



Chatter detection in high-speed milling processes based on ON-LSTM and PBT

Fei Shi¹ · Hongrui Cao¹ · Yuke Wang² · Boyuan Feng² · Yufei Ding²

Received: 5 June 2020 / Accepted: 19 October 2020 / Published online: 10 November 2020
© Springer-Verlag London Ltd., part of Springer Nature 2020

Abstract

Chatter is a kind of self-excited vibration which frequently occurs in high-speed milling processes, which induces severe damage to both spindle tools and workpieces. In this paper, we introduce a new chatter detection technique using ordered-neurons long short-term memory (ON-LSTM) and population based training (PBT). First, we conduct a large number of milling experiments on a computer numerical control (CNC) milling machine with 4 accelerometers to get the dataset and employ vanilla LSTM for chatter detection. Then, to interpret the performance on time series of recurrent neural networks (RNN), a variation of LSTM named ON-LSTM is applied to chatter detection and a hyperparameter tuning method PBT is used for training. Finally, we compare the trained ON-LSTM with the time-frequency spectrum of the original signals obtained by short-time Fourier transform (STFT), and they show a certain degree of consistency.

Keywords Chatter detection · High-speed milling · ON-LSTM

1 Introduction

High-speed milling has been widely used in many manufacturing fields due to its high efficiency and low heat generation. However, a kind of fault named chatter appears frequently because of self-excited vibration. Chatter occurrence in a machining process has several severe

adverse effects, such as poor resultant surface quality, unacceptable inaccuracy, excessive noise, disproportionate tool wear, and machine tool damage [1]. To eliminate the damage chatter causes, scholars all over the world are developing researches from 3 main aspects [2], which are the analytical study of chatter stability [3], chatter detection [4], and online active control [5].

Due to the tight coupling and highly time-varying properties of the spindle system, engineers cannot guarantee the accuracy of analytical studies, and chatter may still occur within the stable zone of the stability lobe diagram [6]. Online active control is an excellent solution for chatter, which can eliminate the relative vibration between the cutting tool and the workpiece by an external force and works only after chatter occurrence, and some scholars also utilize dynamic vibration absorber to control the vibration [7]. However, even if we want to utilize active control techniques, chatter detection techniques should also be applied first to monitor the current condition. Compared with analytical studies, chatter detection techniques work regardless of the components' coupling and parameter identification of the spindle system, as well as the time-varying properties. It is a more generative and efficient way of chatter elimination in the industry.

The last two decades witness considerable growth in chatter detection techniques based on various kinds of signals such as acceleration [8], cutting force [9], and

✉ Hongrui Cao
chr@mail.xjtu.edu.cn

Fei Shi
shifeis@stu.xjtu.edu.cn

Yuke Wang
yuke.wang@umail.ucsb.edu

Boyuan Feng
boyuan@ucsb.edu

Yufei Ding
yufeiding@cs.ucsb.edu

¹ Mechanical Engineering, Xi'an Jiaotong University, Xi'an, China

² Computer Science, University of California, Santa Barbara, Santa Barbara, CA, USA

acoustic signals [10]. Acceleration is the most widely used signal in chatter detection because of its high reliability and low economic cost. Dynamometers are quite expensive, while noises can easily influence acoustic signals. Therefore, we also employ acceleration in our experiments.

Generally, chatter detection techniques can be divided into 3 different groups according to algorithms they rely on and the way they judge chatter occurrence. The first type is a classical one and based on frequency domain signal processing techniques such as the wavelet transform (WT) [11], S-function transform [12], adaptive filter [13], coherence function [14] and ensemble empirical mode decomposition (EEMD) [15]. Unfortunately, frequency domain based methods suffer poor resolution at the edge of time axis where data are most recent and crucial, resulting in poor performance in real-time chatter detection. The second type is derived from the statistics theory, where entropy methods such as the permutation entropy [16], coarse-grained entropy rate [17], and approximate entropy [18] try interpreting chatter from randomness point of view. However, since weights for frequency bands and the threshold for existence of chatter are empirical parameters, entropy-based approaches are still facing challenges in industrial applications. The third type is based on pattern classification algorithms such as artificial neural network [19, 20], fuzzy logic charts [21], and other machine learning methods. These pattern classification algorithms have been deployed in various machining processes, and they become more potent as a result of the fast development of deep learning.

As far as the authors know, the machine learning techniques applied to chatter detection in milling are still traditional ones, while deep learning methods have not been introduced into this field yet. The main difference between deep learning methods and traditional machine learning methods is feature extraction. In previous chatter detection work, the sensitive features have to be found manually, which is a huge problem. The manually found features may not be sensitive to chatter, and the extracting process requires amounts of experience in both metal cutting and signal processing. What's more, sometimes, these features only function well in certain circumstances and lack of generality. The development of deep learning these years provides an alternative of extracting features by the neural network itself automatically.

The detection of chatter can be seen as a time series classification task in which recurrent neural network (RNN) has an advantage. Long short-term memory (LSTM) is a RNN architecture first proposed in 1997 by Sepp Hochreiter [22], and the LSTM setup mostly used nowadays is

described by Graves and Schmidhuber [23] referred as vanilla LSTM [24]. LSTM is an effective and scalable model for several learning problems related to sequential data as handwriting recognition [25] and generation [26], language modeling [27], and translation [28], speech synthesis [29].

Although LSTM is great in sequence tasks and allows different neurons to track information at different time scales, it does not have an explicit bias towards modeling a hierarchy of constituents. In other words, LSTM may have acceptable accuracy in chatter detection, but it cannot reveal the latent structure of time series, which causes less interpretability. Recently, Yikang Shen proposed a variant of LSTM named ordered-neurons long short-term memory [30] (ON-LSTM), which introduces a new inductive bias for RNN. In this way, the neural network gains the ability to perform tree-like composition operations and may hold hierarchical information of vibration signals.

In this work, a large number of cutting experiments are conducted under different cutting parameters to obtain the dataset involving signals in both normal and chatter conditions. Within each cutting, the cutting depth increases smoothly from 0 to 10 mm so that the chatter signals can be obtained at their very beginning. First, vanilla LSTMs with different dimensions are employed for chatter detection, and the classification results are compared to the photos of the resultant surface of workpieces. Then, to reveal the latent structure of vibration signals, ON-LSTM is trained with a novel hyperparameter tuning method named population based training (PBT) [31]. Finally, the hierarchical representation of the vibration signal is captured by two newly designed gates in ON-LSTM, master input and forget gate, and the learned hierarchical information is compared to the frequency spectrum to demonstrate the combination between them.

