

# Milling Tool Wear Prediction Based on 1DCNN-LSTM



Wanliang Xia, Jin Zhou, Wenju Jia, and Miaoxian Guo

**Abstract** The real-time monitoring of tool wear is critical to ensure the high-quality machining. However, there are some problems in machining condition monitoring, such as large amount of data, redundancy of machining information and insufficient prediction accuracy, which affect the reliability of the tool wear monitoring process. This paper presents tool wear monitoring with multi-sensor instrument and tool wear compensation in milling process. Through the optimal feature combination of fusion signals, a tool wear prediction model based on 1DCNN-LSTM is constructed. Compared with the tool wear prediction model without feature selection, the prediction accuracy and calculation efficiency of 1DCNN-LSTM model are significantly improved.

**Keywords** Milling · Tool wear · Prediction · 1DCNN-LSTM

## 1 Introduction

Tool condition is considered to be one of the most important factors in determining production quality, productivity and energy consumption [1]. As a widely used tool wear prediction technique, machine learning algorithm can establish a nonlinear mapping relationship between features and tool wear. Guan et al. [2] performed empirical modal decomposition of acoustic emission signals, and the feature vector composed of the autoregressive model of each modal function is extracted. Suo et al. [3] uses the multi-resolution wavelet method to analyze the milling force signal, extracts the energy and covariance of component into the BP neural network to achieve tool wear prediction.

Dang et al. [4] collected vibration signals during the machining process, automatically extracted features based on 1DCNN (one-dimensional convolutional neural

---

W. Xia · J. Zhou · W. Jia · M. Guo (✉)

School of Mechanical Engineering, University of Shanghai for Science and Technology, Shanghai, China

e-mail: [guomx@usst.edu.cn](mailto:guomx@usst.edu.cn)

network), and used extreme learning machine to predict tool wear. Zhou et al. [5] used Hilbert-Huang transform to extract tool wear features and predict tool wear and tool remaining life based on LSTM (long short term memory) for tool wear and remaining tool life. Zhang et al. [6] carried out wavelet packet transform on milling force signals and used the energy in different frequency bands as feature vectors to predict tool wear state by using sparse auto-encoding network model. Although the above single type of deep learning models can effectively extract spatial and has good prediction effect, it cannot mine information from both spatial and temporal dimensions.

Yan et al. [7] proposed a long short-term memory convolutional neural network (LSTM-CNN) model, which uses LSTM and CNN to extract features of vibration and milling force signals from both sequence and multi-dimensional aspects. The mapping relationship between features and tool wear improves the prediction accuracy of tool wear. An et al. [8] proposed a hybrid model of CNN-SBULSTM (CNN with Stacked Bi-directional and Uni-directional LSTM), using CNN to extract features from internal from internal controller and external sensor signals, which performs features dimensionality reduction. Li et al. [9] proposed a 1DCNN-LSTM hybrid model which made full use of the learning ability of 1DCNN and the time series analysis ability of LSTM to fully mine the information related to tool wear state in vibration and acoustic emission signals, and achieved good tool wear recognition results.

In this paper, the process signals are collected based on the multi-sensor fusion system, and the time domain, frequency domain and time–frequency domain features are extracted from the original signals to reduce the influence of noise. The adaptive moment estimation algorithm is used to optimize the 1DCNN-LSTM model, and the information related to tool wear is mined from both temporal and spatial dimensions in the tool wear feature data set to improve the accuracy of tool wear prediction.

## 2 Theoretical Basis of 1DCNN-LSTM Model

The 1DCNN-LSTM network model structure is shown in Fig. 1. In the prediction model, the useful information in the feature sequence of the input layer is fully excavated by two convolutional layers and two pooling layers alternately, and then the time series information of the output features of pooling layer 2 is extracted by LSTM. Finally the tool wear prediction value is obtained by the fully connected layer.

