribution with dataset1, while dataset22 has a similar distribution with

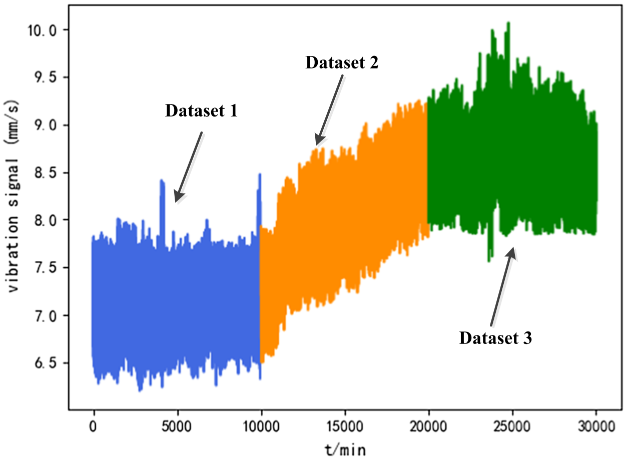


**Fig. 4** Data distribution of the entire dataset

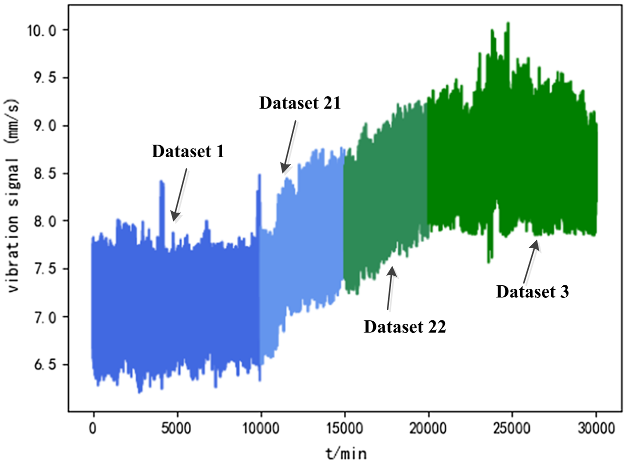


**Fig. 5** Approximate distribution of dataset2

dataset3. The following steps explain why the updated model based on E-LSTM performs better than the updated model based on LSTM.

*Step 1* LSTM model is used to train model $M1$ based on dataset1.
*Step 2* Test model $M1$ is based on dataset21 and dataset22, and test error $E21$ and $E22$ are obtained.
*Step 3* Since dataset21 and dataset1 have similar distribution, theoretically model $M1$ performs better in dataset21, so $E21$ is less than $E22$.
*Step 4* Hence, the number of update modules for dataset22 is greater than for dataset21. In the training process of E-LSTM model, dataset22 is mainly trained and finally model $M2$ is obtained. In addition, LSTM model is used to train model $M2'$ based on dataset2.
*Step 5* Model $M2$ perform better on dataset3 than model $M2'$ for two reasons: (1) Dataset22 and dataset3 have similar distribution, and model $M2$ focuses on training dataset22. (2) More importantly, model $M2$ pays little attention to dataset21 which is different from dataset3, which reduces the interference of data with different distribution to the model. Therefore, the update model based on E-LSTM can quickly adapt to the changes of data, so as to avoid large errors.

In "Methodology" section, the proposed model and related theories are introduced in detail. First, because of the superiority of LSTM model in dealing with time series, LSTM model is adopted as the basic model to forecasting vibration signal. Second, due to the LSTM model is not suitable for online update scenarios, E-SLTM model is proposed to improve the performance of model. Finally, it is proved theoretically why E-LSTM can improve the accuracy and efficiency of the model. In next section, experiments will be conducted to verify effectiveness of the proposed model.

## Experiments and analysis

In this section, two different datasets are used to verify the superiority of the proposed model. First, we construct a set of test function with changing tendency to verify the validity of the model. In addition, to verify the application of the proposed model in industrial production, the vibration signal of the compressor is used to test the model. Experimental data were collected from a reciprocating compressor on the oil production platform in Bohai, China. The speed sensor is fixed on the wall of the main motor of the compressor, and the sampling interval is 1 min. Three commonly used evaluation indices are employed to evaluate the performance of the E-LSTM model. They are mean absolute error (MAE), root mean square error (RMSE) and mean absolute percentage error (MAPE) that defined as follow:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|predicted_i - observed_i| \tag{7}$$

$$RMSE = \sqrt{\frac{1}{\mathrm{N}}\sum_{i=1}^{N}(predicted_i - observed_i)^2} \tag{8}$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{predicted_i - observed_i}{observed_i}\right| \times 100\% \tag{9}$$

where $predicted_i$ and $observed_i$ denote the predicted value and observed value of the $i$ sample respectively. $N$ denotes sample size. Besides, in order to clearer verify the superiority of the proposed model, Promoting Percentages of the MAE (PMAE), Promoting Percentages of the RMSE (PRMSE), Promoting Percentages of the MAPE (PMAPE) and Promoting percentages of the Time (PTime) are executed in this paper. The definitions of these evaluation indexes are demonstrated as follows:

$$P_{MAE} = (MAE_1 - MAE_2)/MAE_1 \tag{10}$$

$$P_{RMSE} = (RMSE_1 - RMSE_2)/RMSE_1 \tag{11}$$

$$P_{MAPE} = (MAPE_1 - MAPE_2)/MAPE_1 \tag{12}$$

$$P_{Time} = (Time_1 - Time_2)/Time_1 \tag{13}$$

It should be noted that all experiments run in the Python 3.6 environment on 2.80 GHz PC with process i5-7440HQ and 16G RAM. Considering the effect of randomness on performance, all experiments in this paper are run independently for 10 times, and then averaged to get the final result.

