

Journal Pre-proofs

A Data-driven Model for Milling Tool Remaining Useful Life Prediction with Convolutional and Stacked LSTM Network

Qinglong An, Zhengrui Tao, Xingwei Xu, Mohamed El Mansori, Ming Chen

PII: S0263-2241(19)31328-4
DOI: <https://doi.org/10.1016/j.measurement.2019.107461>
Reference: MEASUR 107461

To appear in: *Measurement*

Received Date: 29 September 2019
Revised Date: 11 December 2019
Accepted Date: 27 December 2019

Please cite this article as: Q. An, Z. Tao, X. Xu, M. El Mansori, M. Chen, A Data-driven Model for Milling Tool Remaining Useful Life Prediction with Convolutional and Stacked LSTM Network, *Measurement* (2019), doi: <https://doi.org/10.1016/j.measurement.2019.107461>

This is a PDF file of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability, but it is not yet the definitive version of record. This version will undergo additional copyediting, typesetting and review before it is published in its final form, but we are providing this version to give early visibility of the article. Please note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© 2019 Published by Elsevier Ltd.



A Data-driven Model for Milling Tool Remaining Useful Life Prediction with Convolutional and Stacked LSTM Network

Qinglong An^{a,*}, Zhengrui Tao^{a,&}, Xingwei Xu^a, Mohamed El Mansori^{b,c}, Ming Chen^a

^aState Key Laboratory of Mechanical System and Vibration, School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai, 200240, China

^bMSMP – EA 7350, Arts et Métiers ParisTech, Rue Saint Dominique, BP 508, Châlons-en-Champagne 51006, France

^cDepartment of Mechanical Engineering, Texas A&M University, College Station, TX 77840, USA

*Corresponding author. E-mail: qlan@sjtu.edu.cn; Tel: +86-21-34206556

&These authors contributed equally to this work and should be considered co-first authors.

ABSTRACT

This paper introduces a hybrid model that incorporates a convolutional neural network (CNN) with a stacked bi-directional and uni-directional LSTM (SBULSTM) network, named CNN-SBULSTM, to address sequence data in the task of tool remaining useful life (RUL) prediction. In the CNN-SBULSTM network, CNN is firstly utilized for local feature extraction and dimension reduction. Then SBULSTM network is designed to denoise and encode the temporal information. Finally, multiple fully connected layers are built on the top of the CNN-SBULSTM network to add non-linearity to the output, and one regression layer is utilized to generate the target RUL. The cyber-physical system (CPS) is used to collect the internal controller signals and the external sensor signals during milling process. The proposed hybrid model and several other published methods are applied to the datasets acquired from milling experiments. The comparison and analysis results indicate that the integrated framework is applicable to track the tool wear evolution and predict its RUL with the average prediction accuracy reaching up to 90%.

Keywords: Tool condition monitoring; Long Short-Term Memory network; Convolutional neural network; Remaining useful life; Cyber-physical system

1. Introduction

In modern manufacturing systems, the performance of the cutting tool is crucial to generate a high-quality machined surface. A worn tool deforms the surface to a greater depth, tears the surface and lowers the fatigue resistance, and even cause the scrapping of the workpiece or damage to the machine tool. With the increase of flank wear, the rubbing on the workpiece surface becomes stronger, and higher cutting temperature from more friction which in turn results in the faster wear rate and eventually the tool failure. To control tool wear within the wear limit, the excessive protection strategy has been adopted widely in industries, and tool replacement is conducted too early without any theoretical basis entirely by experience, which results in tool life wasting and unnecessary tool change downtime. Therefore, if the RUL could be accurately estimated, the work schedule can be effectively optimized, the tool can be fully utilized, and the tool procurement cost will be reduced. However, the dynamic characteristics of tool wear mechanisms are difficult to be captured, considering intermittent contact with the workpiece. More and more studies have been carried out to develop effective RUL prediction systems,

which can be categorized as the data-driven model and the physics-based model. In the physics-based model, each component of the system is analyzed to establish the physical failure model based on the domain knowledge, which is utilized to predict the wear evolution and estimate the related RUL. The Taylor model [1], the Paris crack growth model [2], and the Forman crack growth model [3], are the physics-based ones. However, those models' application is limited for the following reasons. Firstly, expert knowledge is needed to drive mathematical models. Secondly, most tool life physical models depend on predefined cutting parameters and tool geometry and fail to update with online measurement data.

In this case, the data-driven model can serve as a solution, which utilizes the monitoring data provided by sensors to estimate RUL based on knowledge of statistics and data mining. A standard method is to measure the signals such as vibration, cutting force, acoustics emissions, and spindle power, which are closely connected to the tool condition. And then convert the signals into the reference model. The data-driven model emphasizes modeling from historical data and attempts to make decisions based on the online data collected. Additionally, the advanced sensing technology and computing systems are fast-growing, the data-driven tool condition monitoring (TCM) systems have received more and more attention.

The framework of the data-driven TCM model is composed of four parts: data collection, feature engineering, decision making support systems, and model verification. Various sensor signals, such as acoustic emission, vibrations, cutting force, motor power, spindle motion displacement and so on, are taken as inputs to perform feature engineering to obtain representations of cutting tool conditions. The representations are then fed into diverse decision-making systems, which are usually composed of two major parts: model training and model prediction. T. Benkedouh et al. [4] extracted features from vibration, force, and acoustic emission signals provided by the 2010 PHM competition dataset, then predicted the tool RUL based on support vector regression. Yinfei Yang et al. [5] established an integrated model combining trajectory similarity and support vector regression to estimate tool wear and its RUL. Jihong Yan et al. [6] adopted the logistic regression model to predict the drilling tool RUL based on the vibration signal. The artificial neural network is utilized by Cyril Drouillet et al. [7] to predict RUL using spindle motor power. Tobon-Mejia et al. [8] presented a stochastic model based on dynamic Bayesian network to provide a basis for tool wear estimation and its RUL prediction.

Since the monitoring sensor signal is essentially time-series data, existing studies have focused on feature extraction from statistical, frequency, and time-frequency domains. However, these approaches cannot sufficiently model the intrinsic characteristics of sequence data. On the other hand, feature selection is another big challenge needing high-quality expert knowledge. There are some sequence models, including conditional random field [9], Kalman filter [10], and hidden Markov model [11], which have powerful capabilities to process time-series data.

However, these models cannot obtain long-term dependency from the sequence data. Given that two informative features in sensor data might be segmented by all kinds of noise signals that last for an extended

period, the prediction failure occurs frequently. In recent years, the LSTM network has emerged as a promising method to solve the problem of longtime dependencies in the signal, which is often used in natural language processing and speech recognition [12-14]. Several studies have been carried out to verify the applicability of the LSTM network in RUL prediction [15, 16], and the results confirm the superiority of LSTM. However, the potential capacity of LSTM is far from fully developed in TCM because only the forward dependencies are considered, while backward dependencies from chronologically reversed sequence data received little attention. As for the dependencies in the prediction problem, the sequence data to the LSTM network is chronologically ordered, with the result that the information in the LSTM network passes along the chain structure from time step $t-1$ to time step t . However, valuable information might be filtered out. Therefore, it makes sense to obtain backward dependencies which transmit time series data in the reverse direction. The bi-directional LSTM (BLSTM) network can capture both forward and backward dependencies. Giduthuri Sateesh Babu et al. [17] proposed a deep LSTM-based architecture and a CNN-based model, respectively. The comparison results show that although the LSTM network can obtain long-term dependencies, the feature extraction capability is slightly lower than CNN. Besides, the LSTM network may not be robust when processing raw time series data directly, since the sensor data usually contains noise. Therefore, CNN is introduced for local feature extraction from the original signal sequence data.

In CNN, the hidden layers perform feature learning from the input sequence data. The three most common operations are: (a) Convolution passes the data through a set of convolutional filters, each of which activates a specific feature; (b) Rectified linear unit (*ReLU*) enables more powerful and faster learning by maintaining positive values and mapping negative values to zero; (c) Pooling simplifies the network output by conducting nonlinear down-sampling. These operations are repeated over stacked layers, and each layer learns to recognize different local features.

In this study, a hybrid model, combining CNN with the SBULSTM network, was proposed for RUL prediction in the milling process. In the CNN-SBULSTM, CNN is utilized for local feature extraction and dimension reduction. SBULSTM network, which is added to the top of CNN, is used to learn more abstract features and encode the temporal information. It should be noted that the SBULSTM network is designed to capture full local relationships before and after each time step. To verify the proposed model, several other baseline models are compared and analyzed. To sum up, the main innovations of this paper are: (a) a stacked temporal encoder considering backward dependencies by combining ULSTM with BLSTM to enhance the feature learning ability, and (b) a deep structure combining CNN with SBULSTM is designed to predict RUL with high accuracy.

This paper proceeds as follows: In section 2, related technical details are introduced, and the framework of the CNN-SBULSTM based RUL prediction model is proposed. Section 3 shows the details of the experiment

setup and the processes of data preprocessing, besides several evaluation metrics are introduced. Then, in Section 4, the RUL prediction results are analyzed. Finally, Section 5 summarizes this paper.

2. Methodology

Before the introduction of the proposed CNN-SBULSTM network, some adopted notations and necessary prerequisites are introduced firstly. In this study, the CNN-SBULSTM network is composed of two main components: the local feature detector CNN and the encoder SBULSTM network. After applying multiple CNN layers on the raw sensor data to extract discriminative features, one dense layer with dropout operation is added. Then the temporal encoder SBULSTM network is employed to obtain dependency patterns. Finally, multiple fully connected layers are built on the top of the CNN-SBULSTM network to add non-linearity to the output, and one regression layer is utilized to generate the target RUL.

