In-process tool condition forecasting based on a deep learning method

Huibin Sun[a,*], Jiduo Zhang[a], Rong Mo[a], Xianzhi Zhang[b]

[a]School of Mechanical Engineering, Northwestern Polytechnical University, Xi'an, China

[b]School of Mechanical Engineering, Kingston University, London, UK

Abstract: It is widely acknowledged that machining precision and surface integrity are greatly affected by cutting tool conditions. In order to enable early cutting tool replacement and proactive actions, tool wear conditions should be estimated in advance and updated in real-time. In this work, an approach to in-process tool condition forecasting is proposed based on a deep learning method. A long short-term memory network is designed to forecast multiple flank wear values based on historical data. A residual convolutional neural network is built to enable in-process tool condition monitoring, using raw signals acquired during the machining process. The integration of them enables in-process tool condition forecasting. Median-based correction and mean-based correction are adopted to improve the accuracy. IEEE PHM 2010 challenge data has been used to illustrate and validate this approach. Experimental study and quantitative comparisons showed that future flank wear values could be precisely forecasted during the machining process. The proposed approach contributes to prompt and reliable cutting tool condition forecasting, which will support the decision-making about cutting tool replacement in data-driven smart manufacturing.

Keywords: tool condition forecasting; deep learning; long short-term memory; data correction

1. Introduction

It is widely acknowledged that machining precision and surface integrity are greatly affected by cutting tool condition. In order to avoid unexpected downtime or scrapped parts, cutting tools are normally underused. According to Li et al. [1], only 50%–80% of tool life was actually used. Nevertheless, tool wear may be more severe than expectation in some cases. Cutting tools should be replaced in time prior to the failure. Such a decision is important to machining quality improvement and cost reduction, which requires an accurate tool wear estimation.

However, it is very challenging to precisely estimate cutting tool conditions in the machining process. Tool wear is a dynamic, time-varying, non-linear and stochastic process with many possible interferences. Every cutting tool is unique and has a specific wear curve. Due to the occlusion and cutting fluid, it is not feasible to measure tool wear conditions directly during the machining process. Moreover, there is no precise physics-based model available for tool wear estimation. Hence, data-driven intelligence has been widely used to enable in-process tool condition monitoring (TCM) [2]. However, a research gap still exists, especially in the transformation of today's manufacturing paradigm to data-driven smart manufacturing.

Data-driven smart manufacturing implements predictive manufacturing based on advanced sensing techniques, digital twin [3], big data [4], and so on [5]. By mining the trend of events in time series, data-driven smart manufacturing predicts coming abnormities and takes proactive actions [6][7]. In order to enable precise cutting tool replacement, tool wear conditions should be estimated in advance and updated in real-time. However, it is not easy to

achieve this goal. The relationship between historical and future cutting tool wear curves is still unclear.

Aiming at this problem, this work puts forward an approach to tool wear condition forecast in machining process. By capturing and using data dependencies in time series, it aims to forecast future tool wear conditions precisely. The approach supports the decision-making for prompt and reliable cutting tool replacement in data-driven smart manufacturing.

This paper is organized as follows. The first section introduces the motivation. Section 2 reviews related literature, followed by the overall explanation for the approach in Section 3. Section 4 addresses the integration of in-process TCM model. The experimental study is presented in Section 5, which is followed by concluding remarks in Section 6.

2. Literature review

In the past 30 years, TCM has been a significant research issue both in academia and industry [8]. Using cutting force, vibration and acoustic emission (AE) signals [9], many tool wear stage classification and degradation process regression models have been developed based on support vector machine [10], logistic regression [11], hidden Markov model[12], fuzzy logic [13], and other shallow neural network (SNN) models [14][15]. Data was the cornerstone for these applications [16]. Heterogeneous data, including structured process parameters and unstructured power profiles and machine vision [17], has been integrated for TCM [18]. The quality of features decides the accuracy, generalization ability and robustness, but sensitive features are tricky to extract. The accuracy, robustness and generalization ability should be improved further to make a more reliable estimation.

Compared with monitoring current states, forecasting cutting tool conditions is more meaningful for data-driven smart manufacturing [5]. Data-driven smart manufacturing aims to convert data acquired in the machining process into manufacturing intelligence to yield positive impacts on manufacturing [7]. Based on comprehensive big data analytics, an accurate prediction can be made for decision-making [19]. By making early-warnings and cutting tool replacement decisions, data-driven smart manufacturing can take proactive actions as early as possible [4]. The machining quality can be guaranteed with higher productivity and reduced cost.

Based on machine learning (ML), some models have been established to analyse the related sensing data and to discover patterns of cutting tool condition. For example, Martinov et al. [20] implemented a model to predict a cutting tool's future state based on its current state. Using least-squares method, the cutting tool wear curve was approximated by a straight line. Future cutting tool wear state was determined by the slope of this straight line and time, without considering dynamic and time-varying characteristics of cutting tool wear. Based on AE signal and force signal acquired in the machining process, Pang et al. [21] extracted and selected some dominant features on the basis of singular value decomposition (SVD). These features were used by an autoregressive moving-average (ARMA) model to predict future flank wear (VB) values. Without considering data dependencies in time series, great waves existed in the forecasted tool wear curve. Moreover, this method was not designed for non-linear dynamics of time series.

Therefore, time series forecasting should capture non-linear dependencies that can be used to predict future patterns [22]. Regarding this issue, traditional forecasting techniques were verified to be outperformed by deep learning (DL) models [23]. For example, Babu et al. [24] proposed a regression approach for remaining useful life (RUL) estimation on the basis of deep convolutional neural network (CNN). Deutsch and He [25] presented a DL-based approach for bearing RUL prediction with big data. Ren et al. [26] proposed an integrated DL-based approach for multi-bearing RUL prediction by combining both time domain features and frequency domain features. Li et al. [27] proposed a data-driven approach for RUL prognostics by using CNN.