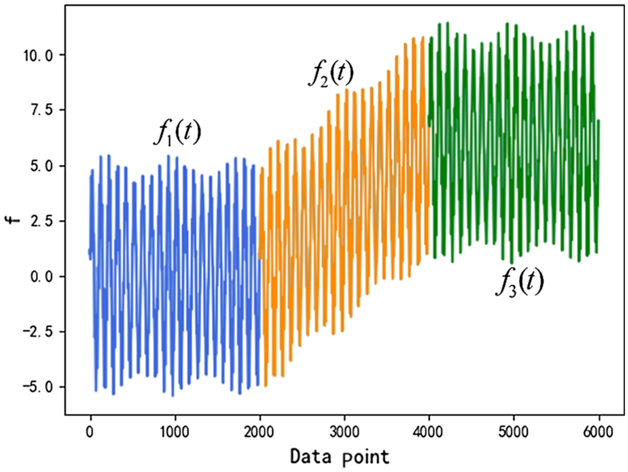


**Fig. 6** The dataset of test function

## Test function dataset

In this section, we generated a set of test function dataset with a trend change to test the proposed model. The test function is defined as follow:

$$h_1(t) = 4\sin 40\pi t \tag{14}$$

$$h_2(t) = (1 + 0.5\sin 5\pi t)\cos(250\pi t + 20\pi t^2) \tag{15}$$

$$f_1(t) = h_1(t) + h_2(t) \quad t \in [0, 1] \tag{16}$$

$$f_2(t) = h_1(t) + h_2(t) + 6t \quad t \in [0, 1] \tag{17}$$

$$f_3(t) = h_1(t) + h_2(t) + 6 \quad t \in [0, 1] \tag{18}$$

Each function of $f_1(t)$, $f_2(t)$ and $f_3(t)$ was sampled 2000 times, and then the three functions were spliced together to get the final test function, as shown in Fig. 6. We first tune hyper parameters of model and then compare it with other forecasting models.

### Hyper parameter tuning

There are many hyper parameters in E-LSTM model, which will greatly affect the prediction performance of the model. Therefore, to ensure the accuracy of the model, we first fine-tune the parameters. Since the trial and error method has the advantages of wide parameter search range and fast search speed (Elsayed et al. 2015), we adopt the trial and error method for parameter tuning. The main parameters that affect the accuracy of model include the number of hidden layers, the number of hidden layer neurons, the window size and the size of error blocks. For these parameters, different values are set based on some prior knowledge, as shown in Table 1.

**Table 1** Setting of experimental parameters

| Parameters | Parameter values | | | | |
|---|---|---|---|---|---|
| Number of hidden layers | 1 | 2 | 5 | 10 | 20 |
| Number of neurons | 20 | 60 | 100 | 200 | – |
| Window size | 500 | 1000 | 2000 | – | – |
| Error block size | 5 | 10 | 20 | 50 | – |

A. The number of hidden layers

The number of hidden layers of neural network is an important parameter that affects the performance of the model. Increasing the number of hidden layers can improve the accuracy of the model but also increase the complexity of the model. In this experiment, to verify the effect of different number of hidden layers on the model performance, the number of hidden layers is set as 1, 2, 5, 10 and 20. The other three key parameters number of neurons, window size and error block size are 60, 500 and 5. The experiment results are shown in Table 2.

As can be seen from Table 2, increasing the number of hidden layers (2, 5 layers) cannot significantly improve the accuracy of the model, and the calculation time of the model increases significantly. If the number of hidden layers continues to increase to 10 and 20 layers, the accuracy of the model declines instead of improving, because the complexity of the model is so high that the overfitting problem arises. At the same time, in the process of model update, the efficiency of the model is an important index, so the number of hidden layers is set as 1.

B. The number of hidden layer neurons

The number of hidden layer neurons is an important parameters of neural network model. Too few neurons will cause the model to fail to fully learn all the features in data. However, the excessive number of neurons not only reduce the efficiency of model, but also may lead to overfitting problem (Henriquez and Ruz 2018). In this experiment, to verify the effect of different number of neurons on the model performance, the number of neurons is set as 20, 60, 100 and 200. The other three key parameters number of hidden layers, window size and error block size are 1, 500 and 5. The experiment results are shown in Table 3, where the best performance is highlighted in bold.