In order to improve the prediction accuracy of the model and the convergence speed of the iterative solution, each feature column and the actual wear value in the optimal feature combination are normalized:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (1)$$

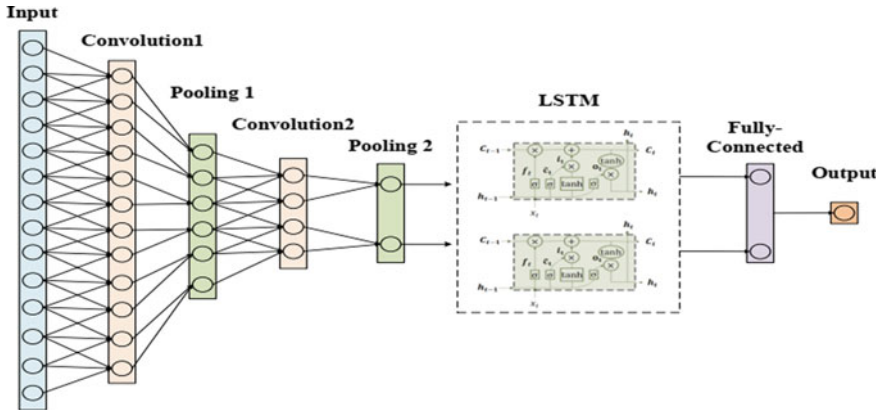


Fig. 1 1DCNN-LSTM network structure

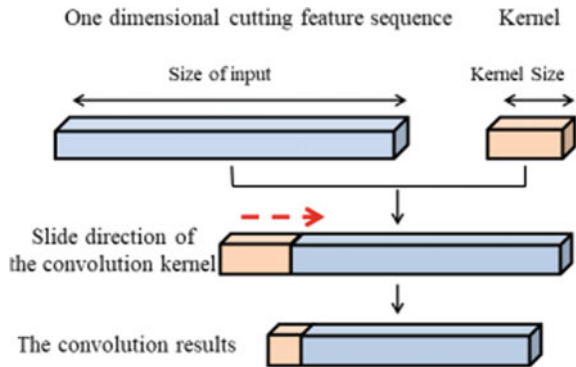
The standardized features are input into the convolutional layer of 1DCNN for deep feature extraction. The convolution process is shown in Fig. 2. Taking a one-dimensional cutting feature sequence of length  $N$  as an example, with a convolution kernel size of  $T \times 1$  and a step size of  $S$ , the resultant length  $G$  after convolution is calculated as:

$$G = \frac{N - T}{s} + 1 \tag{2}$$

In the process of convolution kernel sliding, the convolution kernel with the input data to obtain the feature results, and the calculation formula is shown below:

$$X_j^l = f \left( \sum_{i=1}^{M^{l-1}} (X_i^{l-1} * K_{ij}^l) + b_j^l \right) \quad i = 1, 2, \dots, M^l \tag{3}$$

Fig. 2 Convolution process



where  $X_j^l$  is the  $j$ th feature map of the  $l$ th layer,  $M^l$  is the input array for computing the  $j$ th output,  $X_i^{l-1}$  is the  $i$ th input in the  $l-1$ th layer,  $K_{ij}^l$  is the convolution kernel of the  $l$ th layer,  $b_j^l$  is the bias of the  $j$ th of the  $l$ th layer, and  $f(\cdot)$  is the activation function.

To avoid overfitting caused by overabundance neurons, maximum pooling is added after convolution to retain important feature information and improve training efficiency. The specific calculation formula is as follows:

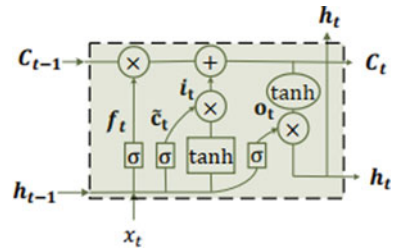
$$P(j) = \max_{t \in K_j} (q(t)) \tag{4}$$

where  $P(j)$  is the  $j$ th feature value after pooling,  $K_j$  is the  $j$ th pooling domain, and  $q(t)$  is the element value of the convolutional feature in the  $j$ th pooling domain before pooling.

The output of the 1DCNN is fed into the LSTM neural network for modeling and extracting temporal information. The LSTM neural network introduces a series gate mechanism based on RNN (recurrent neural network) to obtain long-term memory and alleviate the gradient disappearance and explosion problems, and its cell structure is shown in Fig. 3.

LSTM selectively processes information by combining current information and cell state history information using ingates, forgetting gates and output gates. At each moment  $t$ , the ingates and forgetting gates combine the output value  $h_{t-1}$  of the previous moment and the input  $x_t$  of the current moment to obtain the input coefficient  $i_t$ , the forgetting coefficient  $f_t$ , and the candidate cell state  $\tilde{C}_t$  after the activation function. At each moment  $t$ , the ingates and forgetting gates combine the output value  $h_{t-1}$  of the previous moment and the input  $x_t$  of the current moment to obtain the input coefficient  $i_t$ , the forgetting coefficient  $f_t$ , and the candidate cell state  $\tilde{C}_t$  after the activation function. The cell state  $C_t$  at the current moment is obtained by combining the information filtered by  $f_t$  oblivion from the cell state  $C_{t-1}$  at the previous moment and the information filtered by  $i_t$  from the candidate cell state  $\tilde{C}_t$ . The output coefficients  $o_t$  calculated by  $h_{t-1}$  and  $x_t$  through the activation function after the cell state update. The updated cell state  $C_t$  is multiplied with  $o_t$  after the activation function to obtain the predicted value of the current moment  $h_t$ .