2.1 Problem formulation

During the machining process, the total M -channels monitoring signal relating to RUL are collected from external sensors and internal PLC controller. The data from the n -th sample is represented by $x_n \in R^{L \times M}$, where L is the number of time step in sample x_n . And $x_n^t \in R^M$ is the t^{th} time step of x_n . The monitoring signal of cutter i throughout its useful life is given by $\chi_i = \{x_n \mid n = 1, \dots, N\}$. Considering the time interval of each sample covers Δ minutes, the tool life lasts for $\Delta \times N$ minutes. For each sample x_n , the corresponding RUL of milling tool i is $\Delta \times (N - n)$ and labeled as y_n . A total of K milling tools was used in the machining experiment. The obtained K datasets are divided into training and testing datasets. The hybrid model is trained on the training datasets and verified by the testing datasets.

2.2 Temporal Encoder: stacked Bi-directional and Uni-directional LSTM network

Existing researches [18, 19] showed that deep architecture with several stacked hidden layers could build higher-level representation from sensor data. The output of an LSTM hidden unit not only propagates over time but serves as the input to the next LSTM layer, as shown in Fig. 1. Each hidden layer in the BLSTM network receives the concatenated output of forward and backward layers below. Besides, when predicting RUL, the top-level ULSTM layer utilizes the learned representation to iteratively calculate along the forward direction, generating the input vector to the following fully connected layers.

2.3 Tool remaining useful life prediction model based on CNN-SBULSTM network

The CNN-SBULSTM based RUL prediction model scheme is illustrated in Fig. 2. The model contains three steps: (a) data preprocessing; (b) feature extraction and pattern recognition; (c) regression. In this section, the layer details of the CNN-SBULSTM network are discussed. The data preprocessing will be introduced at full length in the next section. Table 1 shows the layer details of the proposed model. As described in section 3.1, the channels number is 18, and the sequence length is 500. Therefore, the size of the input sequence data is changed

from 500×18 to 63×114 , and the information is still preserved with a smaller size after CNN. A dense layer with dropout operation is employed to connect the CNN to the SBULSTM network. The output of CNN firstly is flattened into a vector and then fed into a dense layer. The two BLSTM layers are added to learn more comprehensive and informative features. Besides, the forward layer and backward layer of each BLSTM layer share the same number of hidden units, which is set to 9 times the number of channels. The last layer of the SBULSTM network is a ULSTM layer built on the top of step (b), and the layer size is designed empirically as 150. Therefore, the output of the SBULSTM network is the hidden state at the last time step of the ULSTM layer, and its dimensionality is 150. After the LSTM layers, two fully connected layers of size $[350; 150]$ are utilized to add non-linearity to the output. The activation functions in the proposed CNN-SBULSTM network are set to *ReLU*. Finally, the learned high-level representation is fed into the linear regression layer to estimate RULs.

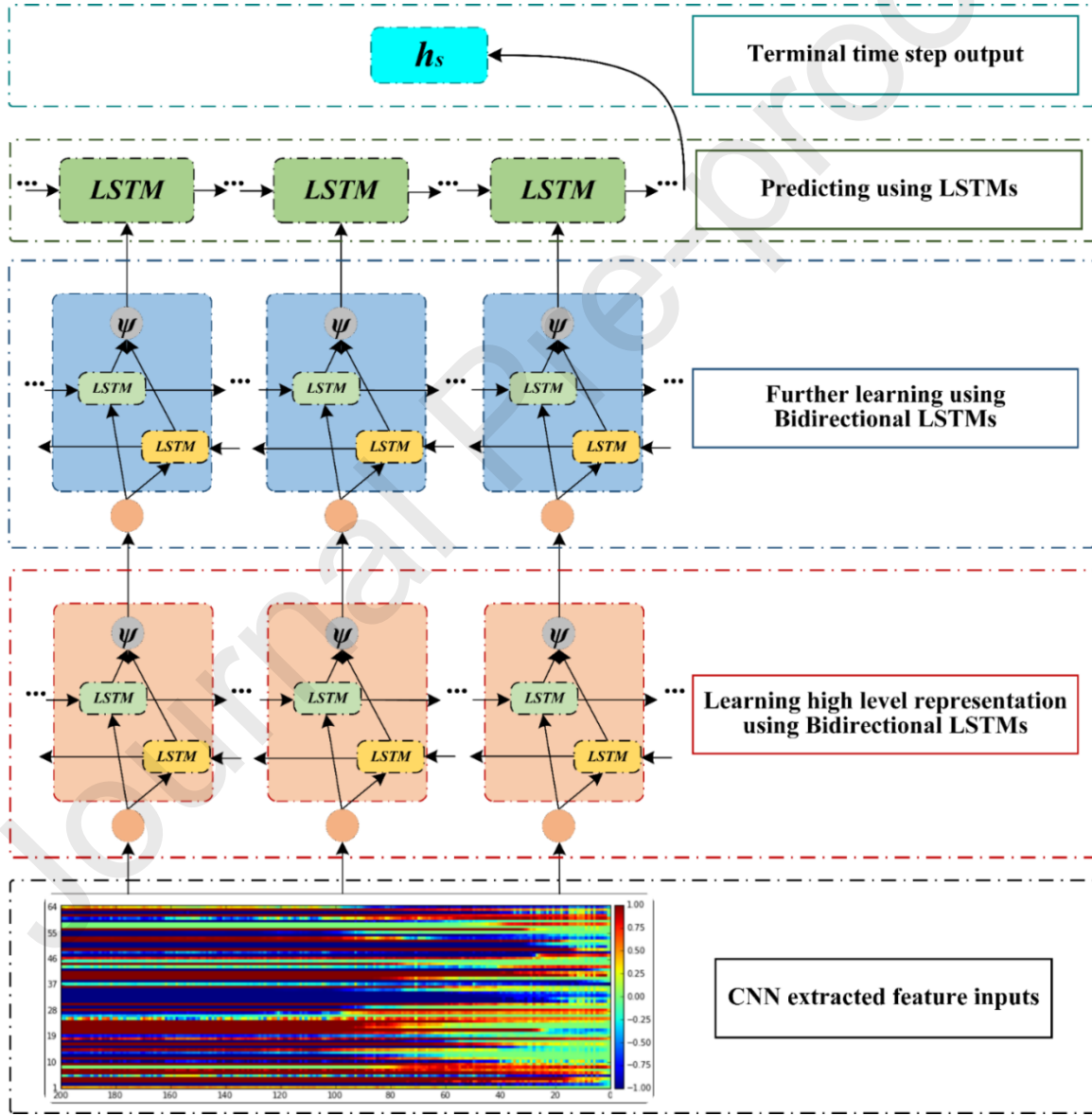


Fig. 1. The SBULSTM network consists of two BLSTM layers and one uni-directional LSTM (ULSTM) layer

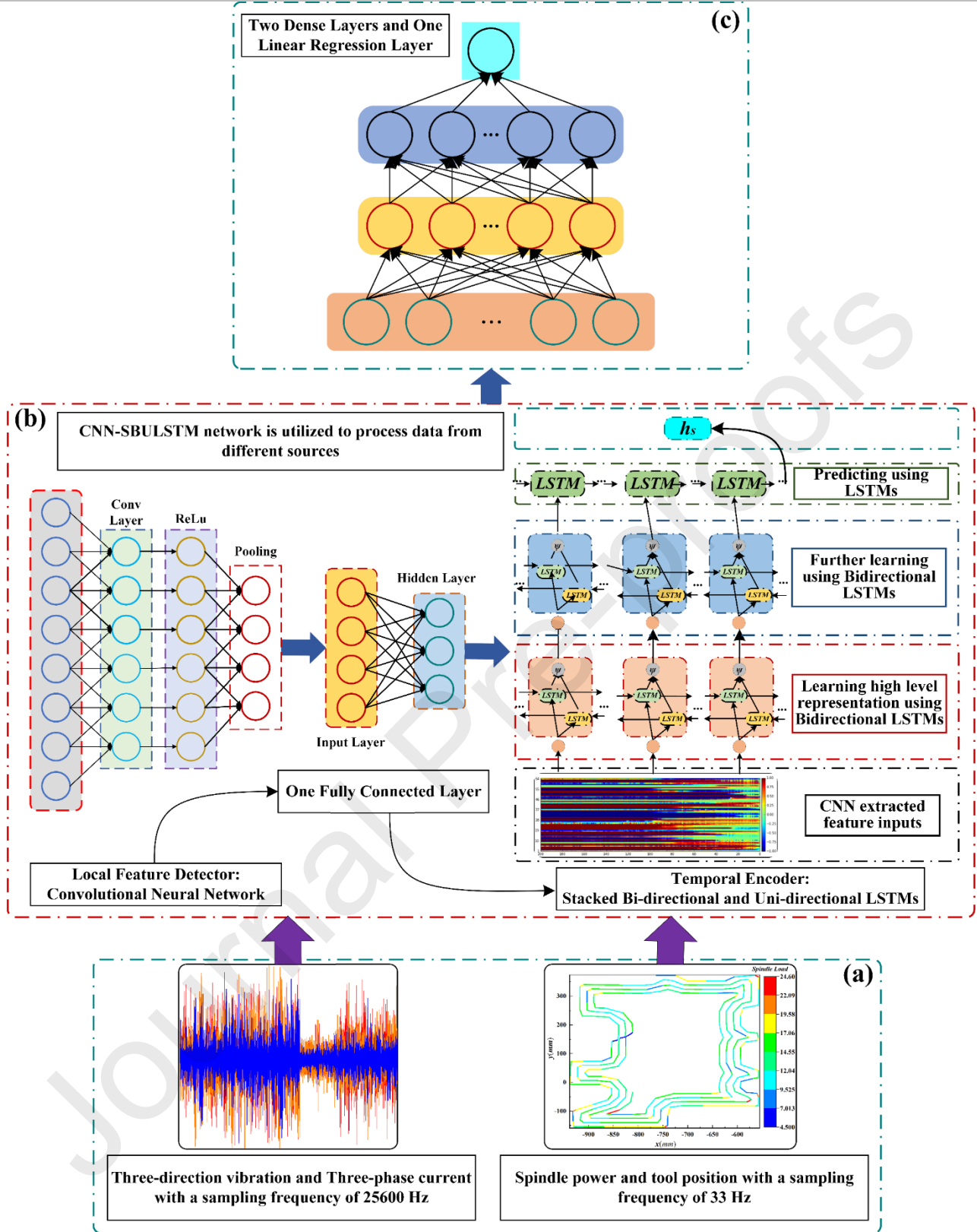


Fig. 2. CNN-SBULSTM network-based tool remaining useful life prediction system scheme including three steps:(a) data preprocessing; (b) feature extraction and pattern recognition; (c) regression.