It can be seen from Table 3 that the model performs best when the number of hidden layer neurons is 60. As the number of neurons continues to increase to 100 or even 200, the performance of model declines instead of getting better, which indicates that the model may cause overfitting prob-

**Table 2** The effect of number of hidden layers on the model performance

| Number of hidden layers | MAE (mm/s) | RMSE (mm/s) | MAPE (%) | Time (s) |
|---|---|---|---|---|
| 1 | 0.1197 | 0.1456 | 8.6992 | 6.3530 |
| 2 | 0.0862 | 0.1146 | 11.622 | 12.014 |
| 5 | 0.0762 | 0.1066 | 9.3621 | 26.222 |
| 10 | 2.6153 | 2.9754 | 233.48 | 60.361 |
| 20 | 2.6153 | 2.9753 | 233.24 | 129.35 |

**Table 3** The effect of number of neurons on the model performance

| Number of neurons | MAE (mm/s) | RMSE (mm/s) | MAPE (%) |
|---|---|---|---|
| 20 | 0.3189 | 0.1656 | 9.6612 |
| 60 | **0.1409** | **0.0350** | **2.6313** |
| 100 | 0.1963 | 0.0640 | 3.3735 |
| 200 | 0.2023 | 0.0709 | 4.5591 |

**Table 4** The effect of window size on the model performance

| Window size | MAE (mm/s) | RMSE (mm/s) | MAPE (%) |
|---|---|---|---|
| 500 | 0.2380 | 0.0921 | 4.9900 |
| 1000 | 0.0455 | 0.0042 | 2.4811 |
| 2000 | 0.0588 | 0.0070 | 11.3571 |

lem. Therefore, the number of hidden layer neurons is set to 60 in subsequent experiments.

### C. The window size

In the process of model updating, the proper of selection of window size is significant to the improvement of model accuracy. The window size is too small, resulting in insufficient sample size and insufficient model training. On the contrary, too large window size leads to information redundancy during model training, which affects the accuracy of the model (Youn et al. 2018). In this experiment, to evaluate the effect of window size on the model performance, the window size is set as 500, 1000 and 2000. The number of hidden layers is 1, the number of neurons is 60 and the error block size is still 5. The experiment results are shown in Table 4.

It can be seen from Table 4 that the model error is the smallest when the window size is 1000. Hence, a window size of 1000 is an ideal value for this dataset.

### D. The size of error blocks

Similarly, to evaluate the effect of error blocks size on the model performance, the error blocks size is set as 5, 10, 20 and 50. The number of hidden layers is 1, the number of neurons is 60 and the window size is 1000. The experiment results are shown in Table 5.

**Table 5** The effect of error blocks size on the model performance

| Error blocks size | MAE (mm/s) | RMSE (mm/s) | MAPE (%) |
|---|---|---|---|
| 5 | 0.0693 | 0.0088 | 1.6250 |
| 10 | 0.0658 | 0.0079 | 4.4054 |
| 20 | 0.0630 | 0.0080 | 3.0983 |
| 50 | 0.0680 | 0.0082 | 4.0015 |