**Fig. 3** Structure of LSTM memory cell unit



The formulae for each threshold, internal memory cell, memory and candidate state are shown below:

Ingate (Threshold):

$$i_t = \sigma(w_i \cdot [h_{t-1}, x_t] + b_i) \quad (5)$$

Oblivion Gate (Threshold):

$$f_t = \sigma(w_f \cdot [h_{t-1}, x_t] + b_f) \quad (6)$$

Output Gate (Threshold):

$$o_t = \sigma(w_o \cdot [h_{t-1}, x_t] + b_o) \quad (7)$$

Internal memory unit (long-term memory):

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (8)$$

Predicted values (short-term memory):

$$h_t = o_t * \tanh(C_t) \quad (9)$$

Candidate state (new knowledge inducted):

$$\tilde{C}_t = \tanh((w_c \cdot [h_{t-1}, x_t] + b_c)) \quad (10)$$

where:  $w_i, w_f, w_o, w_c$  are the parameter matrices to be trained;  $b_i, b_f, b_o, b_c$  are the bias terms to be trained;  $\sigma$  the sigmoid activation function with the output interval  $[0, 1]$ ;  $\tanh$  is the activation function with the output interval  $[-1, 1]$ .

The output of the LSTM is used as the input of the fully connected layer, and the predicted value is obtained after the fully connected layer, which realizes the mapping of features to tool wear values. The calculation formula is as follows.

$$x^l = \sigma(w^l x^{l-1} + b^l) \quad (11)$$

where  $w_l$  is the weight of the  $l$ th layer and  $b_l$  is the bias of the  $l$ th layer.

Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) are used as the two evaluation indicators of model prediction accuracy, which are calculated as follows.

RMSE:

$$P_{rmse} = \sqrt{\frac{1}{n} \sum_{k=1}^n (y_k^{pre} - y_k)^2} \quad (12)$$

MAE:

$$P_{mae} = \frac{1}{n} \sum_{k=1}^n |y_k^{pre} - y_k| \quad (13)$$

where  $y_k^{pre}$  is the predicted value of milling cutter wear,  $y^k$  is the true value of milling cutter wear, and  $n$  is the number of samples.

### 3 Milling Tool Wear Prediction Based on 1DCNN-LSTM

The workpiece is fixed on the dynamometer through the fixture, and the vibration sensor is installed on the side of the workpiece to facilitate accurate acquisition of vibration and force signals. In the whole process of tool wear, there are 300 times of cutting, and finally 300 sets of milling force and vibration signal data are obtained. During the cutting process, the time domain, frequency domain and other characteristic data extracted from the sensor signal are usually time series data which have obvious spatial local and time dependence characteristics [10]. CNN has strong data mining capabilities, and can achieve good prediction results even with less preprocessing. However, it assumes that all inputs and outputs are independent, and correlations between features at different moments are ignored, leads to performance degradation in processing time series data. LSTM can effectively deal with the long-term dependence problem, which exactly makes up for the deficiency of CNN in dealing with time series data [11].

1DCNN is mainly used to deal with one-dimensional time series data, so 1DCNN is chosen to process the input feature sequence in this paper. Based on the local feature extraction ability of 1DCNN and the time dependence of LSTM, this paper designs a tool wear prediction model based on 1DCNN-LSTM. Taking the selected optimal feature combination sequence and the measured actual tool wear value as the input of the model, the nonlinear relationship between the feature and tool wear is established by training the model, and the tool wear prediction is finally realized.

The total number of layers in the network of the model is 8, mainly containing 2 layers of convolution, 2 layers of pooling, 1 layer of LSTM and 1 layer of fully connected layers. The input layer is the optimal feature combination sequence  $A_k$ ,  $A_k = \{A_{k1}, A_{k2}, \dots, A_{kd}\}$ ,  $k = 1, 2, \dots, 300$ , denoting the number of tool walks, and  $d$  is the number of feature dimensions in the optimal feature combination. In order not to miss feature information during convolution, the step size is set to 1. The size of the convolution kernel determines the weight distribution in the convolution process, and  $3 \times 1$  convolution kernel is chosen, and the number of filters chosen in this paper is 16. The pooling domain size is chosen from the commonly used  $2 \times 1$ , where the step size is 2. The number of hidden layer neurons in LSTM is the same as the dimension of feature vectors in the input model.