Table 1 Layer details of the proposed model (channels=18, sequence_length=500).

CNN-SBULSTM	Layer number	Description	Details
CNN	Layer 1	1D-convolution	filters=36, kernel_size=10, strides=1, padding=same,

			activation= <i>ReLu</i>
		1D-max-pooling	pool_size=2, padding=same
		1D-convolution	filters=72, kernel_size=10, strides=1, padding=same, activation= <i>ReLu</i>
		1D-max-pooling	pool_size=2, padding=same
	Layer 3	1D-convolution	filters=114, kernel_size=10, strides=1, padding=same, activation= <i>ReLu</i>
		1D-max-pooling	pool_size=2, padding=same
	Layer 4	fully connected	layer_size=sequence_length*channels, activation= <i>ReLu</i>
		dropout	dropout_probability=0.3
	SBULSTM	Layer 5	Bi-LSTM
			units= channels*9
		dropout-wrapper	dropout_probability=0.3
		Layer 6	Bi-LSTM
			units= channels*9
		dropout-wrapper	dropout_probability=0.3
		Layer 7	Uni-LSTM
			units= 150
		dropout-wrapper	dropout_probability=0.3
Fully Connected and Linear Regression Layers	Layer 8	fully connected	layer_size=350, activation= <i>ReLu</i>
		dropout	dropout_probability=0.3
	Layer 9	fully connected	layer_size=150, activation= <i>ReLu</i>
		dropout	dropout_probability=0.3
	Layer 10	Regression	Layer_size=1

2.4 The training and testing process of CNN-SBULSTM network

According to the above description, the training and testing process of the CNN-SBULSTM network can be summarized in Algorithm 1, as listed in Table 2:

Table 2 The training and testing process of the CNN-SBULSTM network.

Algorithm 1 The training and testing process of CNN-SBULSTM network for RUL estimation

Input: The label $RUL \in R^{1 \times N}$;

The preprocessed monitoring signal samples $\chi = \{\chi_i \in R^{N \times L \times M} \mid i = 1, \dots, K_1\}$, where i denotes the index of cutter contained in the training dataset, N is the number of samples for cutter i , L is the number of timesteps for each sample, M is the number of channels.

Output: Trained CNN-SBULSTM network.

Initialize: CNN layer parameters, LSTM layer parameters and dense layer parameters.

Repeat

Forward Propagation:

Do

Firstly, CNN is applied on the training dataset;

Then SBULSTM network is added on the top of CNN to learn more comprehensive features and encode the temporal information;

End;

Dropout layer are employed to avoid overfitting.

Dense layers and linear regression layers are used for RUL estimation.

The *ReLu* function is introduced to normalized the output.

Compute the loss with the loss function *MSE*.

Backward Propagation:

Compute the error gradient using *Adam* [20] and update network parameters.

Until *Maximum iterations or within the error tolerance*

Use the trained CNN-SBULSTM to estimate the RUL on the testing datasets.

3 Experiment set-up and procedure

3.1 Experimental setup and description of the datasets

To experimentally verify the CNN-SBULSTM network for RUL prediction, a CNC vertical machining center was run to machine smartphone backplate under dry milling operation, and three experiments are conducted. The physical properties of the workpiece material and the cutting parameters are listed in Tables 3 and 4. The detailed information about the milling tools is listed in Table 5.

Table 3 The physical properties of the workpiece.

Material	Poisson's ration	Density (g/cm ³)	Young's modulus (GPa)	Hardness (HB)
TC4	0.34	4.43	113	HRC30

Table 4 The cutting parameters.

No.	Cutting speed (m/min)	Feed (mm/z)	Cutting depth (mm)	Cutting width (mm)
1	75	0.03	1.2	2
2	75	0.04	1.2	2
3	50	0.03	1.2	2

Table 5 Tool geometrical information.

No	Type	Tool diameters (mm)	Rake angle (deg.)	Clearance angle (deg.)	Flutes
1	TiAlN-coated cemented carbide	8	6	11	4
2	Solid carbide	8	6	11	4
3	TiN-coated cemented carbide	6	6	11	4

The experiment setup schematic is shown in Fig. 3(a). According to the CPS framework, the working condition information, such as spindle power and tool position, can be obtained from NC code; three-direction vibration and three-phase current signals are collected using sensors. The accelerometer is installed on the spindle and close to the end face. Current sensors are used to obtain three-phase alternating currents (AC) of spindle motor I_u , I_v , and I_w . The root mean square (RMS) of AC is used to convert the AC to the corresponding equivalent direct current, which can be obtained according to the formula below [21].

$$I_{_RMS} = \sqrt{\frac{1}{3}[(I_u)^2 + (I_v)^2 + (I_w)^2]}. \quad (1)$$

TCM directly embedded in the control system is an emerging direction, and the open CNC system allows access to the internal signals, such as the position of X, Y, Z axes and the spindle motor power. The plug-in tool can communicate with the CNC system and provide a dedicated data acquisition window through the human-

machine interface. The actual experiment setup is illustrated in Fig. 3(b). The signals from the internal PLC controller and external sensors as listed in Table 6.

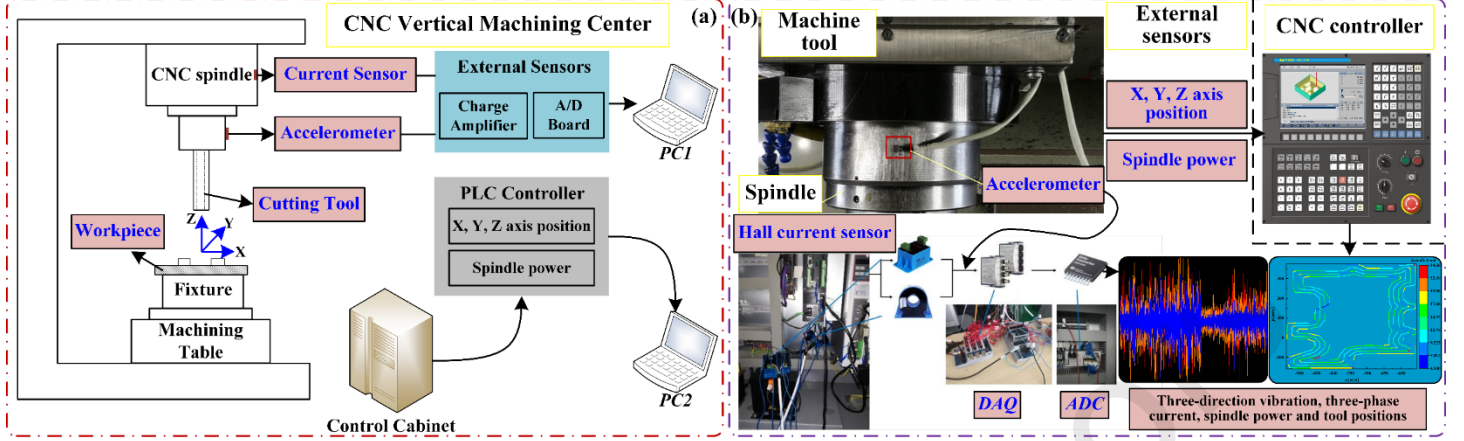


Fig. 3. Experimental set-up: (a) illustration and (b) actual set-up

Table 6 The measurement signals.

Sensor	Signal	Description
Accelerometer	Vib_x, Vib_y, Vib_z	Spindle vibration
Hall Effect Current Transducer	I_u, I_v, I_w	AC spindle motor current
	I_{rms}	RMS of three-phase AC
PLC controller	Pos_x, Pos_y, Pos_z	Axes position
	Pow	Spindle power

A total of 11-channels signal is collected by the external sensors and internal PLC controller. In terms of the sampling frequency, the sampling frequency of the plug-in tool is 33 Hz, while the sampling frequency of the vibration and current sensor reaches up to 25600 Hz. The machining time of each dataset is in units of minutes. The dataset provides a fragment of 0.5 minutes as a sample, which is given by the time sequence 1.csv, 2.csv, ..., n.csv, and the last CSV file is the monitoring data captured from the final 0.5 minutes before the failure of the milling tool. Considering the sampling frequency of the vibration and the current signal is quite high, the time step of each sample reaches up to 768 thousand. Therefore, the entire sequence of each sample is segmented into 500 sections. In each section, the maximum and average values are retained to form a new time step.

The same operation is also applied to the internal PLC controller signal. However, the results show that the maximum and average of each section is almost the same. Therefore, the average is taken, and the maximum is ignored. By preprocessing the datasets, each sample is converted into a sequence data of length 500 with a dimensionality of 18. The corresponding RUL of the i^{th} sample is calculated as $0.5 \times (n-i)$ minutes, which is the ground truth RUL. In each cutting experiment, 8 cutters were used, and the corresponding monitoring data was provided. The datasets are divided into 5 training datasets and 3 testing datasets. The testing datasets named T1, T2, and T3. Besides, T4 is defined which covers all samples from datasets T1, T2, and T3. Take the case of experiment No.1, there are 430 samples for T1, 430 samples for T2, and 320 samples for T3, and each sample has a corresponding RUL. The 130th sample from T3 is shown in Figs. 4 and 5.