However, with the increasing large amounts of data, the complexity of forecasting models has been increased to capture long-term non-linear dynamics of time series. By retaining the recent memories of input patterns, recurrent neural network (RNN) was able to map from the entire history of previous inputs to target vectors in principal [28]. It allowed a memory of previous inputs to be kept in the network's internal state. However, traditional RNN may not capture long-term dependencies. Long short-term memory (LSTM) network, a variant of RNN, was put forward to prevent back propagated errors from vanishing or exploding [29]. Due to its ability to capture long-range dependencies and nonlinear dynamics in time-series data, LSTM has been successfully adopted in various predictive applications, including health state estimation for bearings [29], and machine RUL prediction [30]. LSTM has also been applied in TCM [31]. Raw sequential multi-sensory signals were used to monitor tool wear condition without any pre-processing. Compared with traditional RNN, the LSTM-based TCM model fitted the cutting tool wear curve better. Gated recurrent unit (GRU) was regarded as an updated version of LSTM with a simple structure [32]. A hybrid scheme accomplished by a deep heterogeneous GRU model enabled single-step tool wear condition prediction. To capture the temporal pattern hidden in the sequential input, a local feature extraction method was designed [33]. An intermediate layer was designed to capture the inherent relation for long-term prediction. However, these models were state-related, rather than time-related. Data dependencies in time series have not been effectively decoded and continued, especially for multi-step forecast of future cutting tool conditions.

In summary, forecasting cutting tool conditions is of great significance for data-driven smart manufacturing. Compared with ML-based methods, DL-based models are more suitable for capturing non-linear data dependencies in time series. However, in order to enable multi-step forecast of future cutting tool conditions, more efforts should be made to decode and continue data dependencies in time series further.

3. In-process tool condition forecasting approach

In this paper, an approach to in-process tool condition forecasting (TCF) is proposed. It focuses on the time-varying cutting tool wear curves during the machining processes. It estimates multiple *VB* values in the nearest future by using several sequential *VB* values measured in the latest past. Then, how to capture and continue data dependencies in historical *VB* values becomes the most significant issue. Although standard RNN can retain the recent memories of input patterns, it has difficulty to model time series data when long time lag is present [29]. To deal with this problem, an LSTM network is used in this work. By capturing long-range dependencies in time-series data, the LSTM network contributes to accuracy improvement of

the in-process TCF approach. In Fig. 1, several sequential *VB* values measured in the latest past, denoted by $y_{t-n+1}, y_{t-n+2}, \ldots, y_t$ in sequence, are inputs of the LSTM network. Here, variable *n* stands for the number of historical data. The LSTM network outputs multiple *VB* values in the nearest future, denoted by $y_{t+1}, y_{t+2}, \ldots, y_{t+m}$. Variable *m* stands for the number of forecasted data. A deterministic model, *H*, aims to learn a mapping from the input values to the output values. Formally, there are:

$$(y_{t+1}, y_{t+2}, \cdots, y_{t+m}) = H(y_{t-n+1}, y_{t-n+2}, \cdots, y_t) \tag{1}$$

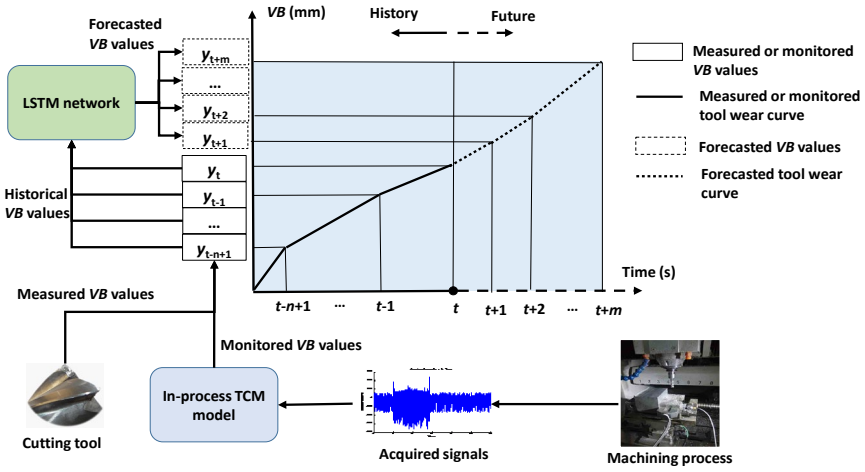


Fig. 1 In-process TCF approach

Normally, it is very difficult to measure *VB* values directly during the machining process. Under this case, the *VB* values judged by the in-process TCM model could be used. Then, cutting force, vibration and AE signals are acquired in the machining process and used by the in-process TCM model to estimate current tool conditions.

3.1 Structure of the LSTM network

As Fig. 2 shows, the LSTM network is logically composed of an encoder, a decoder and a context tensor [31]. The encoder accepts *n* historical and sequential *VB* values and outputs the context tensor. The decoder accepts the context tensor and estimates *m* future and sequential *VB* values. Both the encoder and the decoder use the same LSTM unit [32].