Train the entire model with Categorical-Cross entropy Loss, and the cross entropy error is calculated as follows:

$$Loss(\theta) = - \sum_{i=1}^n y_i^{pre} \log y_i \quad (14)$$

where  $y_i^{pre}$  is the predicted value of milling cutter wear,  $y^j$  is the true value of milling cutter wear,  $\theta$  is the network parameter, and  $n$  is the number of samples.

The algorithm performs the minimum solution along the direction where  $Loss(\theta)$  decreases the fastest and achieves a fit between the predicted and true values after several iterations. Since the traditional gradient descent algorithm is easy to fall into the local optimal and the update speed is slow when solving the model, the ADAM algorithm is used to optimize the model.

ADAM uses the first-order and second-order moment means of the gradient to perform adaptive learning rate calculation and parameter update, respectively, ensuring that the learning rate of each parameter is dynamically adjusted within a determined range, making the parameter changes are relatively stable. The model calculates the error of each network layer by back propagation to achieve an accurate finding of the optimal solution for each network layer parameter  $\theta$ . Let  $\varepsilon$  be the step size and take the value of 0.001; the moment estimation index decay rates  $\rho_1$  and  $\rho_2$  are usually set to 0.9 and 0.999, respectively. The constant  $\delta = 10^{-8}$  is introduced to make the calculated value stable with error limit  $e = 10^{-8}$ . The computational flow of the ADAM algorithm is shown in Table 1.

**Table 1** Flowchart of Adam's algorithm

---

ADAM algorithm:

---

- 1: Initialization parameters: initialize the training parameter  $\theta_0$  when  $t = 0$ ; initialize the first-order  $s_0 \leftarrow 0$  and the second-order moment variable  $r_0 \leftarrow 0$
  - 2: While  $\theta(t)$  does not reach convergence do (start iteration)
  - 3:  $t = t + 1$  (time step  $t$  increases)
  - 4:  $g_t = \nabla_{\theta} Loss(\theta_{t-1})$  (to obtain the  $t$ th round objective function gradient)
  - 5:  $s_t \leftarrow \rho_1 \cdot s_{t-1} + (1 - \rho_1) \cdot g_t$  (Update the biased first-order moment vector)
  - 6:  $r_t \leftarrow \rho_2 \cdot r_{t-1} + (1 - \rho_2) \cdot g_t \odot g_t$  (Update the biased second-order moment vector)
  - 7:  $\hat{s}_t \leftarrow s_t / (1 - \rho_1^t)$  (Correction of deviation of first-order moments)
  - 8:  $\hat{r}_t \leftarrow r_t / (1 - \rho_2^t)$  (Correction of deviation of first-order moments)
  - 9: if  $\|\hat{s}_t / (\sqrt{\hat{r}_t} + \delta)\|_2 \geq e$  do: (Determine whether the conditions are met)
  - 10:  $\theta_t = \theta_{t-1} + \Delta\theta$  (Update parameters)
  - 11: else: (End of judgment)
  - 12: end while (Iteration stop)
-

## 4 Analysis of Different Tool Wear Models

In order to verify and compare the prediction effect of the 1DCNN-LSTM tool wear prediction model proposed in this paper, the other two methods are used for comparison:

- (a) LSTM model with the optimal feature combination sequence as input, denoted as LSTM;
- (b) 1DCNN-LSTM model with three-way force and vibration raw signal as input (Raw Data 1DCNN-LSTM, denoted as RD-1DCNN-LSTM).

### 4.1 Comparison of Model Stability

The LSTM model and 1DCNN-LSTM model were established in Python 3.8.2 and Tensorflow 2.0.1 frameworks, and the same training parameters were set for the three model methods: the optimizer for the training model was Adam, the basic learning rate was 0.001, the batch size was 30, and the number of iterations was 1000. The models were fully trained with the feature data, and then the stability of the model training process was verified by the feature data sets of Experiments 1 and 2, and the variation of the loss function values of the three models RD-1DCNN-LSTM, LSTM, and 1DCNN-LSTM are shown in Figs. 4, 5 and 6.