2 Chatter detection methodology based on LSTM and ON-LSTM

In this section, we first bring some brief introductions to the employed techniques, which are LSTM, ON-LSTM, and PBT. Then we present two chatter detection techniques based on LSTM and ON-LSTM separately. As LSTM has excellent performance on sequences tasks, firstly, we employ LSTM on chatter identification. Then, to explain why RNN performs great in chatter identification and to find the consistency between the trained network and vibration mechanism, we employ ON-LSTM to reveal the latent hierarchical structure. What's more, PBT is used to train ON-LSTM efficiently and adequately.

2.1 Background theories

2.1.1 A brief introduction of LSTM

Let x^t be the input vector at time t , N be the number of LSTM blocks and M be the dimension of input vector. $W_z, W_i, W_f, W_o \in \mathbb{R}^{N \times M}$ are defined as input weights, $R_z, R_i, R_f, R_o \in \mathbb{R}^{N \times N}$ are defined as recurrent weights, and $b_z, b_i, b_f, b_o \in \mathbb{R}^N$ are defined as bias weights. In this way, the forward pass process of a vanilla LSTM layer can be written as

$$z^t = g(W_z x^t + R_z y^{t-1} + b_z) \tag{1}$$

$$i^t = \sigma(W_i x^t + R_i y^{t-1} + b_i) \tag{2}$$

$$f^t = \sigma(W_f x^t + R_f y^{t-1} + b_f) \tag{3}$$

$$c^t = z^t \odot i^t + c^{t-1} \odot f^t \tag{4}$$

$$o^t = \sigma(W_o x^t + R_o y^{t-1} + b_o) \tag{5}$$

$$y^t = h(c^t) \odot o^t \tag{6}$$

where z^t is the block input, i^t is the input gate, f^t is the forget gate, c^t is the cell state, o^t is the output gate, and y^t is the block output. σ, g and h are pointwise nonlinear activation functions. The gate employs the logistic sigmoid $\sigma(x) = \frac{1}{1+e^{-x}}$ as activation function and the hyperbolic tangent $g(x) = h(x) = \tanh(x)$ is used as the block input and output activation function. Pointwise multiplication of two vectors is denoted by \odot .

2.1.2 A brief introduction of ON-LSTM

In ON-LSTM, to enforce the update order and realize a hierarchical structure, a new activation function is introduced as:

$$\widehat{g} = \text{cumax}(\dots) = \text{cumsum}(\text{softmax}(\dots)) \tag{7}$$

where cumsum denotes the cumulative sum. Based on the $\text{cumax}()$ function, a master forget gate f_m and a master input gate i_m are introduced as:

$$f_m^t = \text{cumax}(W_{f_m} x_t + R_{f_m} y_{t-1} + b_{f_m}) \tag{8}$$

$$i_m^t = 1 - \text{cumax}(W_{i_m} x_t + R_{i_m} y_{t-1} + b_{i_m}) \tag{9}$$

where $W_{f_m}, W_{i_m} \in \mathbb{R}^{N \times M}$ are input weights, $R_{f_m}, R_{i_m} \in \mathbb{R}^{N \times N}$ are recurrent weights and $b_{f_m}, b_{i_m} \in \mathbb{R}^N$ are bias weights.

According to the properties of the $\text{cumax}()$ activation, the values in master forget gate will increase from 0 to 1 while the values in master input gate will decrease from 1 to 0.

These two gates can realize high-level control for cell states update. Based on the employed two master gates, a new update rule for cell state will be:

$$\omega_t = f_m^t \odot i_m^t \tag{10}$$

$$f_m^{t'} = f_t \odot \omega_t + (f_m^t - \omega_t) = f_m^t \odot (f_t \odot i_m^t + 1 - i_m^t) \tag{11}$$

$$i_m^{t'} = i_t \odot \omega_t + (i_m^t - \omega_t) = i_m^t \odot (i_t \odot f_m^t + 1 - f_m^t) \tag{12}$$

$$c_t = z_t \odot i_m^{t'} + c_{t-1} \odot f_m^{t'} \tag{13}$$

ω_t is the product of the two master gates and represents the overlap of f_m^t and i_m^t . If some elements in ω_t are larger than 0, the corresponding neurons hold information which should be updated partially.

2.1.3 A brief introduction of PBT

The basic latent idea for neural networks is to optimize a group of parameters θ of a model f to maximize a predefined objective function Q and the trainable parameters θ are updated using an optimization procedure such as stochastic gradient descent. PBT is an effective way to optimize both the trainable parameters θ and the hyperparameters h jointly. To make it clear, a function $\text{eval}()$ is defined to evaluate the objective function Q based on the current trainable parameters θ . In this way, the process of finding the optimal set of parameters that maximize the objective function Q can be written as:

$$\theta^* = \arg \max_{\theta \in \Theta} \text{eval}(\theta) \tag{14}$$

The trainable parameters θ are updated in an iterative manner and in each step condition on the hyperparameters h . In more detail, the update process of the trainable parameters can be expressed as

$$\begin{aligned} \theta^* &= \text{optimise}(\theta \mid \mathbf{h}) \\ &= \text{optimise}(\theta \mid (h_t)_{t=1}^T) \\ &= \text{step}(\dots \text{step}(\theta \mid h_1) \dots \mid h_T) \end{aligned} \tag{15}$$

The solution of the parameters θ^* is typically sensitive to the choice of hyperparameters sequence $h = (h_t)_{t=1}^T$. Improperly chosen of a hyperparameters sequence will lead to bad solutions. In practice, all h_t are equal to each other or according to a simple predefined schedule, where search over multiple possible values of h is needed as shown in Eqs. 16 and 17.

$$\theta^* = \text{optimise}(\theta \mid h^*) \tag{16}$$

$$h^* = \arg \max_{h \in H^T} \text{eval}(\text{optimise}(\theta \mid h)) \tag{17}$$

In PBT, in order to perform such optimization efficiently, N models are trained at the same time to reach the goal.

These models hold different trainable parameters $\{\theta^i\}_{i=1}^N$ and hyperparameters $\{h^i\}_{i=1}^N$ forming a population P . Then, the objective turns to find the optimal model across the entire population.

2.2 Experiment configuration

To obtain a dataset of vibration signals for chatter detection tasks, a number of milling experiments have been conducted on a high-speed milling machine VMC-V5 as shown in Fig. 1.

Chatter develops extremely fast from its very beginning to severe vibration in a milling process. Therefore, it is necessary to identify chatter as early as possible to avoid the loss. Generally, two cutting parameters have a direct relationship with chatter: spindle rotating speed and cutting depth, and this relationship can be expressed by a stable lobe diagram, as shown in Fig. 2. In Fig. 2, the region above the curve is the unstable condition, while the region below is the stable region. From the vertical view, as the spindle speed keeps still, the cutting condition turns into chatter as the cutting depth increases.