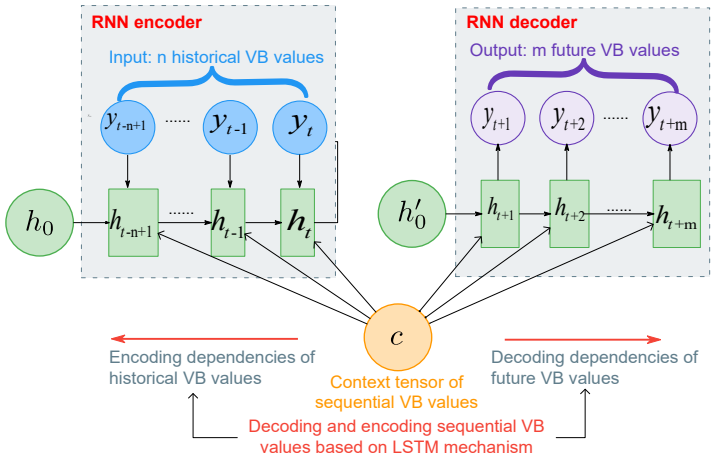


Fig. 2 Logic of LSTM network

The LSTM network used in this work is shown in Fig. 3. Numbers and variables in brackets mean dimensions. The duplicated vector layer is used to extend 2D vectors to 3D vectors. The former two LSTM units are encoders, and the latter two LSTM units are decoders. Driven by the LSTM mechanism, the threshold of an LSTM unit is adjusted to save or delete time series information. Then, the network can use or ignore historical *VB* values in forecasting future tool conditions. By modifying *n* and *m*, future *VB* values in different time ranges can be forecasted. In addition, the LSTM network also contributes to back propagate error prevention from gradient vanishing or exploding problems [34].

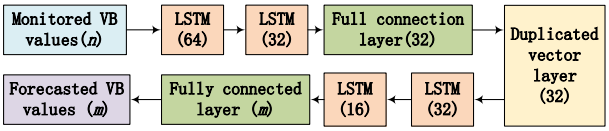


Fig. 3 Structure of LSTM network

If a new time range is used, it is time-consuming to train a new LSTM network. If the input *VB* values are insufficient, the LSTM network could work by using outputs of itself. On the basis of this so-called self-referring mode, the LSTM network can extend its time range. However, if the thresholds are not updated accordingly, small errors will accumulate to great errors.

3.2 Mechanism of the LSTM unit

As shown in Fig. 4, LSTM unit *t* includes two states, the long-term state c_t and the short-term state h_t [32]. The forget gate F_t, the input gate I_t and the output gate P_t are added to regulate the unit states. The forget gate F_t deletes the information from the previous long-term state $\boldsymbol{c}_{t-1}$. The output gate P_t controls the formation of the current short-term state h_t using the information from the long-term state c_t. Its value is in the range [0, 1]. Values 1 and 0 respectively stand for completely saving or removing historical information.

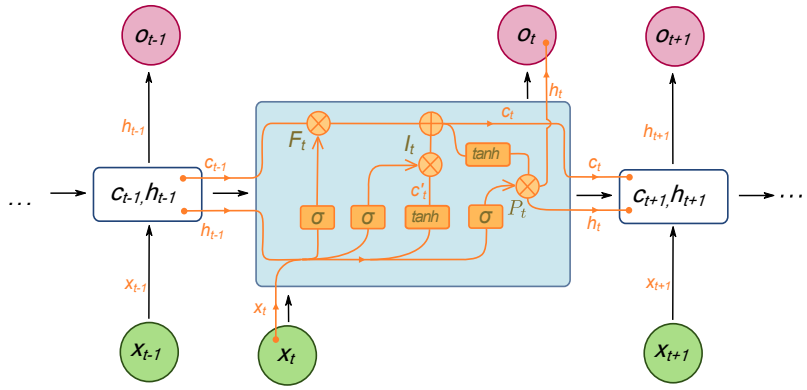


Fig. 4 LSTM unit *t*

The input gate I_t is calculated as

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

Here, σ is a single-layer neural network that uses the transfer function Sigmoid, and x_t is the input tensor of time sequential sample. W_i and b_i are weight and bias, respectively.

The forget gate F_t and the output gate P_t are calculated as

$$F_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{3}$$

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

The long-term state c_t is calculated as

$$c_t = F_t \odot c_{t-1} + i_t \odot c'_t \tag{5}$$

where $\odot$ denotes the element-wise product.

A candidate parameter for unit state updating is named c'_t and calculated as

$$c'_t = \mathrm{Tanh}(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{6}$$

where Tanh is a single-layer neural network which uses the transfer function Tanh.

The hidden state o_t is calculated as

$$o_t = h_t = P_t \odot \mathrm{Tanh}(c_t) \tag{7}$$

3.3 Network training and performance evaluation

In the training stage, the LSTM network is optimized by adjusting weights of three Sigmoid functions and one Tanh function. Adam optimizer is used to optimize the network. It is also robust to train and to make the whole training process steady. Based on DL optimization method, an excellent LSTM network could be trained.

Although a deeper LSTM network means better accuracy, it still fails sometimes since more calculation resources are needed and the generalization ability is impaired. Finding a proper depth becomes a big obstacle to a successful LSTM network. Since there is very tiny difference among candidate networks with diverse depths, an approximate estimation on depth is considered. After every training epoch, a loss evaluation is done by using root mean square error (RMSE), which is defined as

$$RMSE = \sqrt{\frac{1}{s}\sum_{i=1}^{s}(y_i - \hat{y}_i)^2} \tag{8}$$

where s is the sample number, and y_i and $\hat{y}_i$ are respectively measured and forecasted *VB* values. The training ends when the training error decreases and the validation error is increasing. The best network is chosen as the final solution.