It can be seen from three figures, it reach a small value of loss function within 1000 iteration cycles. The loss on the training set is very small from the loss on the test set, both floating around 0.58. Therefore, the training process of the models is normal and the training parameters are reasonable. Among them, the RD-1DCNN-LSTM model tends to be stable after the iteration period reaches 600, and the LSTM model tends to be stable after the iteration period reaches 300. However, the model proposed in this paper is in a stable state only after 20 iterations, indicating that the proposed model is easy to train and can obtain high efficiency. The change trends of the loss function values in the two Experiment are similar, indicating the universality of the model.

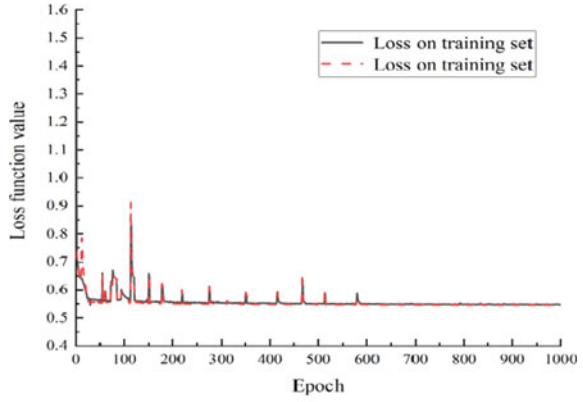
### 4.2 Comparison of Model Prediction Effects

The tool wear prediction effects of the three models, RD-1DCNN-LSTM, LSTM and 1DCNN-LSTM are shown in Figs. 7, 8 and 9.

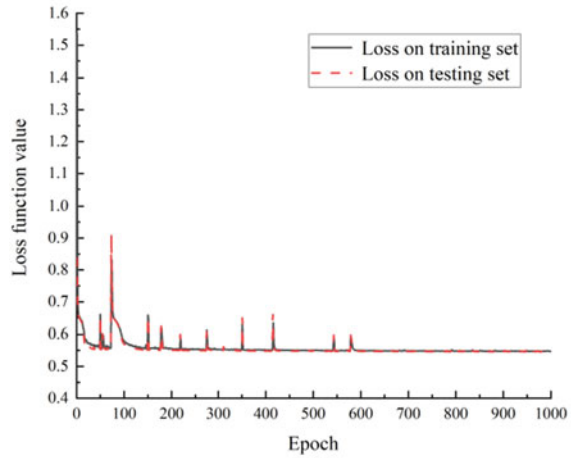
From Figs. 7, 8 and 9, it can be seen that the RD-1DCNN-LSTM model has the worst prediction effect among the three models. It is because that the interference of various factors such as environment makes the original data set contains a large amount of invalid redundant information, and the multi-domain features contain less interference factors. The prediction effect of the 1DCNN-LSTM model is better than



**Fig. 4** Variation of loss function values for RD-1DCNN-LSTM model: **a**  $a_p = 0.1$  mm, **b**  $a_p = 0.2$  mm



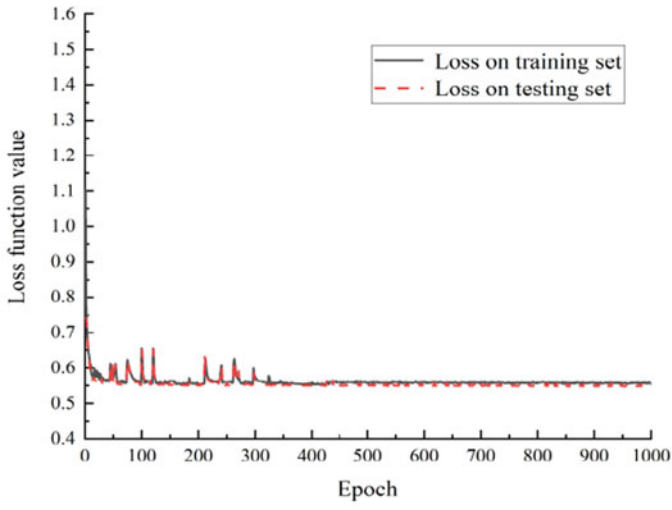
(a)