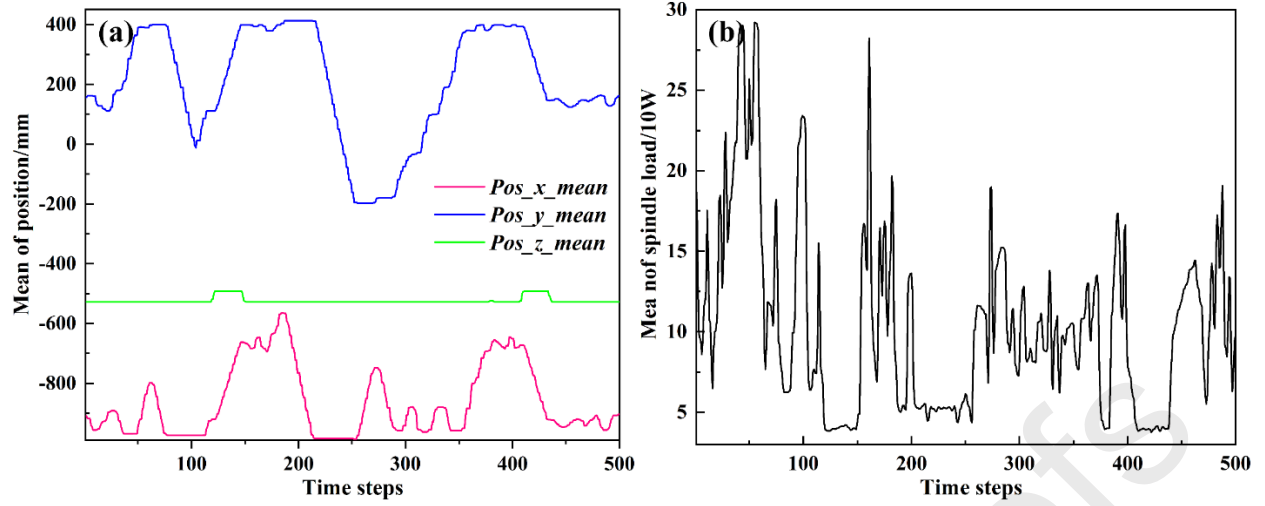


Fig. 4. The internal signals of the 130th sample from T3: (a) and (b) are the average of axes position and spindle load respectively.

3.2 Performance evaluation

To evaluate the performance of RUL prediction, the following metrics are adopted: *Score*, *Root Mean Square Error (RMSE)*, *Accuracy*. The error of the i^{th} testing sample is given by:

$$Er_i = RUL_{Real}(i) - RUL_{Prediction}(i). \quad (2)$$

- *Score*. Considering the late prediction can cause more severe consequences than the early forecast for RUL in TCM, the *score* is adopted, which quantifies the quality of the model. It is defined as follow:

$$S_i = \begin{cases} \exp^{-\ln(0.5) \cdot (Er_i/30)}, & \text{if } Er_i \leq 0 \\ \exp^{+\ln(0.5) \cdot (Er_i/50)}, & \text{else} \end{cases}. \quad (3)$$

$$S = \frac{1}{N} \sum_{i=1}^N (100 \times S_i). \quad (4)$$

Where N represents the number of samples. The final score S is the average of all samples' scores. Eq. (3) gives different penalties for early prediction and late prediction [15]. If the predicted RUL is less than actual RUL, the penalty will be smaller since there is time left to replace the tool before reaching the failure point. If the estimated RUL is higher than real RUL, the tool change will be scheduled after the failure occurrence. Under such estimation, the penalty will be larger because significant system failure may occur.

- *RMSE*. The main drawback of the *score* is its sensitivity to outliers. Therefore, *RMSE* is also employed, which gives an equal penalty weight to each prediction.

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N Er_i^2}. \quad (5)$$

- *Accuracy*. The closer the *accuracy* is to 1, the better the performance.

$$Accuracy = \frac{1}{N} \sum_{i=1}^N \exp^{-|Er_i|/RUL_{Real}(i)}. \quad (6)$$

Fig. 6 shows the *scores* and *RMSEs* on different prediction errors. $Er_i=0$ means estimated RUL equals to the

real RUL.

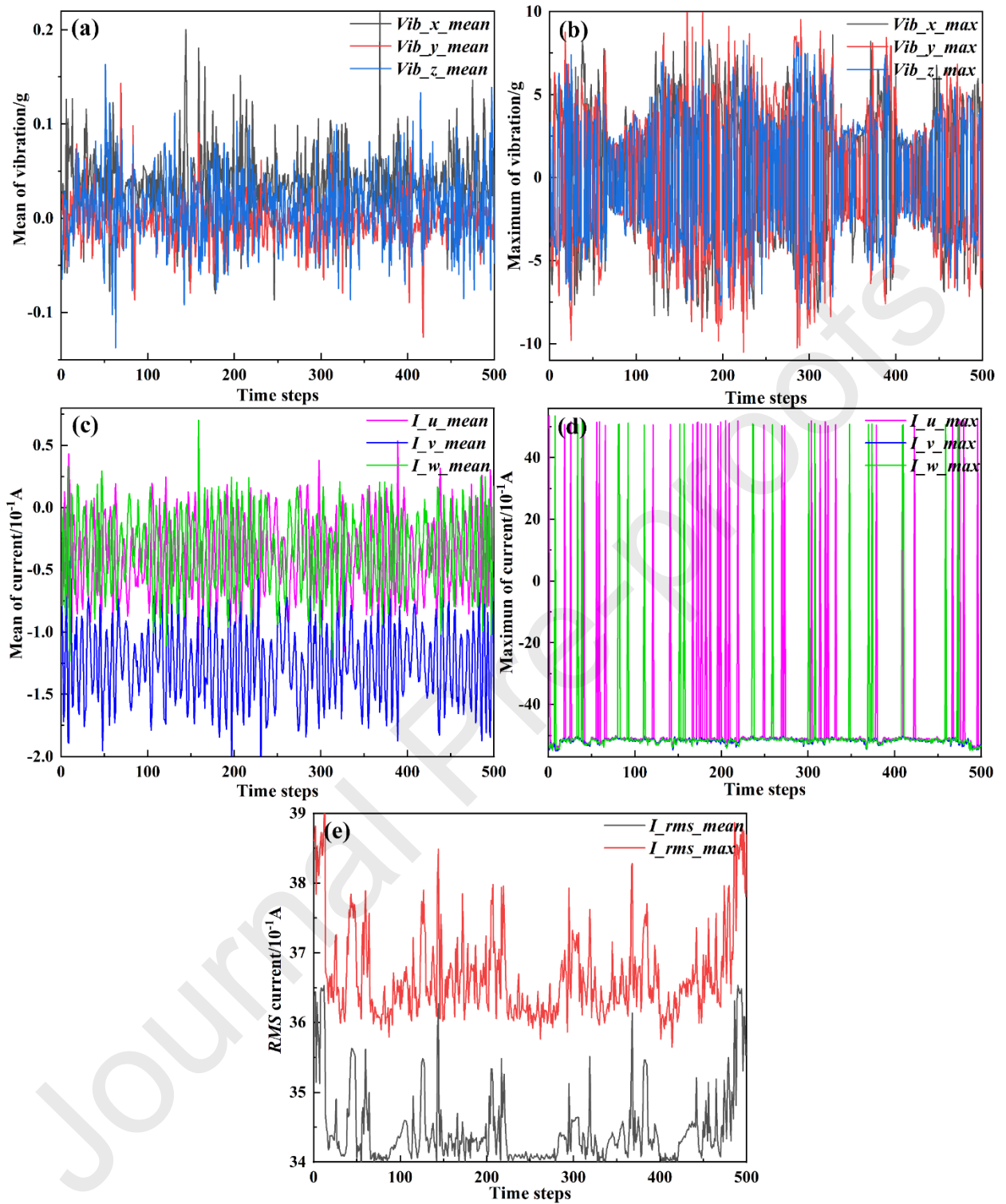


Fig. 5. The external signals of the 130th sample from T3: (a) and (c) are average spindle vibration and average motor current signals respectively; (b) and (d) are maximum spindle vibration and maximum motor current signals respectively; (e) is average and maximum of *RMS* current.

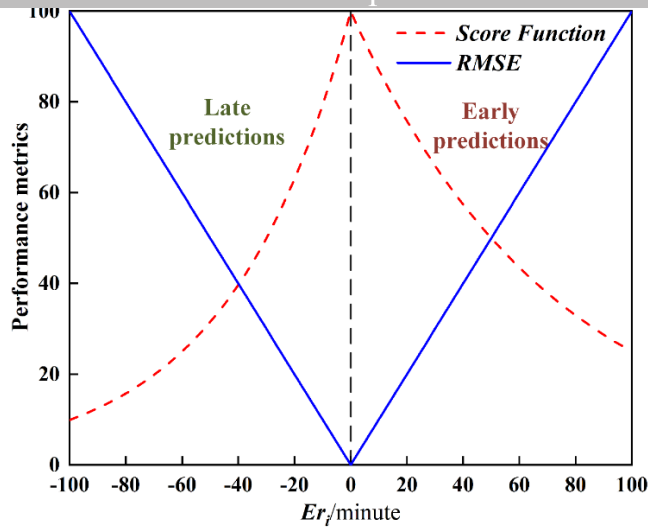


Fig. 6. The comparison of *score* and *RMSE*

4 Results Analysis and Comparison

The CNN-SBULSTM network is compared with several machine learning models and deep learning models in terms of performance metrics. Further analysis of the hidden features learning ability and relative error are also conducted to have a more comprehensive understanding of the proposed model. The following models will be compared, as shown in Table 7.

Table 7 The proposed model and other baseline models.