Moreover, mean absolute error (MAE) is also used to evaluate and compare the LSTM network's performance. It is defined as

$$MAE = \frac{1}{s}\sum_{i=1}^{s}\left|y_i - \hat{y}_i\right| \tag{9}$$

4. Integration of in-process TCM

The LSTM network discussed above uses measured *VB* values as input. Generally, it is very challenging or even impossible to measure *VB* values directly in the machining process. In order to enable in-process TCF, an in-process TCM model is integrated. In order to improve the accuracy, robustness and generalization ability, the model is implemented on the basis of the residual neural network (ResNet) [35][36].

4.1 ResNet-based in-process TCM model

The ResNet-based in-process TCM model is shown in Fig. 5. At time *t*, its inputs are seven-dimensional time series signal segments as $X_t = [x_{t1}, x_{t2}, x_{t3}, x_{t4}, x_{t5}, x_{t6}, x_{t7}]$, including X force, Y force, Z force, X vibration, Y vibration, Z vibration and AE signals. A deterministic model, *G*, aims to learn a mapping from raw signal segments to a single output $\tilde{y}_t$, *VB* value at time *t*. Formally, there are:

$$\tilde{y}_t = G(X_t) = G(x_{t1}, x_{t2}, x_{t3}, x_{t4}, x_{t5}, x_{t6}, x_{t7}) \tag{10}$$

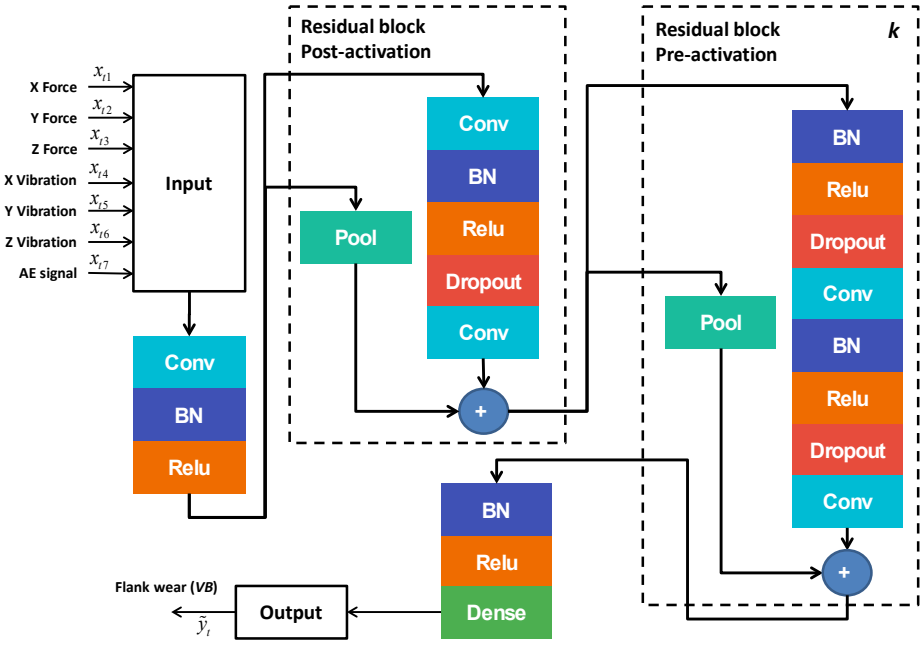


Fig. 5 ResNet-based in-process TCM model

The ResNet-based in-process TCM model includes a post-activation residual block and many pre-activation residual blocks. The post-activation residual block starts with a convolutional layer (Conv). A batch normalization layer (BN) is used to accelerate the calculation and optimize the network. A rectifier linear unit (ReLu) is adopted to manipulate pre-activation architecture. A dropout layer is added to diminish over-fitting in the training process and to improve the generalization ability. In order to reduce computation time and eliminate redundant features, a max-out pooling (Pool) layer is included in the shortcut of residual learning framework (RLF). Then, the post-activation residual block ends with a convolutional layer. The

structure of convolutional layers and max-out pooling layers use classical parameters. The kernel size is 3 and the stride is 2. In every block, kernel numbers of two convolutional kernels are 64 and 128, respectively. A pre-activation residual block starts with a BN, followed by Relu, Dropout, Conv in sequence. This structure repeats in every pre-activation residual block. Finally, a fully-connected layer (Dense) is used to produce a *VB* value.

An adaptive feature extraction mode is implemented based on the convolutional layer. In the convolutional process, the signal segments do convolutional operations with kernels of various sizes. During these steps, some specific patterns are emphasized. Noise is automatically filtered using convolution operations. Based on the good selectivity, the convolution kernel enables adaptive filtering. Parameters of the convolution kernel can be optimized by the training process. After necessary training, the kernels are able to fully extract features from the raw signals. The feature extraction process is nearly the same as the manual process.

Extracted features are selected based on multi-layer perception (MLP), which is a feedforward neural network that simulates how signals pass two neurons. MLP is made up of many neurons in different layers. The output of a layer is the input of next layer. Assisted by the back-propagation (BP) algorithm, tool wear condition information could pass through these layers. Then, the low-level features are assembled into the high-level features. The high-level features approximate tool wear conditions, as no obvious clue could be found from the low-level features.

Gradient always vanishes when the multiplication of two gradients becomes smaller and smaller. Then, information produced by low-level layers may be wiped out, although tool wear condition always exists in minor clues. Based on RLF introduced by He et al. [35][36], the gradient vanishing phenomenon is prevented. Then, ResNet and RLF are merged to leverage their advantages. Moreover, over fitting is avoided by using BN, ReLu and dropout layers.

4.2 Data correction

As discussed above, the integration of ResNet-based in-process TCM enables in-process TCF. However, the compound errors should never be ignored. The ResNet-based in-process TCM model uses sensitive features extracted from raw machining signal segments. Without considering dependencies of time series *VB* values, monitored tool wear curves may wave greatly when the machining process is going on.