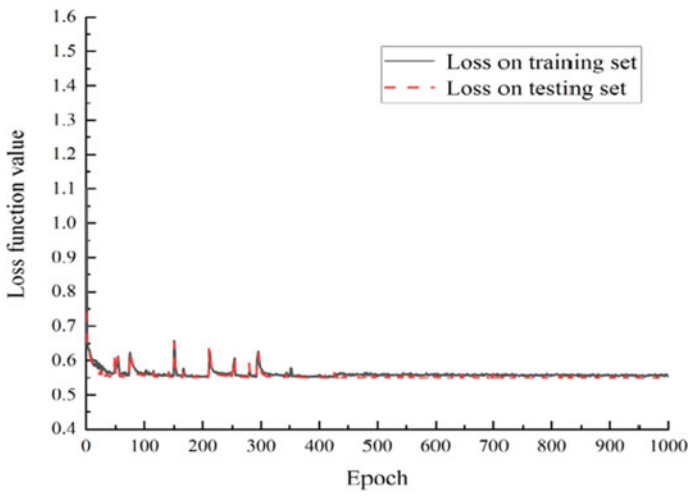
(b)

that of the single LSTM model with smaller error values, indicating that the proposed model has stronger learning ability. In order to better prove the effectiveness of the proposed method, the evaluation criteria MAE and RMSE and the time cost of model operation are calculated as shown in Table 2.

As shown in the Table, compared with the LSTM and RD-1DCNN-LSTM models, the MAE of the 1DCNN-LSTM model was reduced by 7.0 and 11.4, with a decrease rate of 52.9% and 65.0% (experiments No. 1 and 2 mean), RMSE decreased by 10.1 and 15.3, with a decrease rate of 54.9% and 65.3% (experiments No. 1 and 2 average), and the runtimes decreased by 2839.4 and 1565.9 s, with a decreased rate of 75.4% and 62.8% (experiments No. 1 and 2 mean), indicating that the proposed model can predict tool wear more effectively. Therefore, the validity and feasibility of the model can be shown from the time cost and evaluation criteria.

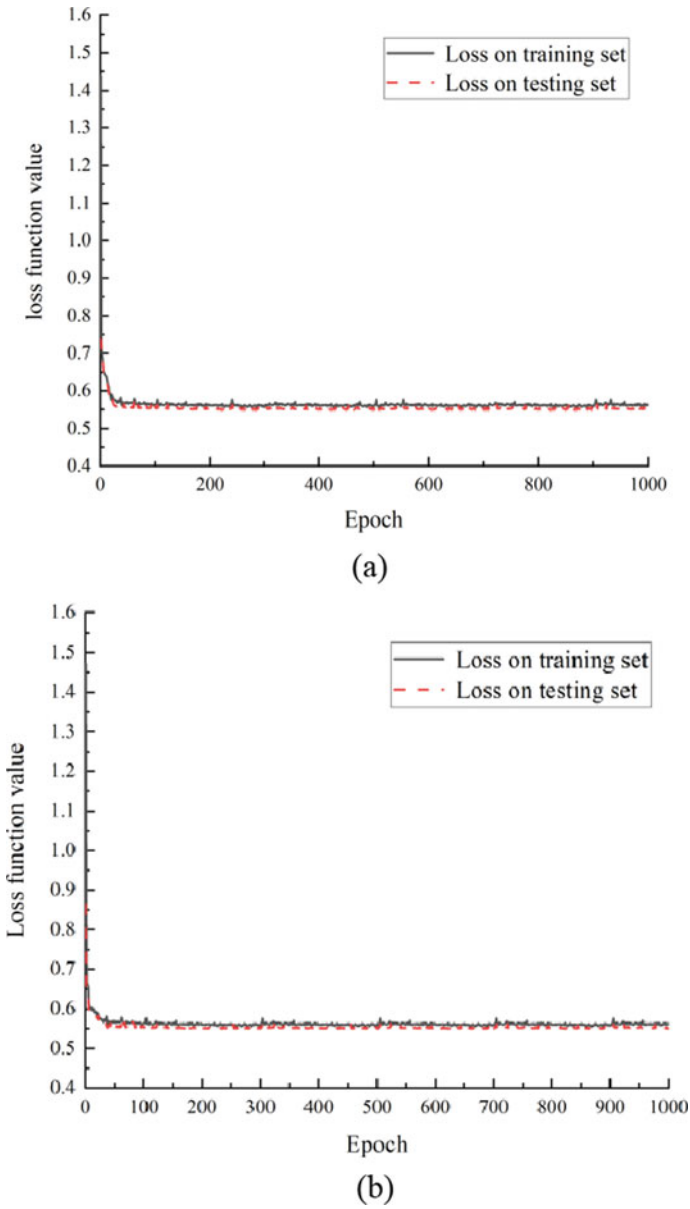


(a)