Model	Description
SVR	Support vector regression
RF	Random forest
Feedforward NN	A 3-hidden-layer neural network
N-layers CNN	An N-layer convolutional neural network
N-layers ULSTM	An N-layer uni-directional LSTM network
N-layers BLSTM	An N-layer bi-directional LSTM network
CNN-SBULSTM	A hybrid model (3-layer CNN + 2-layer bi-directional LSTMs + 1-layer uni-directional LSTM)

4.1 Comparison with classical machine learning models

In this section, the CNN-SBULSTM network is compared with several classical machine learning models. Considering many machine learning methods cannot process sequential data directly, feature engineering is firstly conducted. The mathematical representations of the statistical, frequency and time-frequency features are listed in Table 8.

Table 8 The extracted features [16].

Domain	Signal features	Description or Mathematical expression
Time	Root mean square (<i>RMS</i>)	$x_{rms} = \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}$
	Standard deviation (<i>std</i>)	$x_{std} = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2}$
	Maximum (<i>max</i>)	$x_{max} = \max(x_{1:n})$

	Skewness (<i>skew</i>)	$x_{skew} = E[(\frac{x-\bar{x}}{x_{std}})^3]$
	Kurtosis (<i>Kurt</i>)	$x_{kurt} = E[(\frac{x-\bar{x}}{x_{std}})^4]$
	Peak-to-Peak (<i>p-p</i>)	$x_{p-p} = \max(x_{1:n}) - \min(x_{1:n})$
	Fast Fourier Transform (<i>FFT</i>)	The frequencies and amplitudes of the first 5 peaks
Frequency	Spectral Skewness (<i>skew</i>)	$f_{skew} = \frac{\sum_{k=b_1}^{b_2} (f_k - \mu_1)^3 s_k}{(\mu_2)^3 \sum_{k=b_1}^{b_2} s_k}$
	Spectral Kurtosis (<i>Kurt</i>)	$f_{kurt} = \frac{\sum_{k=b_1}^{b_2} (f_k - \mu_1)^4 s_k}{(\mu_2)^4 \sum_{k=b_1}^{b_2} s_k}$
	Spectral Entropy (<i>entr</i>)	$f_{entr} = \frac{-\sum_{k=b_1}^{b_2} s_k \log(s_k)}{\log(b_2 - b_1)}$
Time-frequency	Wavelet Energy (<i>WT</i>)	$E_{WT} = \sum_{i=1}^N wt_{\phi}^2(i) / N$

Each sample can be denoted by a 220-dimensional vector, which then inputs to subsequent regression models, including SVR, RF, and Feed-forward NN. In the SVR method, the radial basis function kernel is adopted, and the best penalty parameter C is searched from $\log_2 C \in [-3:0.5:3]$, and the best kernel parameter gamma (g) is chosen from $\log_2 g \in [-2.5:0.5:8]$, with the step size of 0.5. For the RF model, the grid-search method from the *sklearn* library is employed to determine the optimal hyperparameters from a specified range of values. Here, the two hyperparameters, the number of trees in the forest and the maximum depth of the tree, are to be optimized. Finally, 50 trees were built, and the maximum depth is set to 11. The feed-forward NN model is composed of three hidden layers with layer sizes of [323;323;269], and the activation function is set as *sigmoid*.

The predicted RUL and the real RUL on the experiment No. 1 are illustrated in Fig. 7. The results of the CNN-SBULSTM network and RF model fluctuated above and below the actual values, and the overall trend of the tool life degradation can be captured. According to Fig. 7(a), the RF performs poorly in the 0~120 *minutes* machining time interval, and it was impossible to discriminate whether the tool was worn or not. While the predicted RULs of the CNN-SBULSTM network always fluctuate around the real RULs, and the closer to the failure threshold, the higher the prediction accuracy. In Fig. 7(b), the predicted RULs of the RF model are below the real RULs with relatively high errors, and life expectancy is less than 0 in some points which makes no sense. The predicted RULs of the CNN-SBULSTM network are always below the real RUL during 170~215 *minutes*, which is significant to replace the tool before reaching the failure point.

The performance metrics of different models are listed in Table 9. Among the expert feature-based regression models, RF performs much better, with a *score* of 64.03 and *accuracy* of 0.71, than SVR and feedforward NN, which makes sense due to the majority voting mechanism of RF. The *RMSE* indicating the prediction error has decreased by 4% and 12% compared to SVR and feedforward NN. When compared to the RF model, the CNN-SBULSTM network is much better with the *accuracy* increased by 25%, the *score* increased by 38%, and the *RMSE* decreased by 77%. To sum up, the CNN-SBULSTM network is significantly better than the other three machine learning models in terms of RUL prediction.

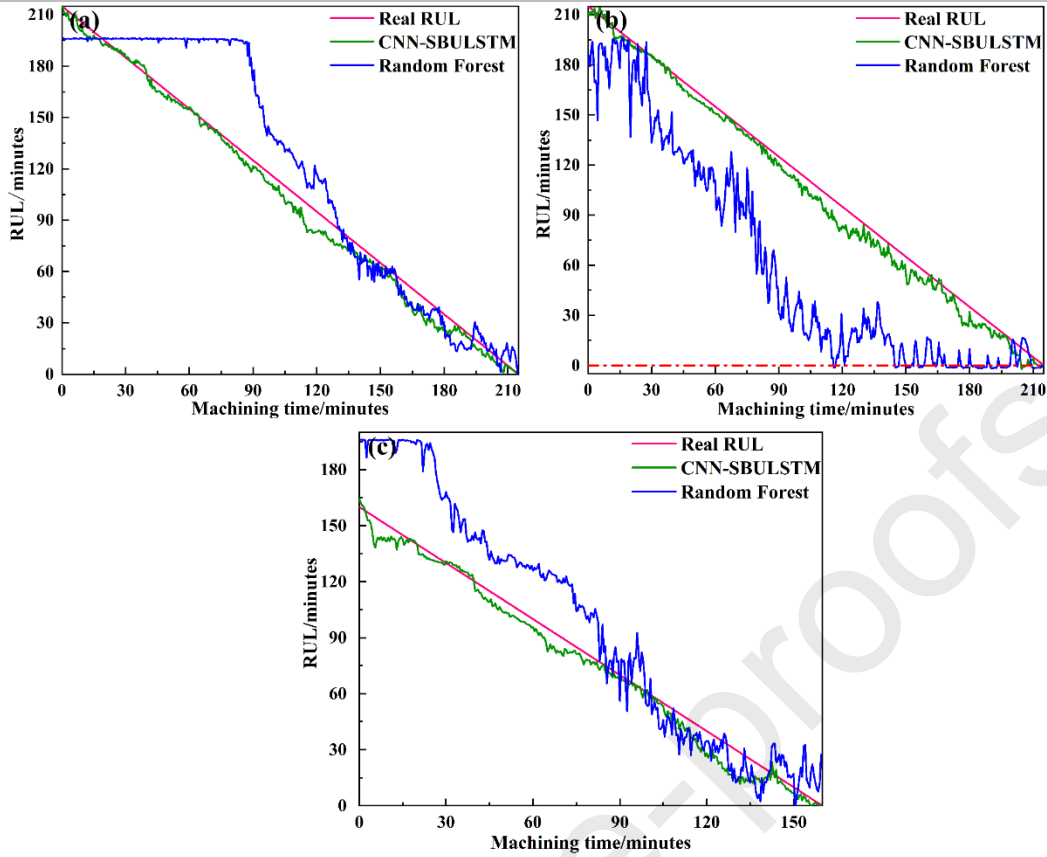


Fig. 7. RUL prediction results: (a) estimated and real RUL of T1; (b) estimated and real RUL of T2 and (c) estimated and real RUL of T3.

Table 9 The performance comparison of the proposed method with other baseline models for RUL prediction.

Models	Score				RMSE				Accuracy			
	T1	T2	T3	T4	T1	T2	T3	T4	T1	T2	T3	T4
SVR	42.72	43.93	95.59	57.50	34.54	28.14	43.62	35.20	0.61	0.62	0.92	0.70
RF	71.46	55.28	65.80	64.03	39.83	39.3	3.03	33.81	0.81	0.58	0.73	0.71
Feed-forward NN	54.91	71.76	41.82	56.50	25.64	53.52	26.28	38.35	0.67	0.77	0.55	0.67
CNN-SBULSTM	85.98	85.81	96.11	88.66	15.83	15.55	2.64	7.81	0.82	0.82	0.95	0.89

4.2 Comparison with deep learning models

Considering the CNN-SBULSTM network is a deep learning method, other deep learning models, such as the CNN, BLSTM network, and ULSTM network, are investigated in this section. Meanwhile, the influence of the number of layers is analyzed. Take the case of experiment No. 1, the comparison results are shown in Table 10. Note that the performance metrics of the CNN-SBULSTM network are listed in Table 9.

Table 10 The performance comparison of the proposed method with other deep learning models for RUL prediction on T4 (Highest performance metrics are in bold).

Models	Number of CNN/BLSTM/ULSTM layers								
	N=1			N=2			N=3		
	Score	RMSE	Accuracy	Score	RMSE	Accuracy	Score	RMSE	Accuracy
N-layers CNN	67.33	38.94	0.73	88.42	8.07	0.86	83.63	15.27	0.84
N-layers ULSTM network	59.28	38.67	0.65	77.04	23.47	0.75	67.61	25.50	0.71

The best performance of the above deep learning models is achieved with a 2-layers structure, which is slightly worse than the proposed hybrid model. The *score*, *RMSE*, and *accuracy* of the 2-layers CNN are 88.42, 8.07, and 0.86, respectively, and they are very close to the performance metrics of 2-layers BLSTM network. Considering the CNN has strong feature learning ability and LSTM is good at long-term dependency learning, both abstract learning and dependency learning are vital elements in addressing the machining process signals. Therefore, combining them can improve the ability to capture the performance degradation of the milling tool.

To verify the effectiveness of the CNN-SBULSTM network, different spindle speeds, feed rates, and tools with various coatings and structures are adopted in cutting experiments, which are commonly used in the processing site. The RUL prediction results of 2-layers CNN and the CNN-SBULSTM network on the testing datasets are illustrated in Figs. 8~10. The prediction errors of the 2-layers CNN are maintained at a high level and vary greatly. However, the hybrid model always provides a relatively accurate RUL prediction, which verifies that the model has a better generalization and higher prediction stability.