Based on these *VB* values, future *VB* values judged by the LSTM network are not accurate and reliable enough. Waves in the monitored tool wear curves may be wrongly regarded as inflection points by the LSTM network, especially in the normal wear stages of cutting tools. Consequently, errors of the in-process TCM model may be amplified by the LSTM network.

Therefore, *VB* values judged by the LSTM network, denoted by $\tilde{y}_{t+1}, \tilde{y}_{t+2}, \cdots, \tilde{y}_{t+m}$, should not be outputted directly. Efforts must be made to reduce the compound errors. Consequently, model H in Eq. (1) can be converted into the following formation.

$$(\tilde{y}_{t+1}, \tilde{y}_{t+2}, \cdots, \tilde{y}_{t+m}) = H(G(X_{t-n+1}), G(X_{t-n+2}), \cdots, G(X_t)) \tag{11}$$

A deterministic model, C, aims to build a function from original forecasted *VB* values to the final outputs. Formally, there are:

$$y_t = C(\tilde{y}_t^{(1)}, \tilde{y}_t^{(2)}, \cdots, \tilde{y}_t^{(q_t)}) \tag{12}$$

where is y_t the final forecasted *VB* value regarding time *t*. q_t is the number of available forecasted *VB* values regarding time *t*, while $\tilde{y}_t^{(k)}$ is the *k*-th original forecasted *VB* value.

Here, model *C* is implemented by two correction functions, including mean-based correction and median-based correction. Mean-based correction finds the mean of the original forecasted *VB* values. The final *VB* value regarding time *t* equals to the average of all available original forecasted *VB* values regarding time *t*

$$y_t = \frac{1}{k}\sum_{k=1}^{q_t} \tilde{y}_t^{(k)} \tag{13}$$

Median-based correction finds the median of the original forecasted *VB* values, which can be done by arranging all numbers from smallest to greatest. If q_t is an odd number, the middle one is picked as the final forecasted *VB* value regarding time *t*:

$$y_t = \tilde{y}_t^{[(q_t+1)/2]} \tag{14}$$

If q_t is an even number, there is no single middle value. The mean of two middle values is calculated as the final forecasted *VB* value regarding time *t:*

$$y_t = [\tilde{y}_t^{(q_t/2)} + \tilde{y}_t^{(q_t/2+1)}]/2 \tag{15}$$

In fact, a real cutting tool wear curve is non-decreasing. Should waves in original forecasted *VB* values be eliminated completely to output a non-decreasing curve? The answer is no, because such an artificial operation would enlarge errors arbitrarily. In order to improve the accuracy, future tool wear condition should always be forecasted and corrected dynamically. The performance of in-process TCF approach should be evaluated quantitatively using MAE and RMSE defined above. Additionally, in order to reduce the compound errors further, the ResNet-based in-process TCM model and the LSTM network should be trained together carefully.

5. Experimental study

5.1 Experimental setup

The IEEE PHM 2010 challenge dataset [37] was used to verify the approach. The machining experiments were performed on a CNC machine (Röders Tech RFM760). Some stainless steel (HRC52) workpieces were milled with 10400 r/min spindle speed, 1555 mm/min feed speed, 0.125 mm radial cut depth in Y direction, and 0.2 mm axial cut depth in Z direction. A Kistler quartz 3-axis dynamometer was installed between the workpiece and the machining table. A Kistler piezo accelerometer was adopted to acquire vibration signals in three coordinators. A Kistler AE sensor was mounted on the workpiece to capture acoustic waves. All signals were collected by NI DAQ PCI 1200 at 50 kHz frequency. A 7-dimension dataset was saved as a

CSV file with 7 columns. These columns stood for X force (N), Y force (N), Z force (N), X vibration (g), Y vibration (g), Z vibration (g) and AE (V) signals in sequence. Six three-flute cutting tools (respectively named C1, C2, C3, C4, C5 and C6) were used in the experiments. Every cutter was used for about 315 cut cycles under the same machining condition. *VB* values of cutting tool C1, C4 and C6 were recorded after every cut cycle. Based on Keras framework, all algorithms are implemented with TensorFlow. An Intel Xeon E7 CPU and an Nvidia Quadro M2000 GPU were used.

5.2 Performance evaluation of the LSTM network

In the training process, the inputs were several measured time series *VB* values, denoted by $y_{t-n+1}, y_{t-n+2}, \ldots, y_t$ in sequence. The outputs were multiple time series *VB* values, denoted by $y_{t+1}, y_{t+2}, \ldots, y_{t+m}$ ($n \leqslant t \leqslant L$-$m$, L is the sample number) in sequence. The model was trained by using datasets of cutting tools C1, C4 and C6, because every sample of them had a corresponding *VB* value. As shown in Table 1, three setups were adopted such that two datasets were used for training and the other one was used for testing. For every setup, the dataset for testing never appeared in the training process.

Table 1 Setup and testing error of the LSTM network

Setup No.	Training dataset	Testing dataset	MAE (μm)	RMSE (μm)
S1	C4, C6	C1	2.607	4.307
S2	C1, C6	C4	0.946	2.734
S3	C1, C4	C6	0.440	0.908

TensorBoard was used to monitor the training process. The time consumption of a training iteration was less than 1s. When using testing datasets, MAE and RMSE values were up to 2.607 μm and 4.307 μm, respectively. Although the trained LSTM networks never saw the testing datasets, the forecast accuracy was good enough.