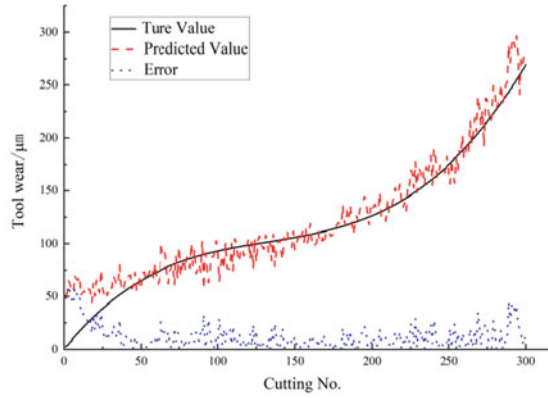
(b)

**Fig. 5** Variation of loss function values for LSTM model: **a**  $a_p = 0.1$  mm, **b**  $a_p = 0.2$  mm

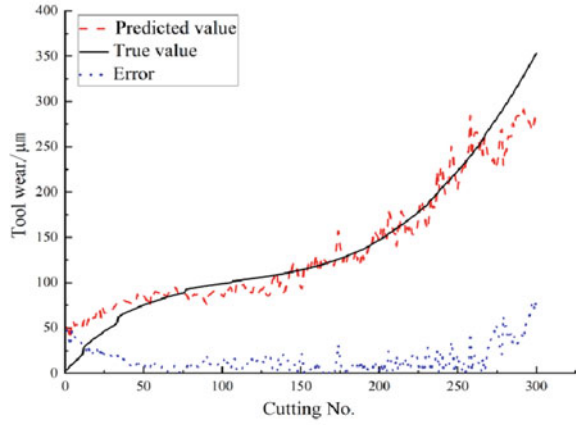


**Fig. 6** Variation of loss function values for 1DCNN-LSTM model: **a**  $ap = 0.1$  mm, **b**  $ap = 0.2$  mm

**Fig. 7** Prediction effect of RD-1DCNN-LSTM model:  
**a**  $a_p = 0.1$  mm, **b**  $a_p = 0.2$  mm

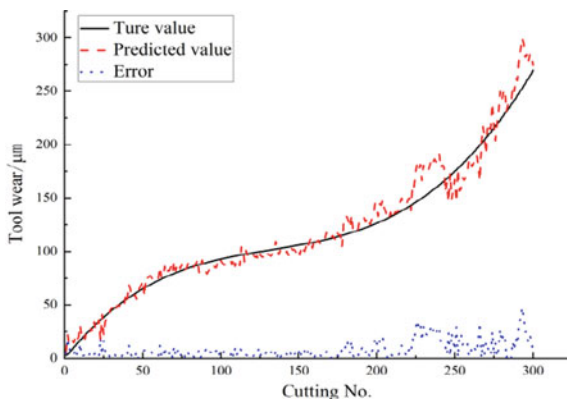


(a)

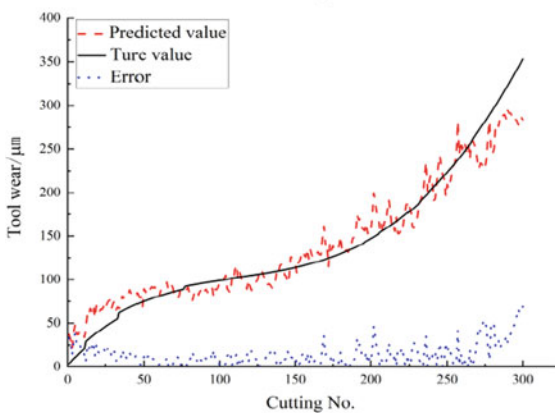


(b)

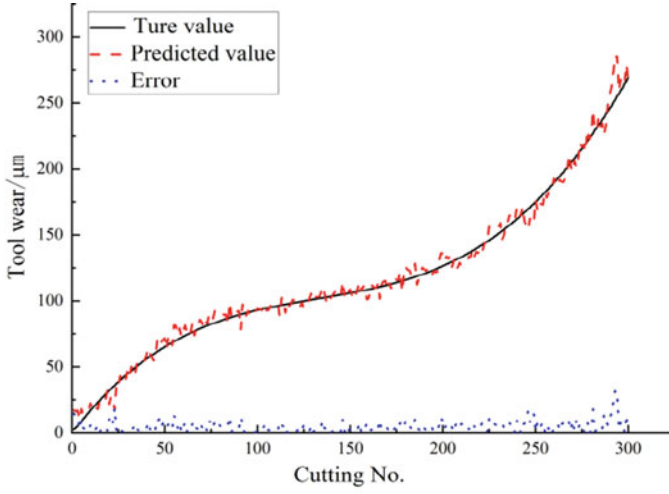
**Fig. 8** Prediction effect of LSTM mode: **a**  $a_p = 0.1$  mm, **b**  $a_p = 0.2$  mm