To get the signals where chatter just occurs, we customize our workpieces with a slope. The workpiece material used in our experiment is a kind of high strength aluminum, 2A12, and it is widely used in aircraft structure, rivets, truck wheel, screw elements, and other various structures. In this way, the cutting depth increases linearly in a milling process, and the increase of cutting depth results in increasing of cutting force and vibration, which may cause chatter. It is known that when the cutting condition is at the border between unstable and stable regions, it is challenging to justify whether chatter occurs or not. By placing this slope, signals in the normal condition, the chatter condition and the border are all collected. Holding signals under all kinds of cutting



Fig. 1 High-speed milling machine VMC-V5

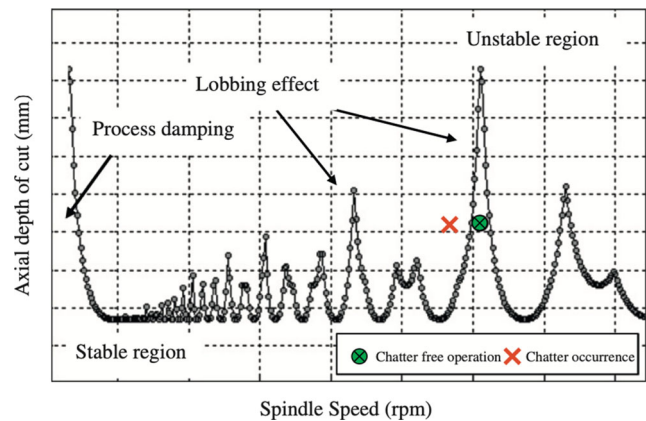


Fig. 2 Stability lobe diagram [1]

conditions is a precondition for classifying all signals into different categories accurately. Since we have plenty of signals at the border between unstable and stable regions in the training set, the trained neural network will have the ability of identifying border signals. To obtain signals under different situations, we carry out cutting experiments under different spindle speeds and different cutting width as listed in Table 1. Considering chatter conditions cause more damage, we replace our milling tool after every three chatter conditions. This can guarantee the milling tool in a relatively healthy state.

The spindle rotating speed is selected within the range from 6000 to 12,000 rpm. Cutting with a spindle rotating speed below 6000 rpm may result in the process damping zone, which we do not have an interest in. The highest rotating speed is 12,000 rpm since the vibration and noise of

Table 1 Cutting parameters in high-speed milling experiments

Number	Rotating speed	Cutting width
1–15	6000	1.5, 1.7, 1.9, 2.1, 2.3, 2.5, 2.7*, 2.8, 3, 3.3, 3.4*, 3.5*, 3.6*, 3.7*, 3.8*
16–23	7000	1.8, 2.4, 2.9, 3*, 3.1, 3.2*, 3.3, 3.4*
24–30	8000	2.8, 3, 3.1*, 3.2, 3.3*, 3.4*, 3.5*
31–38	9000	2.8, 3*, 3.1, 3.2, 3.3*, 3.4, 3.5*, 3.6
39–49	10,000	1.8, 1.9, 2*, 2.1*, 2.2, 2.3*, 2.4, 2.5*, 2.6*, 2.7*, 2.8*
50–59	11,000	1.8, 1.9, 2, 2.1, 2.2*, 2.3*, 2.4*, 2.5*, 2.6*, 2.7
60–66	12,000	1.8, 1.9*, 2*, 2.1*, 2.2, 2.3*, 2.4*

chatter are extremely severe, and we are afraid of accidents happening at a larger rotating speed. The cutting width with a superscript * means the cutting condition turns to chatter from normal in one milling process while the cutting width without a superscript corresponds to a complete stable cutting process. It is shown that some certain cutting width corresponds to chatter, but then a larger width corresponds to stable cutting. This is caused by another phenomenon called isolated islands, which is reported by B.R. Patel [32]. Generally, as cutting depth increases, the unstable zone will expand and some normal cutting scenes can turn into chatter ones. However, it can become different when isolated islands appear. The so-called isolated islands mean in the stability lobe diagram, besides intrinsic lobes, some small areas arise aside. With these islands, as the cutting depth increases, the cutting condition can change from chatter to normal and back to chatter again.

In our experiments, 4 accelerometers are placed on both the spindle and the workpiece. Two accelerometers (IMI 608A11, with a sensitivity of 100 mV/g) are placed on the spindle in both x - and y -directions. Two accelerometers (PCB 333B50, with a sensitivity of 1000 mV/g) are attached to the workpiece in both x - and y -directions. The data acquiring device is ECON AVANTMI-7008, with a sampling frequency of 24,000 Hz. The arrangement of sensors is shown in Fig. 3.

From Fig. 3, we can see the slope on the workpiece with a height of 10 mm, which means the cutting depth of each

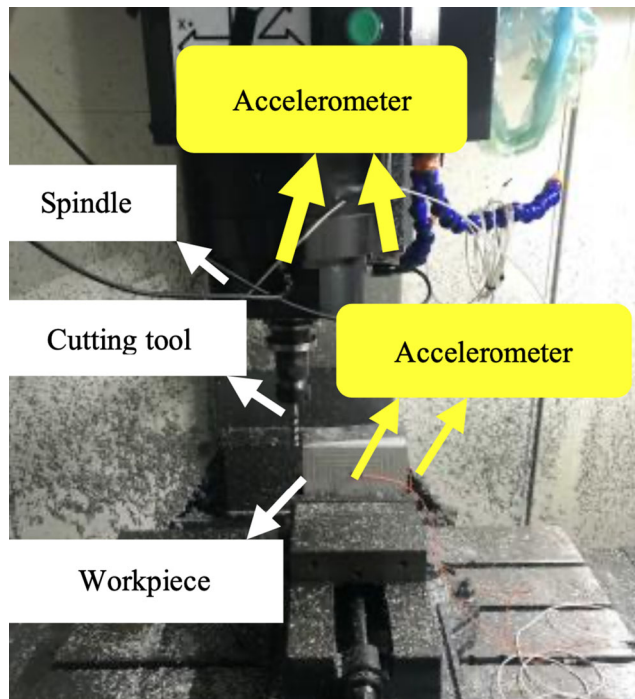


Fig. 3 The arrangement of 4 sensors in a CNC milling machine

milling process begins at 0 mm and ends up with 10 mm. The cutting tool is made of high-speed steel with 3 cutting edges and 60 mm overhang. The feed rate of the spindle is kept as 400 mm/min. Among all the cutting experiments in Table 1, there is a total of 32 groups of stable milling processes and 34 groups of milling processes ending up with chatter.