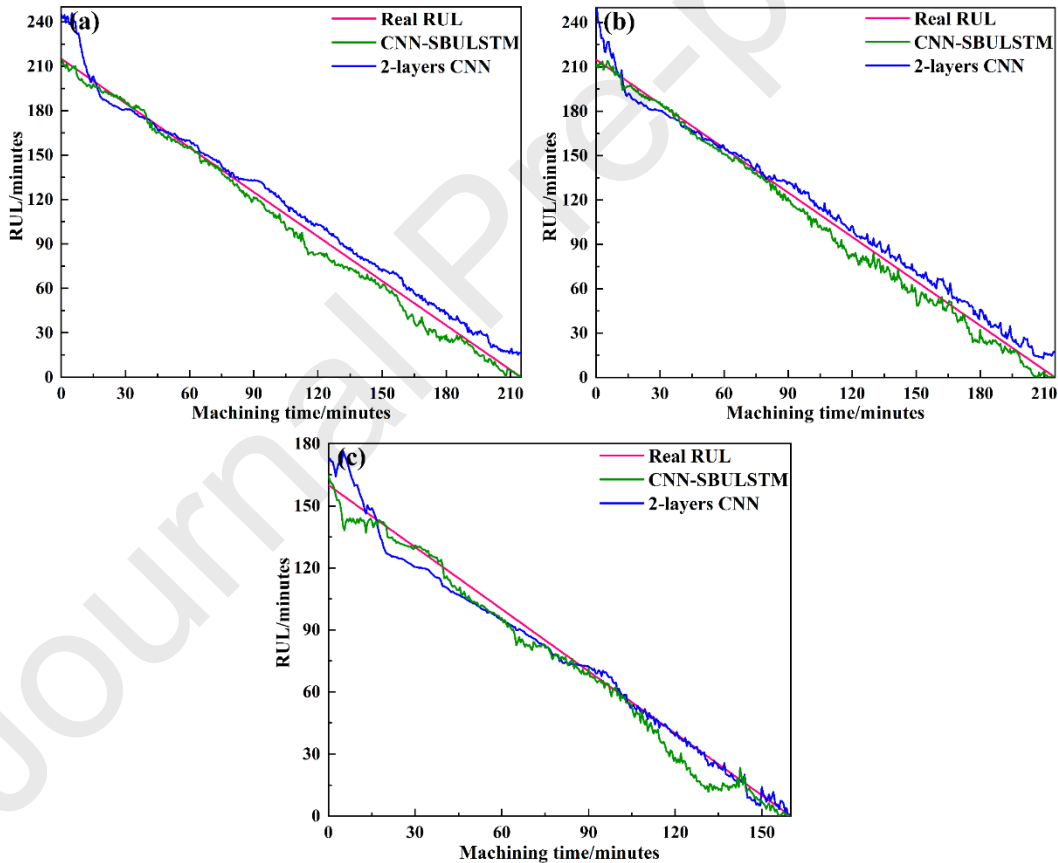


Fig. 8. RUL prediction results of experiment No.1: (a) estimated and real RUL of T1; (b) estimated and real RUL of T2 and (c) estimated and real RUL of T3.

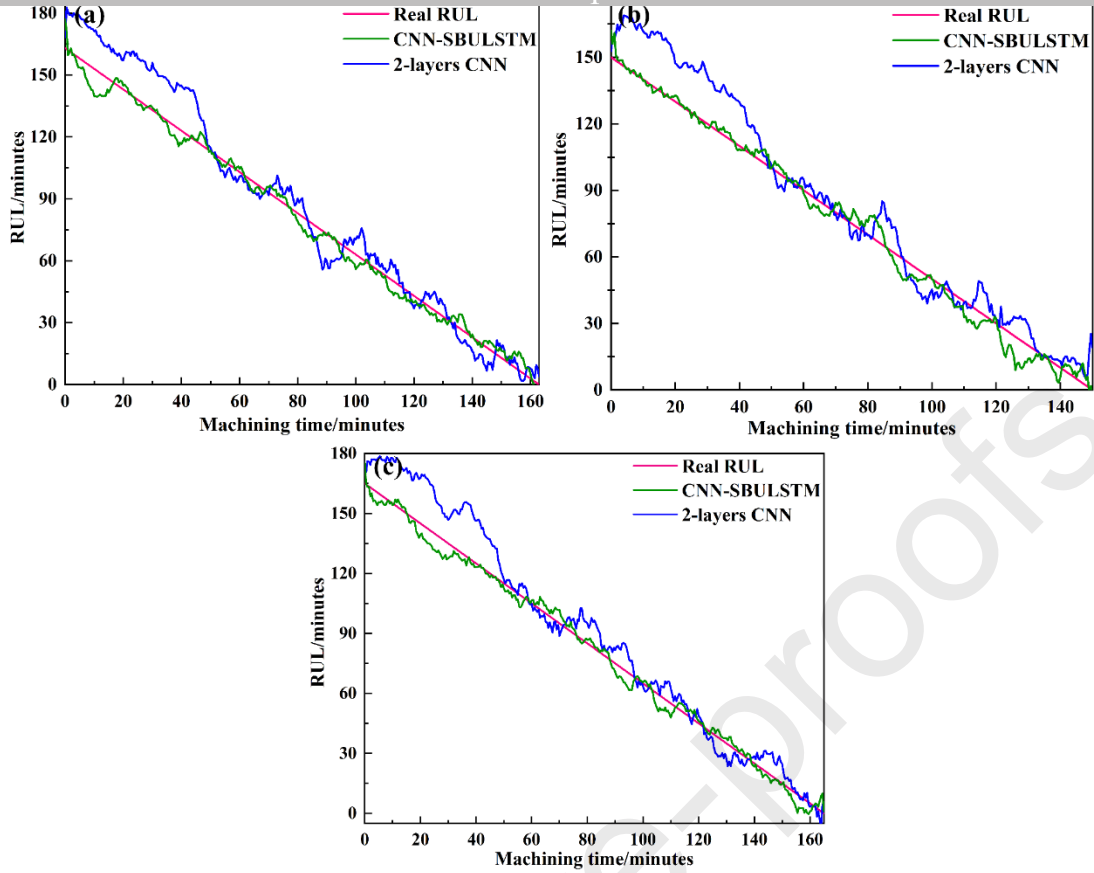


Fig. 9. RUL prediction results of experiment No.2: (a) estimated and real RUL of T1; (b) estimated and real RUL of T2 and (c) estimated and real RUL of T3.

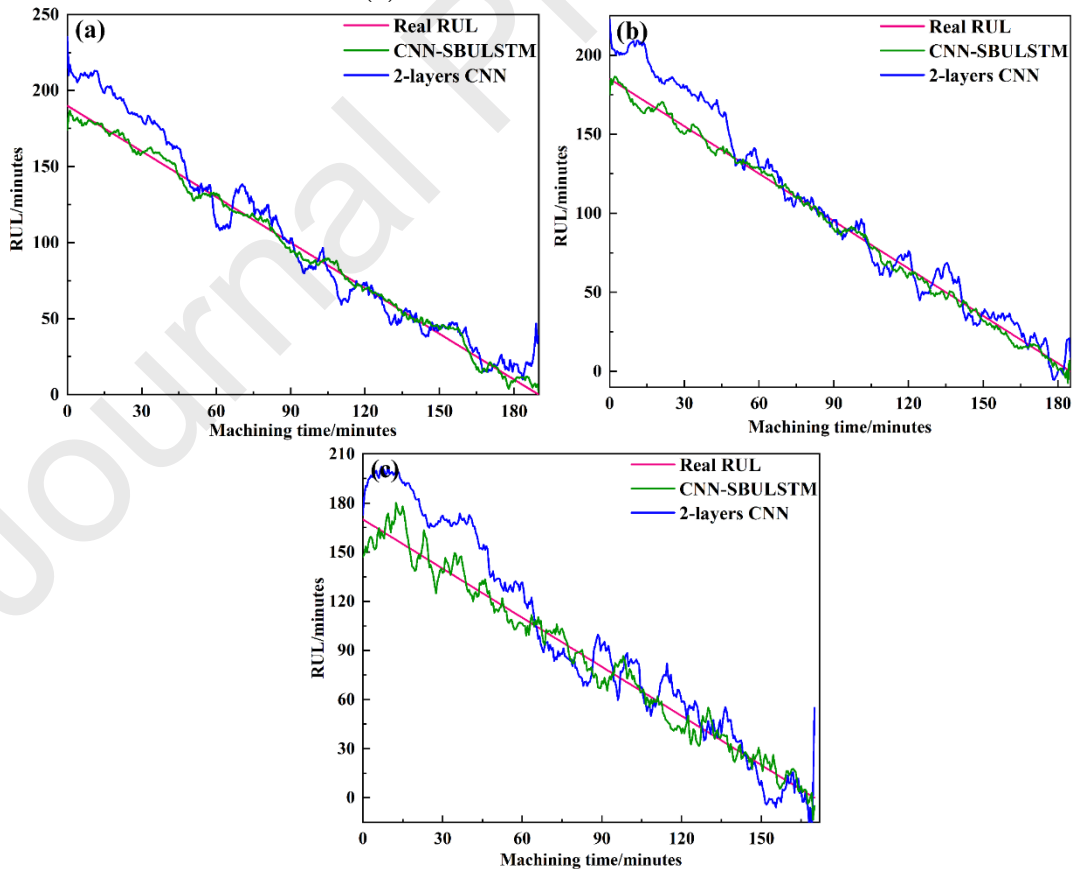


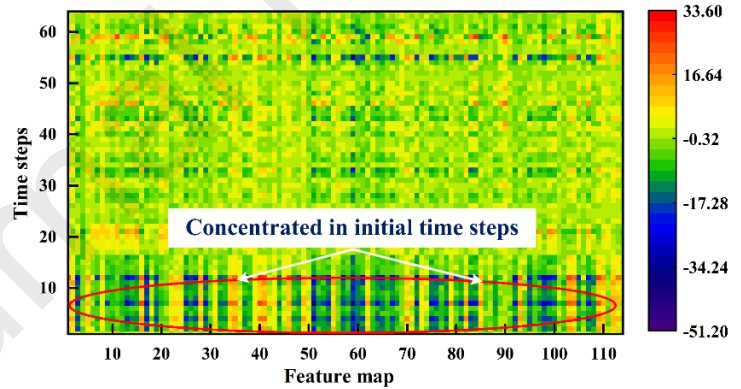
Fig. 10. RUL prediction results of experiment No.3: (a) estimated and real RUL of T1; (b) estimated and real RUL of T2 and (c) estimated and real RUL of T3.