Moreover, both measured *VB* values and forecasted *VB* values had various ranges. In order to investigate this issue, a comparison was done by using orthogonal test. As shown in Fig. 6, both smaller forecasted ranges and bigger historical ranges led to less errors. Short-term forecast was more accurate than long-term forecast, because more input meant more clues. However, when the historical range increased, the marginal benefit become less and less. Considering data availability and accuracy, n=5 and m=2 could be selected as the best combination.

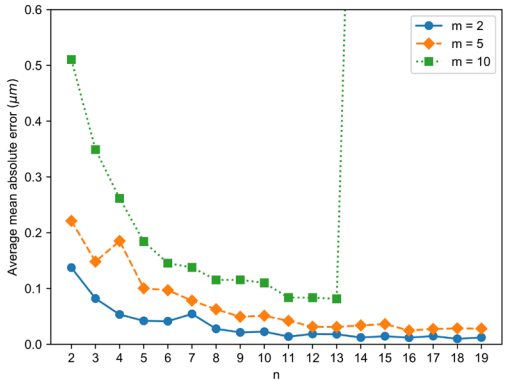


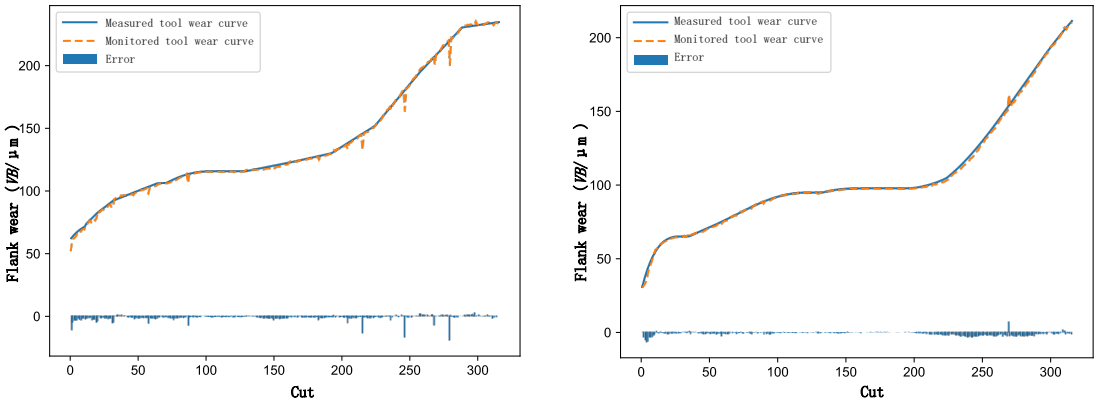
Fig. 6 A performance comparison under various time ranges

5.3 Performance evaluation of the in-process TCF approach

The ResNet-based in-process TCM model was implemented and integrated. The model with 20 residual blocks was selected due to its outstanding performance. As shown in Table 2, the averages of RMSE and MAE were 3.781 μm and 1.226 μm, respectively. As shown in Fig. 7, monitored *VB* values fitted nicely with measured tool wear curve. However, the monitored tool wear curves were not smooth non-decreasing, because random waves still existed.

Table 2 Accuracy evaluation of ResNet-based in-process TCM

Setup No.	S1	S2	S3	Average
RMSE (μm)	5.271	1.82	4.252	3.781
MAE (μm)	1.598	1.04	1.04	1.226

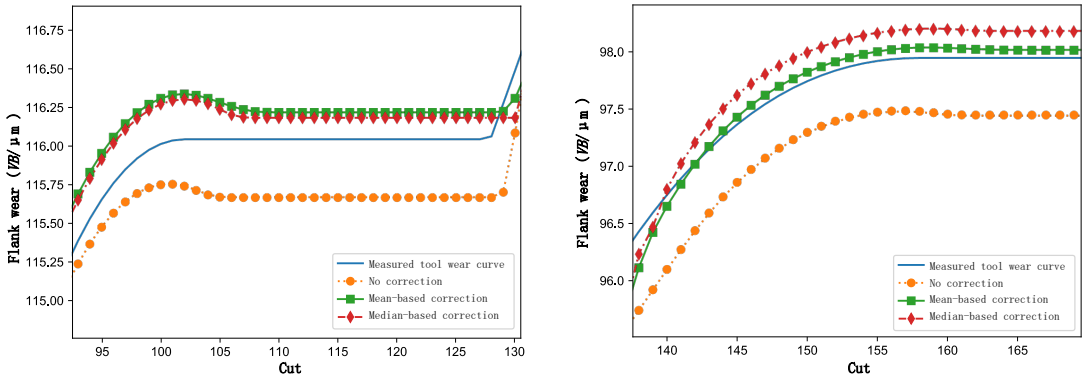


(a) Setup No. S1 (b) Setup No. S2

Fig. 7 ResNet-based TCM result

By integrating the ResNet-based in-process TCM model with the LSTM network, *VB* values in the nearest future could be forecasted when the machining process was going on. As shown in Fig. 8, the forecasted *VB* curves waved greatly. The LSTM network was not as accurate as it worked individually. Both median-based correction and mean-based correction contributed to

accuracy improvement of forecasted tool wear curves. By removing singular points or scatter points, waves in the forecasted cutting tool wear curve were reduced. According to Table 3, before correction, the average MAE value was 1.331 μm. After correction, the average MAE values were 1.253 μm for the median-based correction and 1.288 μm for the mean-based correction, respectively.