2.3 LSTM based training and test

Recurrent neural network with LSTM appears to be one of the most effective and scalable models for the majority of learning problems related to sequential data. Graves and Schmidhuber [23] originally described the most widely used LSTM setup in the industry referred to as vanilla LSTM, which consists of three gates defining the stream of historical information, one cell representing the current state and an output activation function. The LSTM block's output is recurrently connected back to the input of block in the next iteration.

First, we need to label the signals before putting them into training. According to both the frequency spectrum and the resultant surface, signals are divided into normal cutting parts and chatter parts. In a stable cutting process, all signals are classified into normal parts. In an unstable cutting process, chatter start time is defined according to the change of frequency spectrum and the roughness of resultant surface. Signals before chatter start time is classified as normal parts while signals after chatter start time are put into chatter parts.

Besides labels, the sequence length should also be restricted before put into neural networks. In our experiments, the sequence length is fixed as 250 and this value is based on the spindle rotating speed and the sampling frequency. The maximum and the minimum value of spindle rotating speed is 6000 rpm and 12,000 rpm, and the sampling frequency is 24,000 Hz, which means the maximum number of sampled points within one rotation is 240 and the minimum number is 120. In this way, the selected sequence length covers at least one complete spindle rotation time and we can realize chatter identification by a relatively short time series.

The employed structure of LSTM for chatter identification is shown in Fig. 4. As four sensors are used in experiments, the dimension of input is 250×4 . The first layer is LSTM layer with a dimension of N and input is fed into it successively. The last hidden state of LSTM is collected as the input of second layer, which is a fully connected layer with 10 cells and the activation function is RELU. The learned feature with a dimension of 10 is fed into the final layer which is also a fully connected layer which has only one output with a sigmoid function. In this way, the probability of chatter occurrence is obtained.

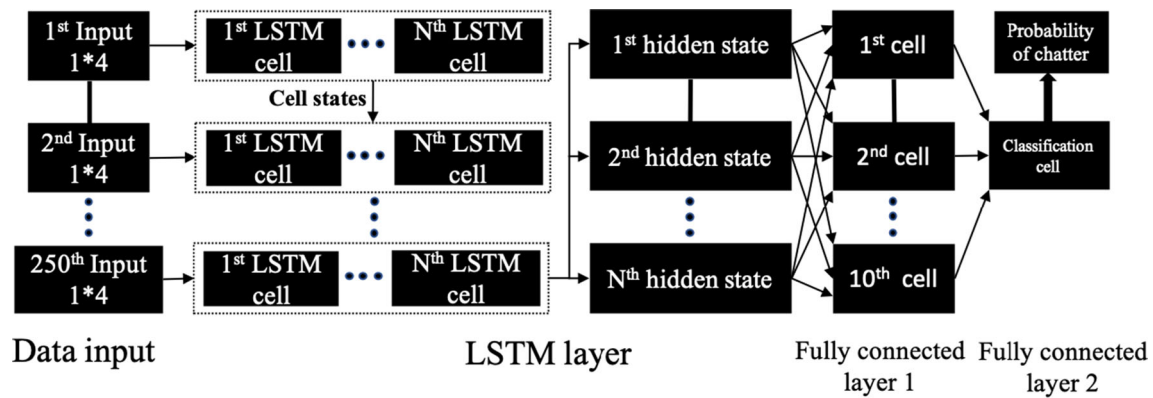


Fig. 4 Network structure for chatter identification

2.4 ON-LSTM and PBT based chatter detection methodology

Although LSTM has a great performance on chatter identification, we have no idea why it works so well. To dive in the work mechanism of recurrent neural network, ON-LSTM is employed. In ON-LSTM, two new gates, master forget gate and master input gate are introduced into LSTM networks to reveal the latent structure within time series. The neurons will be updated according to the overlap vector, original forget gate and original input gate. For neurons with higher rank than the overlap ones, the information will be held still and no input will affect them. On the contrast, the information in the neurons which have lower rank than the overlap ones will be completely replaced by the input information. The hierarchical structure within ON-LSTM lies on the control of the update frequency. If one wants to erase the information in high ranking neurons, the information in all lower-ranking neurons should be erased first.

However, new gates also introduce new network parameters which still need to be trained which may cause difficulty in the convergence of network training. Therefore, we employ PBT for model training. The goal is to optimize the learning rate, which is one of the critical hyperparameters during a model training process and we make a little change from the original PBT to reduce training time. The complete process for chatter identification based on ON-LSTM and PBT is shown in Fig. 5.

The whole network structure we employed is similar with that in LSTM where the only difference is to replace LSTM cell by ON-LSTM cell. In ON-LSTM training, an improper value of learning rate may lead to extremely long training time. Therefore, to train the ON-LSTM network efficiently, PBT is used where several workers with different initial learning rates are employed into ON-LSTM training, which forms a so called population.

To adjust the value of hyperparameters according to the entire population, two strategies from the original PBT, *exploit* and *explore*, are employed.

The goal of *exploit* is to desert the worst performed worker, which holds inappropriate hyperparameter value, so there is no need to continue training with current hyperparameter. Therefore, in the *exploit* process, we find the worker with largest training loss and replace its hyperparameter value by that in the worker with lowest training loss. In this way, we save training time by transform the unpromising worker into the promising one and can find out more optimal hyperparameter value.

The goal of *explore* is to extend the searching space for current hyperparameter value. There is high probability that current hyperparameter value is not the optimal one, so it is rational to try new value around the current one. The *explore* process can find new hyperparameter value to better explore the solution space given the current solution, this new value should not be so far away from the current one, because the current one already has acceptable performance and stability of the algorithm should be guaranteed.

The combination of multiple steps of gradient descent using *exploit* and *explore* results in hyperparameter copy and perturbation. The learning algorithms can benefit from not only local optimization by gradient descent, but also periodic model selection and hyperparameter refinement.

Besides these two strategies, we also introduce another simple strategy and name it *explode* to make it harmony with the aforementioned strategies. The goal of *explode* is to fire several workers to save training time. It is a straightforward trick and in our training the *explode* process fires bad performance workers after a preset times of *exploit* and *explore*. A complete training process is shown in Table 2 and a flow chart is shown in Fig. 6.