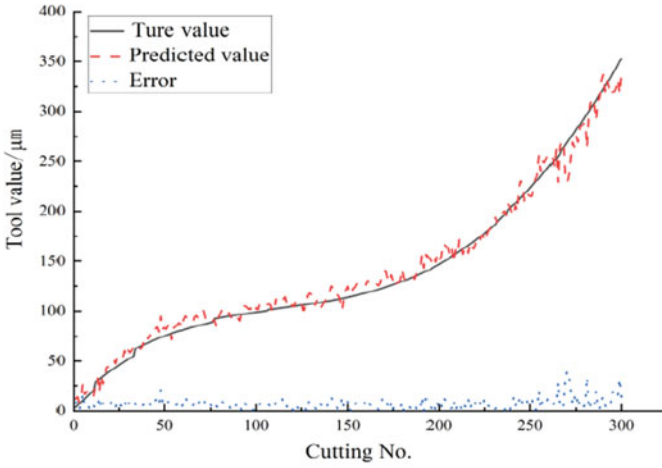
(a)



(b)



(a)



(b)

Fig. 9 Prediction effect of 1DCNN-LSTM model: **a**  $a_p = 0.1$  mm, **b**  $a_p = 0.2$  mm

**Table 2** RMSE and MAE of different models on the data set

No	RMSE		MAE		Running time (s)	
	1	2	1	2	1	2
1DCNN-LSTM	7.8	8.3	5.5	6.7	905.3	948.3
RD-1DCNN-LSTM	20.8	25.9	15.4	19.5	2413.3	2572.0
LSTM	15.5	20.8	10.7	15.6	3729.9	3802.4
Compared to RD-1DCNN-LSTM	-13.0	-17.6	-9.9	-12.8	-1508.0	-1623.7
Compared to LSTM	-7.7	-12.5	-5.2	-8.9	-2824.6	-2854.1
Rate compared to RD-1DCNN-LSTM (%)	-62.5	-68.0	-64.3	-65.6	-62.5	-63.1
Rate compared to LSTM (%)	-49.7	-60.1	-48.6	-57.1	-75.7	-75.1

## References

1. Zhaopeng H, Tielin S, Jianpeng X et al (2021) Research on tool wear prediction based on temperature signals and deep learning. *Wear* 478–479:203902
2. Shan G, Lihong Y, Chang P (2015) Application of LS-SVM regression algorithm in tool wear prediction. *China Mech Eng* 26(02):217–222
3. Xiaohong S (2016) Tool wear prediction based on wavelet analysis and neural network. *Agric Dev Equip* 10:67–69
4. Ying D, Weixi J, Jiahui L (2021) Research on milling tool wear state recognition technology based on convolutional neural network. *Mod Manuf Eng* 5:116–124
5. Jintao Z, Xu Z, Jing G (2019) Tool remaining useful life prediction method based on LSTM under variable working conditions. *Int J Adv Manuf Technol* 104(9–12):4715–4726
6. Xiaoyu Z, Lilan L, Xiang W et al (2021) Tool wear online monitoring method based on DT and SSAE-PHMM. *J Comput Inf Sci Eng* 1–18
7. Yan H, Junjie L, Yulin W (2020) Online monitoring model of tool wear based on long and short term memory convolutional neural network. *China Mech Eng* 31(16):1959–1967
8. Qinglong A, Zhengrui T, Xingwei X et al (2020) A data-driven model for milling tool remaining useful life prediction with convolutional and stacked LSTM network. *Measurement* 154:107461
9. Peng L, Yixiang H, Pengcheng X (2021) Multi-signal fusion tool wear assessment based on one-dimensional convolutional long and short term memory network. *Mech Electron* 39(05):8–14
10. Yuxuan C, Yi J, Galantu J (2018) Predicting tool wear with multi-sensor data using deep belief networks. *Int J Adv Manuf Technol* 99(5–8):1917–1926
11. Wen D, Chaoyong Z, Leilei M (2020) Support vector machine milling tool wear prediction model based on deep learning and feature post-processing. *Comput Integr Manuf Syst* 26(09):2331–3234