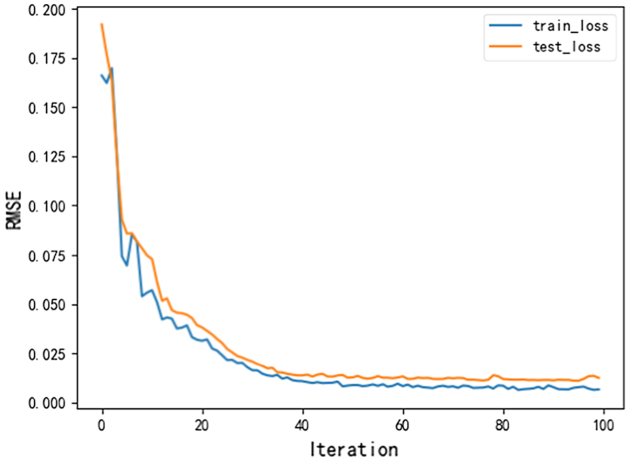
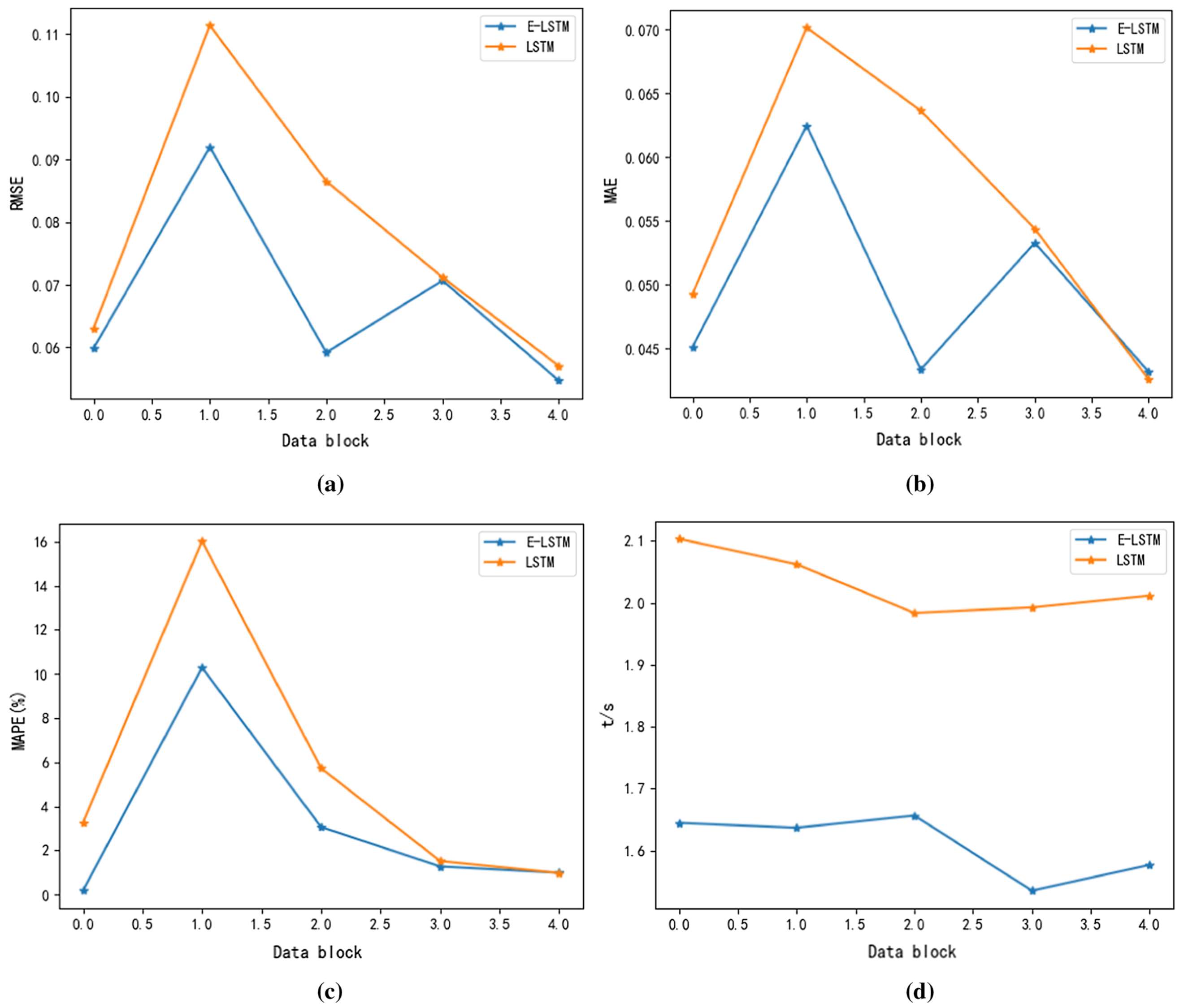


**Fig. 7** Learning curve for E-LSTM model

**Table 6** The parameters of models

| Parameters | Values |
|---|---|
| Number of hidden neurons | 60 |
| Number of modules (for E-LSTM) | 4 |
| Window size | 1000 |
| Error blocks size (for E-LSTM) | 20 |
| Batch size | 256 |
| Learning rate | 0.01 |
| Iteration | 100 |
| Optimizer | *adam* |

As can be seen from Table 5, the model performs best when error block size is 20. In the other words, information at current time step is most relevant to information at previous 20 times step. As the error block size continues to increase to 50, the performance of model declines, which indicates that the current time step information will be interfered if the associated information is too long.



(a)

(b)

(c)

(d)

**Fig. 8** The forecasting results of different models on test function dataset **a** RMSE; **b** MAE; **c** MAPE; **d** time

**Table 7** The promoting percentages of different models on test function dataset

| Data block | $P_{RMSE}$ (%) | $P_{MAE}$ (%) | $P_{MAPE}$ (%) | $P_{Time}$ (%) |
|---|---|---|---|---|
| 0 | 5.0612 | 8.4556 | 94.3037 | 21.8112 |
| 1 | 17.4217 | 10.9704 | 35.7535 | 20.6479 |
| 2 | 31.5793 | 31.9168 | 46.5334 | 16.4906 |
| 3 | 0.7201 | 2.0238 | 15.9967 | 22.9710 |
| 4 | 4.1023 | – 1.4028 | – 1.2051 | 21.6143 |
| Average (%) | 11.7769 | 10.3928 | 38.2765 | 20.7071 |

Based on the above experimental results, after tuning the key parameters of model, the number of hidden layers is 1, the number of neurons is 60, the window size is 1000 and the error blocks size is 20. It should be pointed out that for a new dataset, it is necessary to use the trial and error method to search for parameters again. In the following section, we will verify the validity of E-LSTM model based on these optimized parameters.

### Checking overfitting for the model

In order to prevent the model may fit well in the training dataset, but cannot be generalized to new examples (Bouktif et al. 2018). It is necessary to plot learning curve that shows the model performance on training and testing data. RMSE for both the training and testing datasets for optimized E-LSTM model (i.e., after hyper parameter optimization) decrease with the increase of iteration times and converge at similar values, which shows that our model is not overfitting, as shown in Fig. 7.