Being trained based on PBT, the network will converge quickly and finally only one worker will be selected as the working model. In our case, the hyperparameters are

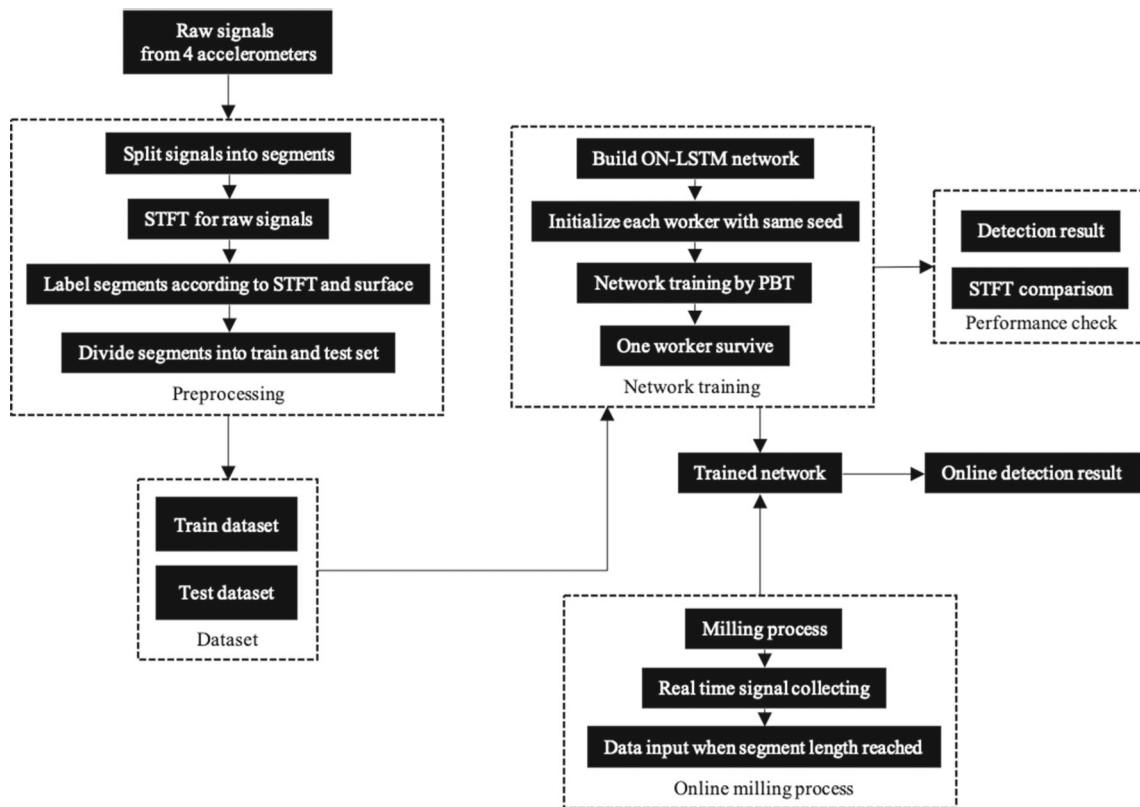


Fig. 5 The complete process for chatter identification based on ON-LSTM and PBT

restricted to learning rate, and ten workers are employed as the population which are initialized as 0.0001, 0.0003, 0.0006, 0.001, 0.003, 0.006, 0.01, 0.03, 0.06, and 0.1 respectively in line 1 of Table 2. If there are more than one worker in population, we run the circulation from line 3 to line 10. In line 3, we define the times of training in each explode process as 20, which means we perform explode operation each 20 epochs of training. From line 4

to line 6, we optimize the parameters of network in each worker by Adam and the corresponding learning rate in each worker for one epoch, and calculate the training loss for each worker. Then in line 7, the learning rate of the worker having best performance will replace that of the worker having worst performance. Next in lines 8–9, the learning rate in each worker except the replaced one will change to a new value according to Gauss distribution, the center of which is the current learning rate and variance is ten percent of the current learning rate. Then in line 10, after specified times of epochs, half of the worker holding worse performance will be deserted (round down if not integer). The training lasts until there is only one worker left and the trained parameters of this worker are returned.

The next thing we want to know is that how the network learns the knowledge from the original signals. To infer the latent structure of one segment of time series, at each time step, we compute an estimate of master forget gate value \hat{d}_f^t and master input gate value \hat{d}_i^t :

$$\begin{aligned} \hat{d}_f^t &= \mathbb{E}[d_f^t] = \sum_{k=1}^{D_m} k p_f(d_t = k) \\ &= D_m - \sum_{k=1}^{D_m} f_m^{tk} \end{aligned} \tag{18}$$

Table 2 A complete training process by population based training

A training method by modified PBT

- 1: Initialize workers with different hyperparameters
- 2: While more than one worker in population:
 - 3: For each train in training times in a explode process:
 - 4: For each worker in population:
 - 5: Optimize parameters with current hyperparameters
 - 6: Calculate current performance
 - 7: Exploit
 - 8: For each worker in population and not in exploit:
 - 9: Explore
 - 10: Explode
 - 11: Return the surviving worker and its trained parameters

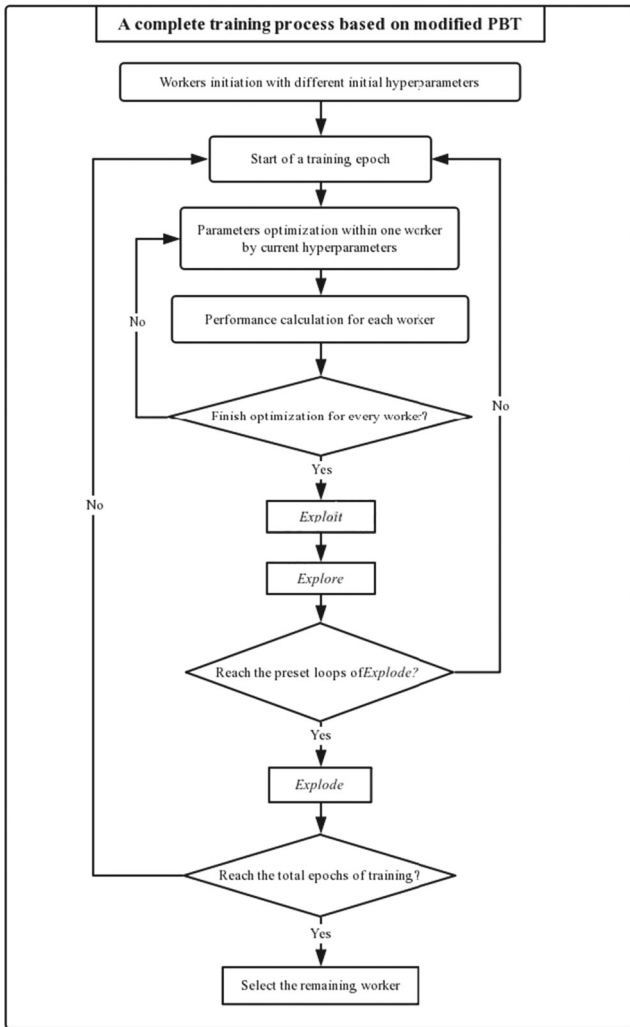


Fig. 6 A complete training process based on modified PBT

$$\hat{d}_i^t = \mathbb{E}[d_i^t] = \sum_{k=1}^{D_m} k p_i(d_i = k) = D_m - \sum_{k=1}^{D_m} i_m^{tk} \quad (19)$$

where p_f is the probability distribution over split points associated to the master forget gate and D_m is the size of the hidden state.