The RUL prediction performances on T4 in each experiment are listed in Table 11. The average *accuracy* and *score* of the hybrid model are 0.90 and 92.68, which are 5% and 10% higher than CNN, respectively. Besides, the *RMSE* of the hybrid model is smaller than that of CNN for all three experiments, and the average *RMSE* is decreased by 50%. Among the three experiments, the best *score* is 95.00, the lowest *RMSE* is 3.86, and the highest *accuracy* is 0.92, which can be clearly seen from Fig. 10 that the prediction curve is very close to the real curve. Statistics indicate that the hybrid model is useful for RUL prediction in milling operations under various cutting conditions. The predicted RULs are within small errors, and the tool can be replaced in advance before the failure point. Table 11 The performance comparison of the proposed method with 2-layers CNN for RUL prediction on T4 in each experiment (Highest performance metrics are in bold).

Models	The index of experiments								
	No.1			No.2			No.3		
	<i>Score</i>	<i>RMSE</i>	<i>Accuracy</i>	<i>Score</i>	<i>RMSE</i>	<i>Accuracy</i>	<i>Score</i>	<i>RMSE</i>	<i>Accuracy</i>
2-layers CNN	88.42	8.07	0.86	82.21	12.60	0.84	82.78	11.90	0.87
CNN-SBULSTM	88.66	7.81	0.89	94.37	4.48	0.90	95.00	3.86	0.92

4.3 Hidden features learning and relative error analysis

To verify the hidden feature learning ability from the sensor signal. Figs. 11(a) and (b) visualize the features learned by the three-layer CNN on the 10th and 420th samples of dataset T3 in the experiment No.1. Fig.11(a) shows that the peaks and valleys concentrated in initial time steps; Fig. 11(b) indicates that the energy is scattered when closing to the failure point. However, there are no apparent trends when comparing the original sensor data from two samples. Therefore, the hybrid model can learn more abstract and discriminative representations from the raw signal through three-layers CNN.



(a) 3rd layer CNN feature (63 rows, 114 columns) for $RUL=210$ minutes

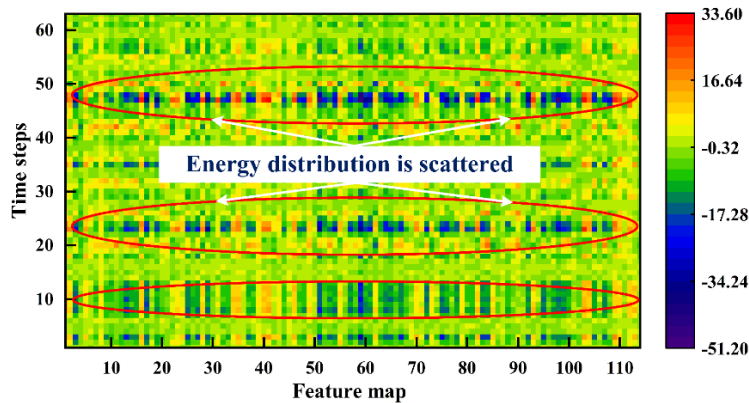
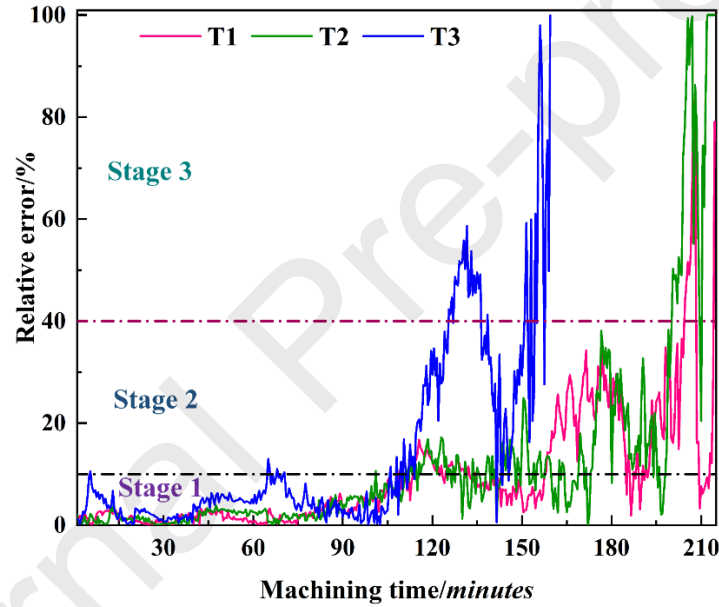


Fig. 11. CNN extracted features visualization

Finally, the relative errors of RUL prediction results are illustrated in Fig. 12. The whole machining process is divided into three stages based on the error thresholds: 10% and 40%. In the relative error stage 1, the proposed model can track the process of tool wear evolution, fit the variation of the signal characteristics, and accurately predict the RUL. Stage 1 mainly lies at the beginning of machining as the wear form is relatively simple, and the tool is very sharp. With the increase of machining time, the relative errors have been increased from stage 1 to stage 2. In stage 2, the cutting edge becomes less sharp, and the tool plows through the material with a greater force resulting in more friction. Furthermore, the depth of cut may not be steady, especially when machining a rough surface for the first time. All these reasons are random, and they can add up throughout machining to cause considerable uncertainty in predicting RUL. In the last stage, due to the severe tool wear, sudden chipping, fracture, and unexpected machine down, the relative error dramatically increases up to over 40%.

**Fig. 12.** The relative error of the RUL prediction results

4.4 Discussion

From the comparison results in Section 4.1 and 4.2, the hybrid model is the most accurate one among all discussed models. In the CNN-SBULSTM network, the dropout operation and BLSTM layer are adopted, which are the two critical factors in RUL prediction. To figure out the reasons for accuracy increase, the effects of the dropout and BLSTM layer are investigated as follows.

Dropout layer: The applied dropout layers can relieve possible overfitting [16]. Dropout is a technique that randomly ignores the selected neurons in the training process, meaning that their contribution to downstream neuron activation is temporarily removed as during the forward pass, and no weights are updated on the neurons during the back pass.

With the learning of the neural network, neuron weights tend to stabilize in the context of the network. The weights of neurons are tuned to make them specialized to specific features. Adjacent neurons begin to depend on such specialization, and if taken too far, a fragile model is established that is too specialized for the training dataset. This phenomenon, which depends on the context in which neurons are located during training, is called complex co-adaptations [22]. However, when neurons are removed randomly during training, and other neurons will have to step in and make predictions for the missing neurons to produce the representation required [23]. Therefore, the network can learn many independent internal representations, which helps to build a better-generalized network.

To verify the effectiveness of the dropout layers, the CNN-SBULSTM network without dropout is established using training datasets. Take the case of experiment No.1, the *Score*, *RMSE*, and *Accuracy* measures of CNN-SBULSTM without dropout operation are shown in Fig. 13(a)(b)(c) marked in orange. The comparison results on the dataset T4 are concluded as follows: the prediction accuracy increased by 18.7%, the score increased by 7.5%, and the RMSE decreased by 49.2%. The results confirm that dropout operation can improve RUL prediction performance because it can enable the network to learn more robust and more independent features. It is also useful in combination with different random subsets of neurons and eventually reduces the RUL prediction error.

BLSTM layer: The adopted BLSTM layer enable the SBULSTM network to consider the full context of each time step. The ULSTM network can only learn the previous relationship of each time step. However, when it comes to RUL prediction, the signals have strong temporal dependencies, and the future information of each time step should also be considered. The main idea of the BLSTM network is to process sequence data in parallel by two side-by-side LSTM layers [24]. The first is on the input sequence, and the second is on the reverse copy of the input sequence. Thus, additional context information is provided, so that the learning process of the proposed model can be more quickly and comprehensively.

Here, the performances of the CNN-SULSTM network are evaluated, and the corresponding *Score*, *RMSE*, and *Accuracy* are shown in Figs. 13(a)(b)(c) marked in green. Compared to the CNN-SBULSTM network, the CNN-SULSTM network uses ULSTM as the temporal encoder, and other parameter settings are the same as that of the CNN-SBULSTM network. Take the dataset T4 as an example, the accuracy increased by 11.9%, the score increased by 23.6%, and the RMSE decreased by 60.7%. The results indicate that the BLSTM layer is an effective structure that can capture the full local relationship before and after each time step.