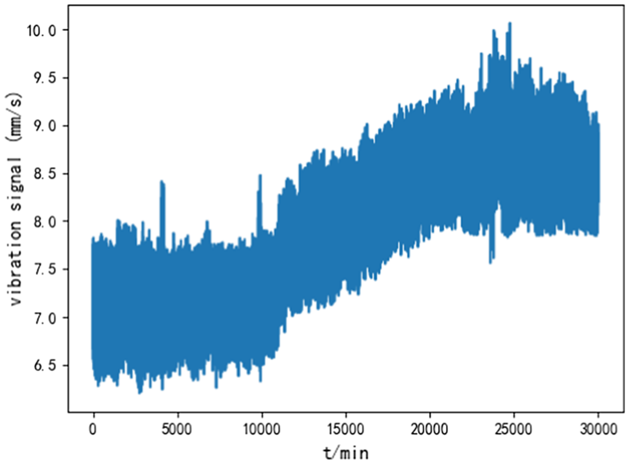


**Fig. 9** The dataset of vibration signal

## Comparison and analysis

In this experiment, we will verify the superiority of E-LSTM model in algorithm efficiency and model accuracy. LSTM model is used for comparison. For a fair comparison, the parameters of models are the same as shown in Table 6.

Since the dataset has 6000 data and the window size is 1000, the dataset can be divided into 6 data blocks. The purpose of the experiment is to compare the test error and the training time of different models on each data block. Considering that the first data block has no test error, we only study the performance of the model on the remaining 5 data blocks (labeled 0–4). The forecasting result of different models is shown in Fig. 8 and Table 7. From Fig. 8 and Table 7, the following conclusions can be drawn:

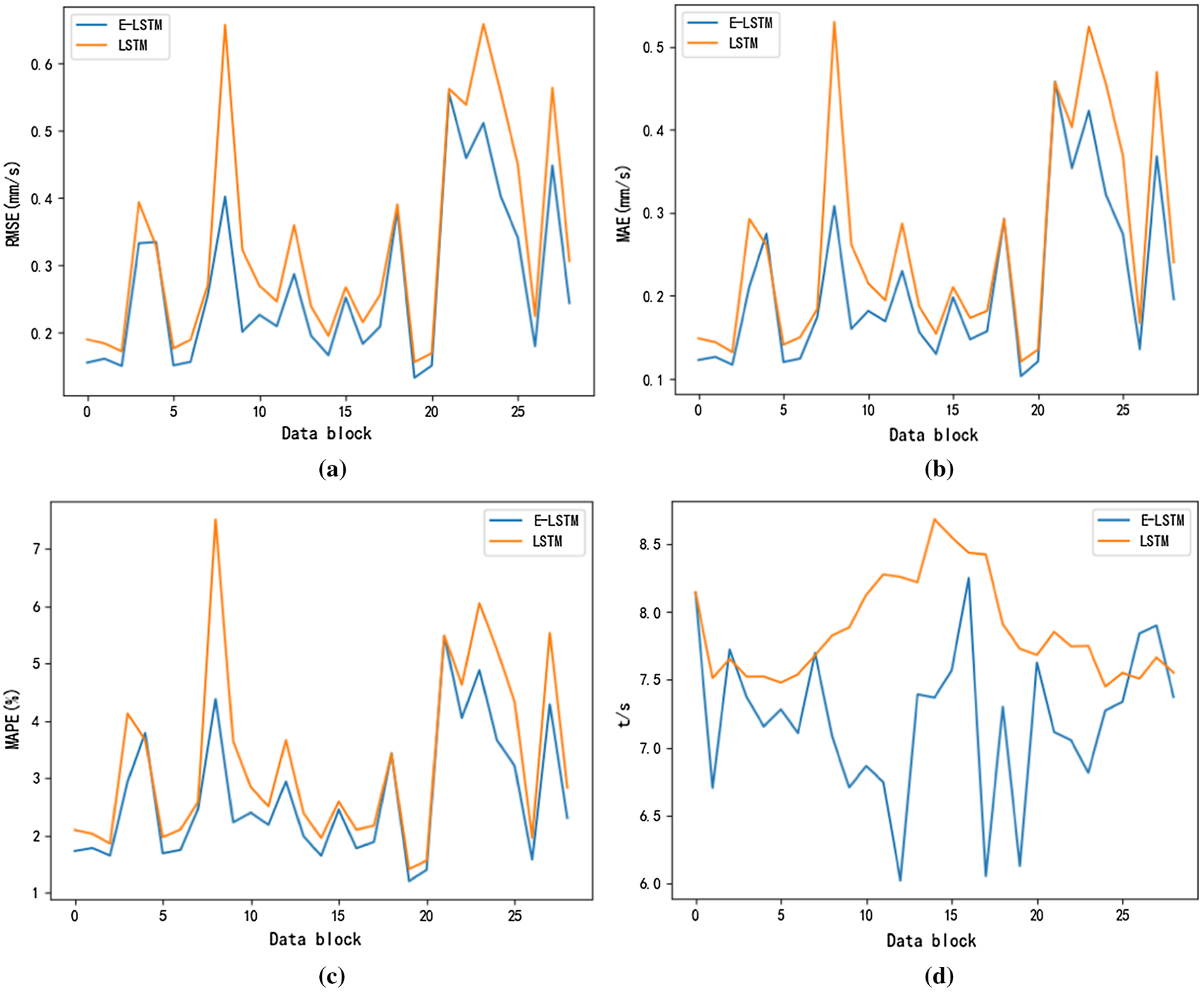


**Fig. 10** The forecasting results of different models on vibration signal dataset **a** RMSE; **b** MAE; **c** MAPE; **d** time