In ON-LSTM, we use the master input gate and the master forget gate to sequence the cells, but we cannot obtain the current information level from these two gates directly. When we feed raw data into a trained neural network, we can get a series of number by the cumax function representing the level for every at each time node. Then we use a synthesis of forget gate, input gate and two master gates to control the information flow of both input and existing information. The synthesis is denoted by i_m^t and f_m^t , and they have a dimension of cell numbers.

Since each cell represents an information level, we can use expectation of a cell representing current information level among all cells. In statistics, it is a sum of multiplications of current cell and its corresponding probability. Mathematically, it can be obtained by the number of cells subtracted by the sum of the synthesis. In this way, Eqs. 18 and 19 are obtained.

\hat{d}_f^t and \hat{d}_i^t are estimated hierarchical level at time t for master forget gate and master input gate separately. Large value of \hat{d}_f^t corresponds to a high forget level and large value of \hat{d}_i^t corresponds to a low input level. In this way, we can get two sequences representing the hierarchical level at each sampling time, which are $\hat{d}_f = (\hat{d}_f^1 \dots \hat{d}_f^L \dots)$ and $\hat{d}_i = (\hat{d}_i^1 \dots \hat{d}_i^L \dots)$, where L is the segment length. Then, these two sequences are transformed into time-frequency domain by

$$X_f(t, f) = \int_{-\infty}^{\infty} \omega(t - \tau) \hat{d}_f(\tau) e^{-j2\pi f \tau} d\tau \quad (20)$$

$$X_i(t, f) = \int_{-\infty}^{\infty} \omega(t - \tau) \hat{d}_i(\tau) e^{-j2\pi f \tau} d\tau \quad (21)$$

where $\omega()$ is the window function. The STFT results of the two master gates are compared to the STFT results of raw data to get interpretation of network working mechanism. What's more, to make it more clear, we also create a so called mixed master gate value, which represents the difference between master input gate and master forget gate.

3 Experiments and results

3.1 Dataset preparation

The raw signals collected from the 4 sensors in a milling process is shown in Fig. 7. From the top to the bottom, these four subplots correspond to acceleration of spindle at x -direction, acceleration of spindle at y -direction, acceleration of workpiece at x -direction, acceleration of work piece at y -direction. In Fig. 7, the acceleration of spindle keeps a relatively stable magnitude, because the spindle is always rotating at a specified speed even without cutting and the imbalance of the spindle itself causes some vibration. The vibration of the workpiece is caused by its movement with the holder and the cutting force, so the termination of cutting can be seen clearly from the two bottom subplots at around 18.2 s.

Among the researches of chatter identification, frequency domain methods are widely used to observe the onset of chatter. In our experiments, we obtain the start time of chatter based on both frequency spectrum and resultant surface of the workpiece. Short-time Fourier transform

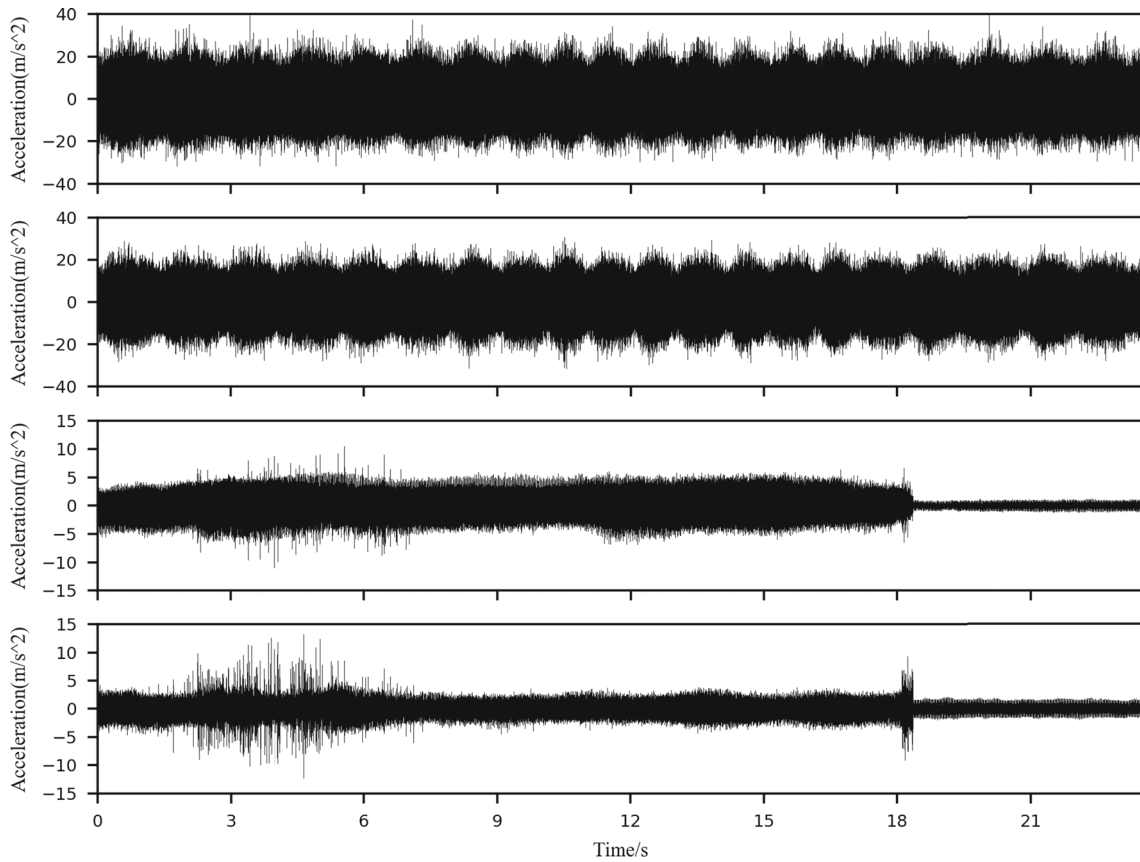


Fig. 7 Signals from the 4 sensors in a milling process (spindle rotating speed = 6000 rpm, cutting width = 1.5 mm)

(STFT) is employed to get the frequency spectrum of every milling process as shown in Fig. 8.