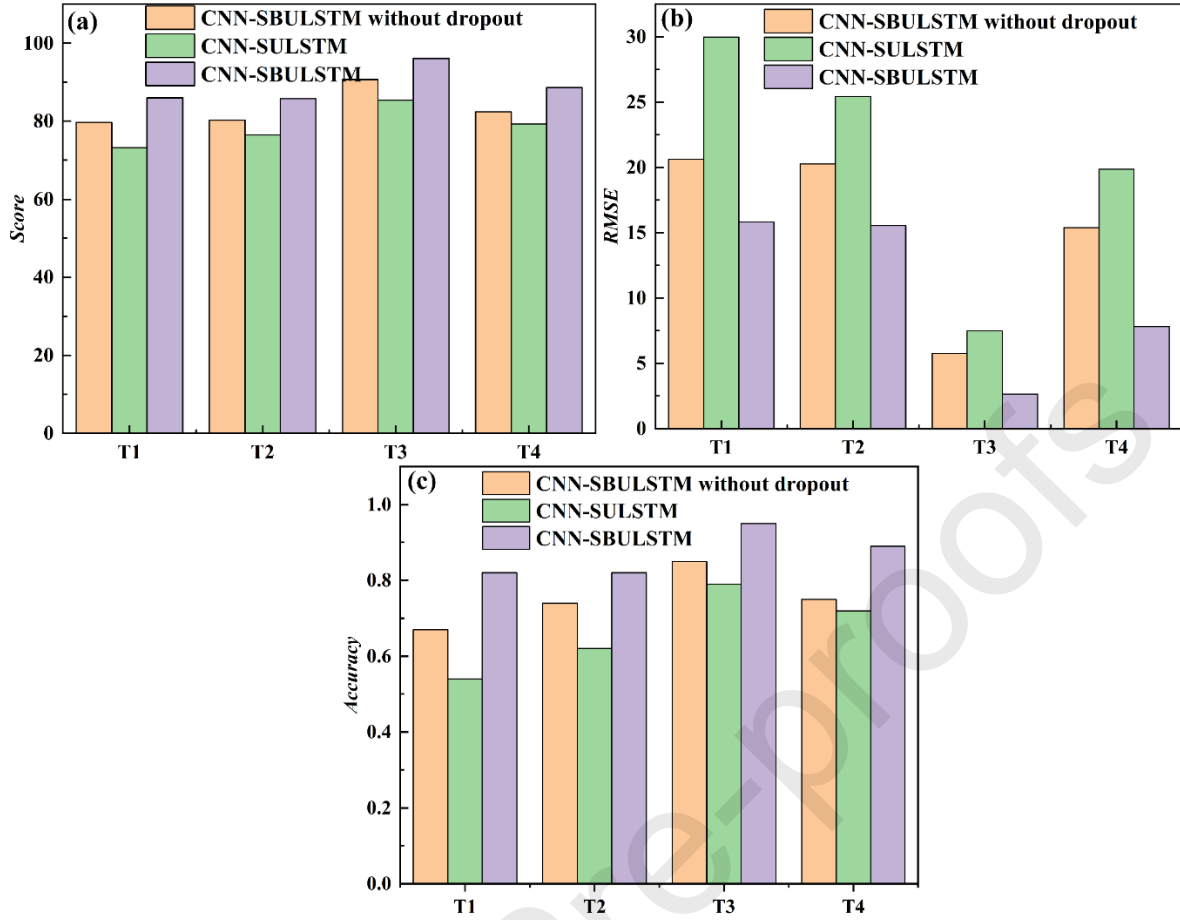


Fig. 13. Measures of CNN-SBULSTM without dropout, CNN-SULSTM, and CNN-SBULSTM on testing datasets: (a) *Score*; (b) *RMSE*; (c) *Accuracy*.

5 Conclusions

In this paper, the CNN-SBULSTM network has been proposed to track the RUL of the milling tools. Based on the results of experimental research, the following conclusions can be drawn:

- (1) In the CNN-SBULSTM network, three-layers CNN is firstly utilized for local feature extraction and dimension reduction from raw time series data. Then a stacked model of two-layers BLSTM network and one-layer ULSTM network is designed to denoise and encode the temporal information which consists of past and future contexts of each time step, long-term dependencies in local features. The stacked structure enables the CNN-SBULSTM network to obtain more in-depth and abstract features, which does not require feature engineering and difficulty-obtained high-quality expert knowledge.
- (2) Experimental results have verified the validity and feasibility of the integrated model. Three evaluation metrics (*score/RMSE/accuracy*) are selected to quantify prediction performance. Take the case of experiment No.1, the *accuracy* is increased by 25%, the *score* is increased by 38%, and *RMSE* is decreased by 77% compared with the RF model. Besides, the CNN-SBULSTM network performs slightly better than 2-layers CNN with the *score* increasing by 0.27%, *RMSE* decreasing by 3% and *accuracy* rising by 3%. The results indicate that the CNN-SBULSTM network can generate satisfactory RUL estimation.

- (3) The effects of the dropout and BLSTM layer are also investigated. The results confirm that dropout operation can improve RUL prediction performance, considering it can enable the network to learn more robust and more independent features, which are useful in combination with different random subsets of neurons and eventually reduce the RUL prediction error. Besides, the results indicate that the BLSTM layer is an effective structure that can capture the full local relationship before and after each time step.
- (4) With the extensive application of the Industrial Internet of Things, a large amount of data is produced in the machining process. Given that deep-learning methods have strong abilities to process big machinery data and capture hierarchical/multi-level representation behind data. Thus, deep learning methods will be an effective solution for TCM.

ACKNOWLEDGMENTS

This research is supported by National Key R&D Program of China (2018YFB2002200).

REFERENCES

- [1] F.W. Taylor, On the Art of Cutting Metals, American society of mechanical engineers, 1906.
- [2] B.N. Colding, A wear relationship for turning, milling and grinding, 1959.
- [3] W. Gilbert, Machining theory and practice, American Society for Metals, Cleveland, Ohio, (1950) 465.
- [4] T. Benkedjouh, K. Medjaher, N. Zerhouni, S. Rechak, Health assessment and life prediction of cutting tools based on support vector regression, *Journal of Intelligent Manufacturing*, 26 (2015) 213-223.
- [5] Y. Yang, N. Chen, Y. Guo, Z. Huang, L. Li, Y. Jiang, N. He, Research on the Milling Tool Wear and Life Prediction by Establishing an Integrated Predictive Model, *Measurement*, (2019).
- [6] J. Yan, J. Lee, A Hybrid Method for On-line Performance Assessment and Life Prediction in Drilling Operations, 2007 IEEE International Conference on Automation and Logistics, 2007, pp. 2500-2505.
- [7] C. Drouillet, J. Karandikar, C. Nath, A.-C. Journeaux, M. El Mansori, T. Kurfess, Tool life predictions in milling using spindle power with the neural network technique, *Journal of Manufacturing Processes*, 22 (2016) 161-168.
- [8] D.A. Tobon-Mejia, K. Medjaher, N. Zerhouni, CNC machine tool's wear diagnostic and prognostic by using dynamic Bayesian networks, *Mechanical Systems and Signal Processing*, 28 (2012) 167-182.
- [9] G. Wang, X. Feng, Tool wear state recognition based on linear chain conditional random field model, *Engineering Applications of Artificial Intelligence*, 26 (2013) 1421-1427.
- [10] F.O. Silva, E.M. Hemerly, W.C. Leite Filho, On the measurement selection for stationary SINS alignment Kalman filters, *Measurement*, 130 (2018) 82-93.
- [11] C.U. Mba, V. Makis, S. Marchesiello, A. Fasana, L. Garibaldi, Condition monitoring and state classification of gearboxes using stochastic resonance and hidden Markov models, *Measurement*, 126 (2018) 76-95.
- [12] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, J. Schmidhuber, A novel connectionist system for unconstrained handwriting recognition, *IEEE transactions on pattern analysis and machine intelligence*, 31 (2008) 855-868.
- [13] A. Karpathy, L. Fei-Fei, Deep visual-semantic alignments for generating image descriptions, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3128-3137.
- [14] O. Vinyals, A. Toshev, S. Bengio, D. Erhan, Show and tell: A neural image caption generator, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3156-3164.
- [15] S. Zheng, K. Ristovski, A. Farahat, C. Gupta, Long Short-Term Memory Network for Remaining Useful Life estimation, 2017 IEEE International Conference on Prognostics and Health Management (ICPHM), 2017, pp. 88-95.
- [16] R. Zhao, R. Yan, J. Wang, K. Mao, Learning to Monitor Machine Health with Convolutional Bi-Directional LSTM Networks, *Sensors*, 17 (2017) 273.
- [17] G. Sateesh Babu, P. Zhao, X.-L. Li, Deep Convolutional Neural Network Based Regression Approach for Estimation of Remaining Useful Life, Springer International Publishing, Cham, 2016, pp. 214-228.
- [18] A. Graves, N. Jaitly, A.-r. Mohamed, Hybrid speech recognition with deep bidirectional LSTM, 2013 IEEE workshop on automatic speech recognition and understanding, IEEE, 2013, pp. 273-278.
- [19] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, *nature*, 521 (2015) 436.
- [20] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, *arXiv preprint arXiv:1412.6980*, (2014).
- [21] X. Li, X. Guan, Time-frequency-analysis-based minor cutting edge fracture detection during end milling, *Mechanical Systems and Signal Processing*, 18 (2004) 1485-1496.
- [22] G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R.R. Salakhutdinov, Improving neural networks by preventing co-adaptation of feature detectors, *arXiv preprint arXiv:1207.0580*, (2012).

[23] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, The journal of machine learning research, 15 (2014) 1929-1958.

[24] Z. Cui, R. Ke, Y. Wang, Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction, arXiv preprint arXiv:1801.02143, (2018).

Qinglong An: Conceptualization, Validation

Zhengru Tao: Methodology, Software, Data curation, Writing- Original draft preparation

Xingwei Xu: Visualization, Investigation.

Mohamed El Mansori: Supervision.

Ming Chen: Writing- Reviewing and Editing

Highlights

1. The CNN-SBULSTM network is proposed to predict tool remaining useful life.
2. The CNN is utilized for local feature extraction and dimension reduction.
3. The SBULSTM network is designed to denoise and encode temporal information.
4. Time-domain and frequency-domain analysis of monitoring signals are implemented.
5. The proposed hybrid network performs better in terms of prediction accuracy.

All the authors have approved this version of manuscript and the authors declare no conflict of interest in this paper.

Sincerely yours,
Qinglong An

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☒ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Declarations of interest: none

Journal Pre-proofs