(1) It can be seen from Fig. 8a–c that three error indicators RMSE, MAE and MAPE have similar trends. In almost all data blocks, E-LSTM model has smaller errors than LSTM model. When the data distribution changes, E-LSTM model can quickly adapt to this change, so as not to produce large errors. However, LSTM model does not consider the influence of data distribution, which leads to large errors. From Table 7, the RMSE, MAE, MAPE average promoting percentages of the LSTM model by the E-LSTM model are 11.78%, 10.39%, 38.28%, respectively.

(2) It can be seen from Fig. 8d that E-LSTM model has a faster training speed than LSTM model.
From Table 7, the Time average promoting percentages of the LSTM model by the E-LSTM model is 20.71%.

(3) Compared with LSTM model, E-LSTM model has obvious improvements in both model accuracy and efficiency, which indicates that the proposed model is effective.

The data block corresponds to the data block in Fig. 8.

## Vibration signal dataset

To further verify the application of the proposed model in actual production scenarios. In this section, vibration signal generated by compressor is used to test the performance of the model. In order to better verify the performance of the model, the dataset with obvious change are selected. As is shown in Fig. 9. The dataset contains 30,000 data and sampling interval is 1 min.

Similarly, the four parameters the number of hidden layers, the number of hidden layer neurons, window size and error block size are respectively tuned. Due to limited space, the tuning process is in “Appendix”. Finally, the number of hidden layers is 1, the number of hidden layer neurons is 60, the window size is 1000 and the error blocks size is 20. We verify the performance of the E-LSTM model by comparing it with the LSTM model. Similarly, considering that the first data block has no test error, we only study the performance of the model on the remaining 29 data blocks (labeled 0–28). The forecasting result of different models is shown in Fig. 10 and Table 8.

The following conclusions can be drawn from Fig. 10 and Table 8:

(1) It can be seen from Fig. 10a–c that three error indicators RMSE, MAE and MAPE have similar trends. In almost all data blocks, E-LSTM model has smaller errors than LSTM model. From Table 8, the RMSE, MAE, MAPE average promoting percentages of the LSTM model by the E-LSTM model are 16.26%, 16.12%, 16.25%, respectively.

**Table 8** The promoting percentages of different models on vibration signal dataset

| Data block | $P_{\text{RMSE}}$ (%) | $P_{\text{MAE}}$ (%) | $P_{\text{MAPE}}$ (%) | $P_{\text{Time}}$ (%) |
|---|---|---|---|---|
| 0 | 18.1022 | 17.4560 | 17.5061 | 0.0414 |
| 1 | 12.3281 | 12.2781 | 12.1838 | 10.7870 |
| 2 | 12.5256 | 11.4455 | 11.2222 | – 0.8997 |
| 3 | 15.4784 | 27.8617 | 28.7010 | 2.0308 |
| 4 | – 1.8485 | – 5.1545 | – 3.3587 | 4.9045 |
| 5 | 14.2124 | 14.7331 | 14.2960 | 2.6140 |
| 6 | 17.3441 | 17.1429 | 16.9138 | 5.7607 |
| 7 | 5.5278 | 5.2044 | 4.8024 | – 0.2152 |
| 8 | 38.8382 | 41.8928 | 41.7395 | 9.4970 |
| 9 | 37.5743 | 38.9434 | 38.8093 | 14.9506 |
| 10 | 16.0256 | 15.4096 | 15.6611 | 15.5261 |
| 11 | 14.8479 | 12.9633 | 12.8547 | 18.4984 |
| 11 | 20.2022 | 19.9564 | 19.6563 | 27.1014 |
| 13 | 18.1322 | 16.6123 | 16.8345 | 10.0356 |
| 14 | 14.9080 | 15.7561 | 15.9037 | 15.1329 |
| 15 | 5.7633 | 5.8031 | 5.4720 | 11.4853 |
| 16 | 14.9865 | 14.9789 | 15.3314 | 2.1941 |
| 17 | 18.2557 | 13.5779 | 13.1227 | 28.1403 |
| 18 | 1.8295 | 0.1456 | – 0.1497 | 7.6955 |
| 19 | 14.8284 | 14.5797 | 14.7582 | 20.7034 |
| 20 | 10.7783 | 10.5079 | 10.4542 | 0.7270 |
| 21 | 1.2924 | – 0.1711 | 0.2090 | 9.3865 |
| 22 | 14.7075 | 12.2206 | 12.5293 | 8.9462 |
| 23 | 22.3074 | 19.3715 | 19.3192 | 12.0720 |
| 24 | 27.6731 | 29.4214 | 30.1427 | 2.3563 |
| 25 | 24.1114 | 25.5974 | 25.8030 | 2.8431 |
| 26 | 19.9788 | 18.9470 | 19.4344 | – 4.4197 |
| 27 | 20.4637 | 21.6379 | 22.4896 | – 3.0815 |
| 28 | 20.3948 | 18.4627 | 18.6411 | 2.3811 |
| Average (%) | 16.2611 | 16.1235 | 16.2512 | 8.1792 |

(2) It can be seen from Fig. 10d that E-LSTM model has a faster training speed than LSTM model in most data blocks. In some data blocks with large test error, duo to the large number of updated modules, the training time of E-LSTM model is similar to LSTM model. However, for whole dataset, E-LSTM model still has advantages over LSTM model in model efficiency. From Table 8, the Time average promoting percentages of the LSTM model by the E-LSTM model is 8.18%.