(a) Setup No. S1 (b) Setup No. S2

Fig.8 In-process TCF results before and after correction

Table 3 MAE comparison (μm)

Setup No.	S1	S2	S3	Average
No correction	2.607	0.946	0.440	1.331
Median-based correction	2.436	0.919	0.403	1.253
Mean-based correction	2.588	0.863	0.413	1.288

5.4 Comparisons and discussions

Comparisons have been conducted against some significant works in this field. Martinov et al. [20] approximated the cutting tool wear curve by a straight line. However, future cutting tool wear curve was regarded as a linear process without considering dynamics and time-varying characteristics. Because weak data dependencies in time series were captured and used, the long-term forecast might come with big errors. The ARMA with exogenous model developed by Pang et al. [21] predicted future *VB* values by using some dominant features extracted from machining signals. However, sensitive features were difficult to identify and select. Compare with these two works, the proposed approach was designed to forecast the non-linear cutting tool wear curves. By using median-based or mean-based correction, the waves in the forecasted cutting tool wear curves have been reduced dramatically. Regarding the time-varying cutting tool wear curves, the proposed approach was more accurate.

Wang et al. proposed a deep heterogeneous GRU model with an intermediate layer for tool wear prediction [32] [33]. However, the model was designed for single-step forecasting, while this work has implemented multi-step forecasting. No correction was made by the model proposed by Wang et al. Based on the same dataset (PHM 2010) [37] and time range (from 0 to 270 cuts) [32], more quantitative comparisons have been made by using MAE and RMSE. As shown in Table 4, the approach proposed in this paper outperformed the model proposed by

Wang et al. [32]. Under every setup, both MAE values and RMSE values have been reduced dramatically.

Table 4 Performance comparisons

Setup No.	S1		S2		S3	
Measurement (μm)	MAE	RMSE	MAE	RMSE	MAE	RMSE
5 steps-ahead prediction [32]	3.70	7.25	13.23	16.24	27.96	24.89
Proposed approach (Median-based correction)	0.994	1.337	1.349	1.995	0.838	1.444
Proposed approach (Mean-based correction)	1.233	1.696	1.294	1.935	0.846	1.381

Normally, a typical tool wear curve includes initial wear, steady wear and accelerated wear stages. According to above experiments, the proposed approach performed consistently in various stages, except that errors were likely to rise at inflection points.

Moreover, additional efforts should be made to improve the reliability and robustness of the proposed approach. Except continuous cutting tool wear, cutting tool breakage and chipping are unexpected, discrete and random incidents, which present further challenges for future research.

6. Conclusions

In this paper, a DL-based in-process TCF approach has been proposed. Using the latest historical *VB* values, the LSTM network can forecast tool conditions. The ResNet-based TCM model is integrated to enable in-process TCF. According to the above discussions and comparisons, the following conclusions can be drawn.

- By capturing data dependencies in time series, the LSTM network deduces multiple future *VB* values based on several measured *VB* values in time series. It bridges the gap between historical and future cutting tool wear curves.
- The integration of ResNet-based TCM enables in-process TCF. Both mean-based correction and median-based correction improves the accuracy.
- Experimental studies support that the proposed approach can accurately forecast cutting tool wear curves. Comparisons showed that the proposed approach outperformed existing models.

However, there are limitations to the approach, which call for further research. For example, the model's performance largely depends on the availability of data. More experiments with various machining conditions should be conducted to test the approach's robustness and reliability.

Acknowledgement

The research is under the support of the National Natural Science Foundation of China (No. 51875475), the National Science and Technology Major Project (No. 2017-VII-0010-0104) and the Natural Science Basic Research Plan in Shaanxi Province of China (No. 2018ZDXM-GY-068). Authors hereby thank them for the financial supports.

Reference

[1] Y. Li, C. Liu, J. Hua, J. Gao, P. Maropoulos, A novel method for accurately monitoring and predicting tool wear under varying cutting conditions based on meta-learning, CIRP Annals - Manufacturing Technology 68 (2019) 487-490.
[2] E. Kannatey-Asibu, J. Yum, T. H. Kim, Monitoring tool wear using classifier fusion. Mechanical Systems & Signal Processing 85 (2017) 651-661.
[3] F. Tao, Q. Qi, Make more digital twins, Nature 573 (2019) 490-491.
[4] A. Kusiak, Smart manufacturing must embrace big data, 554 Nature (2017) 23-25.
[5] F. Tao, Q. Qi, A. Liu, A. Kusiak, Data-driven smart manufacturing, Journal of Manufacturing Systems 48 (2018) 157-169.
[6] S. Ren, Y. Zhang, Y. Liu, T. Sakao, D. Huisingh, C. Almeida, A comprehensive review of big data analytics throughout product lifecycle to support sustainable smart manufacturing: a framework, challenges and future research directions, Journal of Cleaner Production 210 (2019) 1343-1365.
[7] J. Wang, Y. Ma, L. Zhang, R. X. Gao, D. Wu, Deep learning for smart manufacturing: Methods and applications, Journal of Manufacturing Systems 48 (2018) 144-156.
[8] Y. Zhou, W. Xue, Review of tool condition monitoring methods in milling processes. International Journal of Advanced Manufacturing Technology 96(5-8) (2018) 2509-2523.
[9] H.B. Sun, W.L. Niu, Hilbert-Huang transform based tool wear feature extraction. in: Proceedings of the 24th International Conference on Flexible Automation and Intelligent Manufacturing, San Antonio, USA, 20-23 May 2014, pp. 657-662.
[10] H.Y. Zhang, C. Zhang, J.L. Zhang, L.S. Zhou, Tool wear model based on least squares support vector machines and Kalman filter, Production Engineering 8 (2014) 101-109.
[11] G.F. Wang, Z.W. Guo, L. Qian, Tool wear prediction considering uncovered data based on partial least square regression, Journal of Mechanical Science and Technology 28(1) 2014 317-322.
[12] Z.R. Liao, D. Gao, Y. Lu, Z.K. Lv, Multi-scale hybrid HMM for tool wear condition monitoring, International Journal of Advanced Manufacturing Technology 84 (9-12) 2016 2437-2448.
[13] B. Cuka, D. W. Kim, Fuzzy logic based tool condition monitoring for end-milling, Robotics and Computer Integrated Manufacturing 47(10) (2017) 22-36.
[14] D.M. D'Addona, R. Teti, Image Data Processing via Neural Networks for Tool Wear Prediction, Procedia CIRP 12 (2013) 252-257.
[15] G.F. Zhang, S. To, G.B. Xiao, Novel tool wear monitoring method in ultra-precision raster milling using cutting chips, Precision Engineering 38 (2014) 555-560.
[16] W. Liu, C. Kong, Q. Niu, J. Jiang, X. Zhou, A method of NC machine tools intelligent monitoring system in smart factories, Robotics and Computer Integrated Manufacturing 61 (2020) 101842.
[17] R. G. Lins, P. R. M. Araujo, M. Corazzim, In-process machine vision monitoring of tool wear for Cyber-Physical Production Systems, Robotics and Computer Integrated Manufacturing 61 (2020) 101859.
[18] P. Wang, Z. Liu, R. X. Gao, Y. Guo, Heterogeneous data-driven hybrid machine learning for tool condition prognosis, CIRP Annals - Manufacturing Technology 68 (2019) 455-458.