However, frequency domain based methods have two inherent flaws for online chatter monitoring which they cannot overcome. The first problem is the signal length and

resolution. For online chatter monitoring, the time series closest to current time is most valuable, but time-frequency methods need a relatively long time series for analysis and the most valuable fragment is squeezed to the very side. If so, the frequency resolution will become pretty poor at

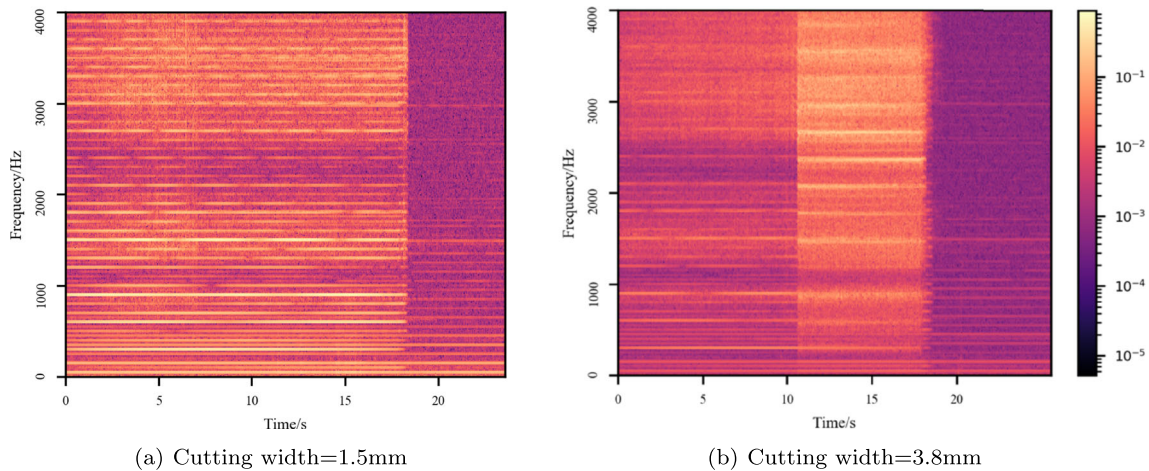


Fig. 8 Frequency spectrum of the workpiece at *x*-direction (spindle rotating speed = 6000 rpm)

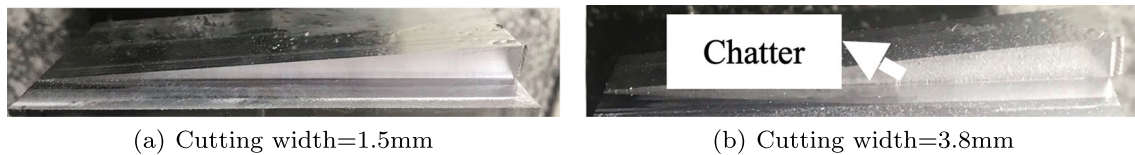


Fig. 9 Resultant surface of the workpiece with different cutting width

the chatter start time. The reason is time-frequency methods usually has poor frequency resolution at the edge of time axis, and this is disastrous for online chatter monitoring.

The second problem is the indicators. After transforming the signal into frequency domain, we still need to define the criteria of chatter occurrence. Sensible indicators and their threshold value need to be found in this procedure. This can be a hard task since specialized knowledge in both metal cutting and signal processing is needed. If the indicators are inappropriate, it will increase the difficulty of chatter identification. Sometimes, machine learning techniques are also employed for this pattern recognition task.

Deep learning based methods can handle such problems. The core strength of deep learning methods is that they can handle time series with very short length, because raw time series are fed into neural networks directly and they can deal with any length. In our experiments, the segment length is fixed as 250 while the sampling frequency is 24,000 Hz, so the proposed chatter identification method can accomplish its task within the sampling time 0.0104 s. As for the second problem, neural networks can take raw time series as input and output the classification results directly. Therefore, deep learning methods have the natural ability of finding the most sensitive indicators. Since deep learning based methods can handle the two weaknesses of the traditional ones, they are more promising and become more popular.

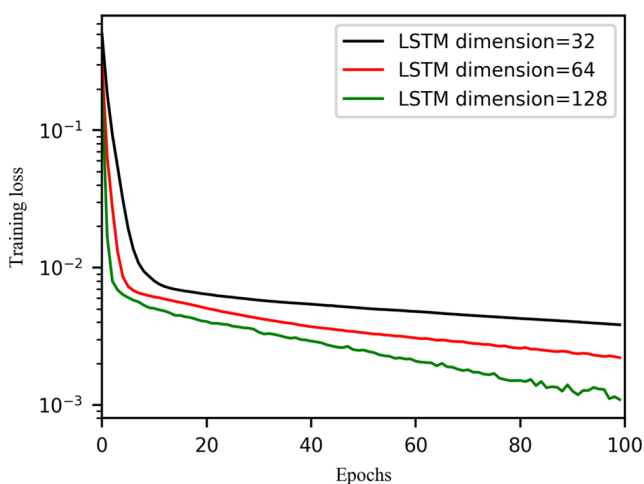


Fig. 10 Log training loss with different LSTM dimensions

The spindle rotating speed in Fig. 8 is 6000 rpm, which means the frequency of rotation is 100 Hz. Since the cutter we used has 3 edges, the cutting frequency will be 300 Hz. In Fig. 8a, the three frequency components with the lightest color are 300 Hz, 600 Hz, and 900 Hz, which are the cutting frequency and its harmonics. After metal cutting, these 3 frequency components disappear as expected. Figure 8b shows an unstable cutting situation, where after 10 s frequency spectrum becomes much more obscure than before. Several new frequency components appear which are called chatter frequency components. We can distinguish chatter scenes from normal scenes easily from Fig. 8 because we already have the whole-length signal. For online chatter detection, frequency domain based methods perform undesirable.

The resultant surface of the workpiece under different cutting width is shown in Fig. 9. The whole surface keeps smooth in normal cutting condition as shown in Fig. 9a. However, in Fig. 9b, as a result of the increase of cutting width, the cutting force and the vibration of the workpiece become severe. As cutting depth increases, the cutting condition changes from normal to chatter where the surface is extremely rough. While chatter, the severe vibration and the enormous heat generation lead to fatal damage to cutting tools, and life of the spindle system and even the whole machine tool will decrease a lot.