In this section, we use two types of datasets to verify the performance of the model. Experimental results show that E-LSTM model has better performance than LSTM model in terms of model efficiency and accuracy. Therefore, E-LSTM model can ensure that the accuracy of the model is not affected when the data distribution changes. Furthermore, E-

LSTM model provides a new solution for industrial big data mining.

## Conclusion

In industrial big data, the equipment will accumulate a large amount of data in a certain period of time. Meanwhile, with the change of time, the distribution of data will change, resulting in that the previously generated model does not meet the current moment. Therefore, the model needs to be constantly updated to meet the current data distribution. To solve the problem, the online learning algorithm is introduced into the prediction model, a new vibration signal prediction model Error-LSTM (E-LSTM) is proposed in this paper. Based on LSTM model, E-LSTM model improves the accuracy and efficiency of the model. First, the hidden layer neurons are divided into blocks, and only part of the modules are counted at each moment. Thus, the training speed of the model is accelerated and the efficiency of the model is improved. Second, for the different data distribution in dataset, E-LSTM model can adopt different training methods according to the test error. In other words, E-LSTM model can focus on training data with similar distribution, and reduce the interference of data with different distribution, so as to improve the accuracy of the model.

In the experimental part, we used two different datasets to test the model. LSTM model is compared with E-LSTM model. In order to better demonstrate the superiority of the E-LSTM model, the improvement effect of the model is quantified. In summary, compared with LSTM model, E-LSTM model has advantages in both model accuracy and efficiency, which indicates that the proposed model is effective.

In conclusion, this paper is proposed against the background that the accuracy of vibration signal prediction model decreases when the data distribution changes. Therefore, the application scenario of E-LSTM model is to predict the time series data containing the change of data distribution. However, if the time series data runs smoothly and the data distribution does not change, the accuracy of E-LSTM model is not necessarily higher than that of LSTM model, but the efficiency is still higher than that of LSTM model. In addition, the research object of this paper is the field of industrial big data. In the future, the idea of E-LSTM model can also be applied to other fields, such as financial big data and medical big data.

**Acknowledgements** This work was supported by the National Natural Science Foundation of China under Grant (61703406 and 71602143), Tianjin Natural Science Foundation (18JCYBJC22000), Tianjin Science and Technology Correspondent Project (18JCTPJC62600 and 19JCTPJC47600), Tianjin high school innovation team training Program (TD13-5038), State Key Laboratory of Process Automation in Mining and Metallurgy/Beijing Key Laboratory of Process Automation in Mining and Metallurgy Research Fund Project (BGRIMM-KZSKL-2019-08).

## Appendix

See Tables 9, 10, 11 and 12.

**Table 9** The effect of number of hidden layers on the model performance

| Number of hidden layers | MAE (mm/s) | RMSE (mm/s) | MAPE (%) | Time (s) |
|---|---|---|---|---|
| 1 | 0.2079 | 0.2667 | 2.6185 | 122.2584 |
| 2 | 0.2178 | 0.2805 | 2.7377 | 196.7627 |
| 5 | 0.2360 | 0.3041 | 2.9516 | 446.821 |
| 10 | 0.2574 | 0.3215 | 3.2813 | 942.013 |
| 20 | 0.3285 | 0.3882 | 4.1932 | 2105.93 |

**Table 10** The effect of number of neurons on the model performance

| Number of neurons | MAE (mm/s) | RMSE (mm/s) | MAPE (%) |
|---|---|---|---|
| 20 | 0.2923 | 0.1440 | 3.6839 |
| 60 | 0.2672 | 0.1255 | 3.3413 |
| 100 | 0.2659 | 0.1276 | 3.3670 |
| 200 | 0.2743 | 0.1314 | 3.4556 |

**Table 11** The effect of window size on the model performance

| Window size | MAE (mm/s) | RMSE (mm/s) | MAPE (%) |
|---|---|---|---|
| 500 | 0.2849 | 0.1196 | 3.6441 |
| 1000 | 0.2361 | 0.1080 | 2.9761 |
| 2000 | 0.2770 | 0.1337 | 3.4890 |
| 5000 | 0.3254 | 0.1697 | 4.1020 |

**Table 12** The effect of error blocks size on the model performance

| Error blocks size | MAE (mm/s) | RMSE (mm/s) | MAPE (%) |
|---|---|---|---|
| 5 | 0.2787 | 0.1356 | 3.5088 |
| 10 | 0.2750 | 0.1326 | 3.4603 |
| 20 | 0.2721 | 0.1288 | 3.4239 |
| 50 | 0.2751 | 0.1323 | 3.4615 |

# References

Bouktif, S., Fiaz, A., Ouni, A., & Serhani, M. A. (2018). Optimal deep learning LSTM model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches. *Energies, 11*(7), 1–20.