[19] W. Wang, Y. Zhang, R. Y. Zhong, A proactive material handling method for CPS enabled shop-floor, Robotics and Computer Integrated Manufacturing 61 (2020) 101849.
[20] G. M. Martinov, A. S. Grigoryev, P. A. Nikishechkin, Real-Time Diagnosis and Forecasting Algorithms of the Tool Wear in the CNC Systems, in: Proceedings of the 6th International Conference in Swarm Intelligence, Beijing, China, 25-28 June 2015, pp. 115-126.
[21] C. K. Pang, J. H. Zhou, Z. W Zhong, F. L. Lewis, Tool wear forecast using Dominant Feature Identification of acoustic emissions, in: Proceedings of 2010 IEEE International Conference on Control Applications, Yokohama, Japan, 8-10 September 2010, pp. 1063-1068.
[22] R. Gao, L. Wang, R. Teti, D. Dornfeld, S. Kumara, M. Mori, M. Helu, Cloud-enabled prognosis for manufacturing, CIRP Annals - Manufacturing Technology 64 (2015) 749-72.
[23] R. Zhao, R.Yan, Z. Chen, K. Mao, P. Wang, R. X. Gao, Deep learning and its applications to machine health monitoring, Mechanical Systems and Signal Processing 115 (2019) 213-237.
[24] G.S. Babu, P. Zhao, X. L. Li, Deep Convolutional Neural Network Based Regression Approach for Estimation of Remaining Useful Life, in: S.B. Navathe et al. (Eds.), International Conference on Database Systems for Advanced Applications, Dallas, USA, 16-19 April 2016, pp. 214-228.
[25] J. Deutsch, D. He, Using Deep Learning-Based Approach to Predict Remaining Useful Life of Rotating Components, IEEE Transactions on Systems Man & Cybernetics Systems 99 (2017) 1-10.
[26] L. Ren, J. Cui, Y.Q. Sun, X.J. Cheng, Multi-bearing remaining useful life collaborative prediction: A deep learning approach, Journal of Manufacturing Systems, DOI: 10.1016/j.jmsy.2017.02.013
[27] X. Li, Q. Ding, J. Q. Sun, Remaining useful life estimation in prognostics using deep convolution neural networks, Reliability Engineering & System Safe 172 (2018) 1-11
[28] J. Schmidhuber, A local learning algorithm for dynamic feedforward and recurrent networks, Connection Science 1(4) (1989) 403-412.
[29] S. Hochreiter and J. Schmidhuber, Long short-term memory, Neural computation 9(8) (1997) 1735-1780.
[30] Z. Chen, Y. Liu, S. Liu, Mechanical state prediction based on LSTM neural network, in: Proceedings of the 36th Chinese control conference, Dalian, China, 26-28 June 2017, pp. 3876-3881.
[31] J. Zhang. P. Wang, R. Yan, R.X. Gao, Long short-term memory for machine remaining life prediction. Journal of Manufacturing Systems, 48 (2018) 78-86.
[32] J. Wang, J. Yan, C. Li, R. X. Gao, R. Zhao, Deep heterogeneous GRU model for predictive analytics in smart manufacturing: Application to tool wear prediction, Computers in Industry 111 (2019) 1-14.
[33] Q. Qiao, J. Wang, L. Ye, R. X. Gao, Digital Twin for Machining Tool Condition Prediction, Procedia CIRP 81 (2019) 1388-1393.
[34] X. Fang, Z. Yuan, Performance enhancing techniques for deep learning models in time series forecasting, Engineering Applications of Artificial Intelligence 85 (2019) 533-542.
[35] K.M. He, X.Y. Zhang, S.Q. Ren, J. Sun, Deep residual learning for image recognition, in: Proc. 29th IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, USA, 26 June - 1 July 2016, pp.770-778.
[36] K.M. He, X.Y. Zhang, S.Q. Ren, J. Sun, Identity mappings in deep residual networks, in: Proceedings of the European Conference on Computer Vision, Amsterdam, Netherland, 8-16 October 2016, pp. 630-645
[37] PHM Society, 2010 PHM Society Conference Data Challenge, https://www.phmsociety.org/competition/phm/10, 2010 (accessed 20 December 2018).