Carino, J. A., Delgado-Prieto, M., Iglesias, J. A., Sanchis, A., Zurita, D., Millan, M., et al. (2018). Fault detection and identification methodology under an incremental learning framework applied to industrial machinery. *IEEE Access, 6,* 49755–49766.

Cavalcante, R. C., Brasileiro, R. C., Souza, V. L. P., Nobrega, J. P., & Oliveira, A. L. I. (2016). Computational intelligence and financial markets: A survey and future directions. *Expert Systems with Applications, 55,* 194–211.

Chen, Z. S., Yang, Y. A., Hu, Z., & Shen, G. J. (2006). Detecting and predicting early faults of complex rotating machinery based on cyclostationary time series model. *Journal of Vibration and Acoustics-Transactions of the Asme, 128*(5), 666–671.

Elsayed, S. M., Sarker, R. A., & Essam, D. L. (2015). Training and testing a self-adaptive multi-operator evolutionary algorithm for constrained optimization. *Applied Soft Computing, 26,* 515–522.

Fei, S. W. (2016). Kurtosis prediction of bearing vibration signal based on wavelet packet transform and Cauchy kernel relevance vector regression algorithm. *Advances in Mechanical Engineering, 8*(9), 1–7.

Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. *Neural Computation, 12*(10), 2451–2471.

He, H. B., Chen, S., Li, K., & Xu, X. (2011). Incremental learning from stream data. *IEEE Transactions on Neural Networks, 22*(12), 1901–1914.

Henriquez, P. A., & Ruz, G. A. (2018). A non-iterative method for pruning hidden neurons in neural networks with random weights. *Applied Soft Computing, 70,* 1109–1121.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780.

Liu, J. T., & Yang, X. X. (2018). Learning to see the vibration: A neural network for vibration frequency prediction. *Sensors, 18*(8), 1–14.

Liu, Y., Duan, L. X., Yuan, Z., Wang, N., & Zhao, J. P. (2019). An intelligent fault diagnosis method for reciprocating compressors based on LMD and SDAE. *Sensors, 19*(5), 1–19.

Malaca, P., Rocha, L. F., Gomes, D., Silva, J., & Veiga, G. (2019). Online inspection system based on machine learning techniques: Real case study of fabric textures classification for the automotive industry. *Journal of Intelligent Manufacturing, 30*(1), 351–361.

Mao, W. T., He, L., Yan, Y. J., & Wang, J. W. (2017). Online sequential prediction of bearings imbalanced fault diagnosis by extreme learning machine. *Mechanical Systems and Signal Processing, 83,* 450–473.

Mohamed, M. (2018). Parsimonious memory unit for recurrent neural networks with application to natural language processing. *Neurocomputing, 314,* 48–64.

Ozay, M., Esnaola, I., Vural, F. T. Y., Kulkarni, S. R., & Poor, H. V. (2016). Machine learning methods for attack detection in the smart grid. *IEEE Transactions on Neural Networks and Learning Systems, 27*(8), 1773–1786.

Prabhavalkar, R., Alsharif, O., Bruguier, A., & McGraw, I. (2016). On the compression of recurrent neural networks with an application to LVCSR acoustic modeling for embedded speech recognition. In *IEEE international conference on acoustics, speech and signal processing* (pp. 5970–5974).

Razavi-Far, R., Hallaji, E., Saif, M., & Ditzler, G. (2019). A novelty detector and extreme verification latency model for nonstationary environments. *IEEE Transactions on Industrial Electronics, 66*(1), 561–570.

Rizk, Y., & Awad, M. (2019). On extreme learning machines in sequential and time series prediction: A non-iterative and approximate training algorithm for recurrent neural networks. *Neurocomputing, 325,* 1–19.

Song, L. Q., Tekin, C., & van der Schaar, M. (2016). Online learning in large-scale contextual recommender systems. *IEEE Transactions on Services Computing, 9*(3), 433–445.

Tang, X. Y. (2019). Large-scale computing systems workload prediction using parallel improved LSTM neural network. *IEEE Access, 7,* 40525–40533.

Tian, H. X., Ren, D. X., & Li, K. (2019). A hybrid vibration signal prediction model using autocorrelation local characteristic-scale decomposition and improved long short term memory. *IEEE Access, 7,* 60995–61007.

Wu, T. Y., & Lei, K. W. (2019). Prediction of surface roughness in milling process using vibration signal analysis and artificial neural network. *The International Journal of Advanced Manufacturing Technology, 102*(1), 305–314.

Ye, R., & Dai, Q. (2018). A novel transfer learning framework for time series forecasting. *Knowledge-Based Systems, 156,* 74–99.

Youn, J., Shim, J., & Lee, S. G. (2018). Efficient data stream clustering with sliding windows based on locality-sensitive hashing. *IEEE Access, 6,* 63757–63776.