In our experiments, we only obtain signals from high-speed milling processes. Therefore, the trained neural network can only be applied into chatter identification in high-speed milling operations. We focus on high-speed milling because it is becoming a popular manufacturing method and chatter can easily occur during high-speed milling. If signals in normal milling processes are also fed into training dataset, the obtained neural network will have the ability of chatter identification for normal milling.

3.2 Chatter detection based on LSTM

We select two milling processes to test the performance of trained network which are No. 37 and No. 65 in Table 1, while the other cutting experiments are used as the training dataset. In this way, there are totally 116,555 normal sequences and 19,518 chatter sequences in our training dataset. LSTM layers with a dimension of 32, 64, and 128

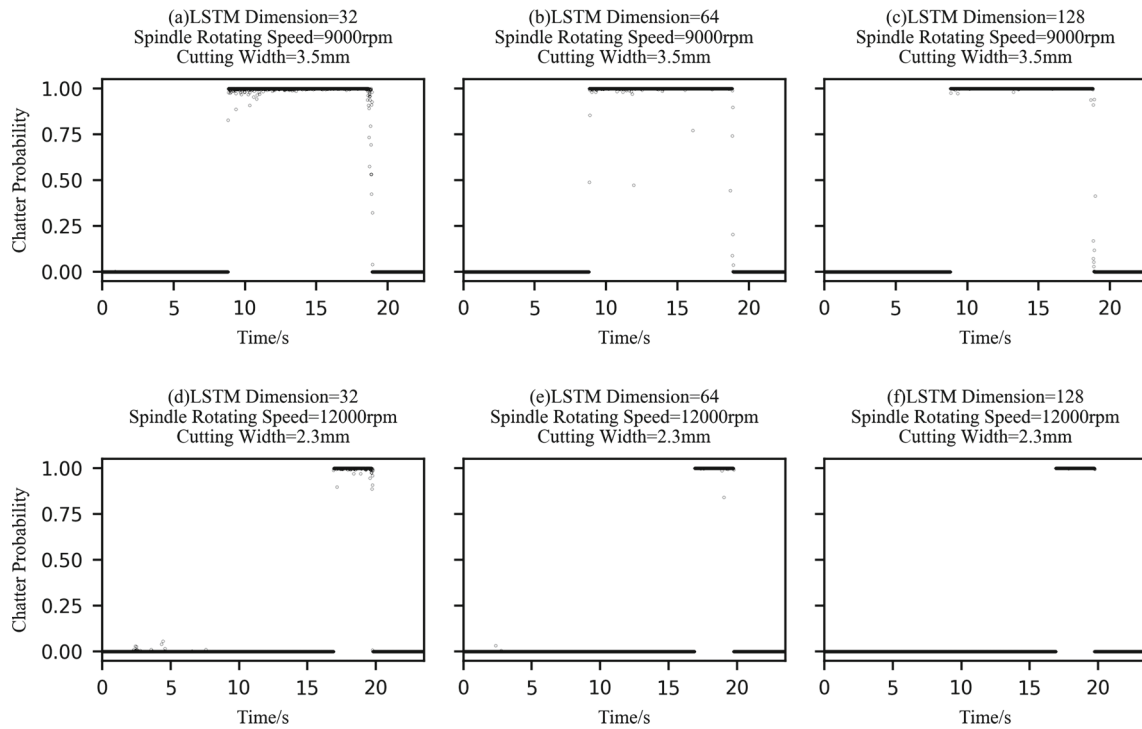


Fig. 11 Predicted chatter probability by LSTM

are all employed and the optimizer is Adam. The loss in training dataset in these three situations is shown in Fig. 10.

Training loss decreases dramatically in the first several epochs and becomes gentle then. When the dimension of LSTM gets larger, the convergence of network seems faster because of larger capacity, but there is a latent problem of overfit. The trained network is then utilized to predict the cutting condition for signals in the test set. The output of the last sigmoid layer is shown in Fig. 11.

Figure 11 shows the prediction result of the trained LSTM network. In the first row, the spindle rotating speed

is 9000 rpm and the cutting width is 3.5 mm. At the beginning of cutting, the probability of chatter remains 0 because the cutting process is under normal condition at first. Then at around 9 s, the probability of chatter increases up to 1 and the cutting condition turns to chatter. The probability of chatter is extremely close to either 0 or 1 at all time especially in Fig. 11b, which means the trained network classifies the signals definitely and LSTM has a

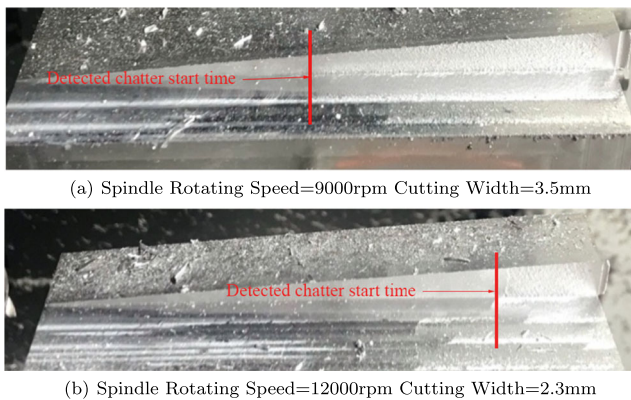


Fig. 12 Resultant surfaces of the test set

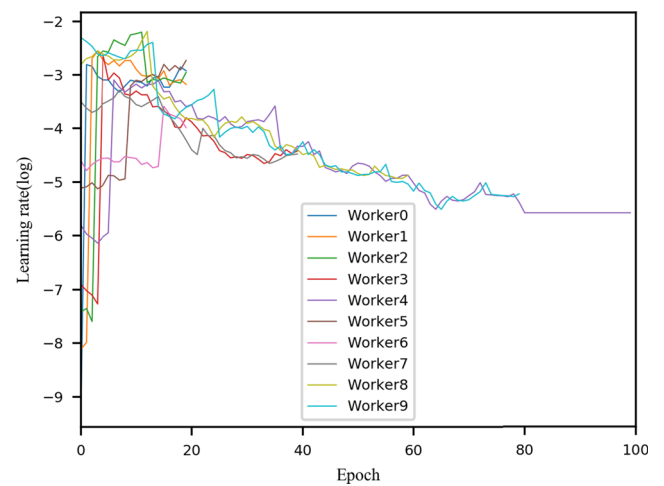


Fig. 13 Learning rate in a complete training process (ON-LSTM dimension = 32)

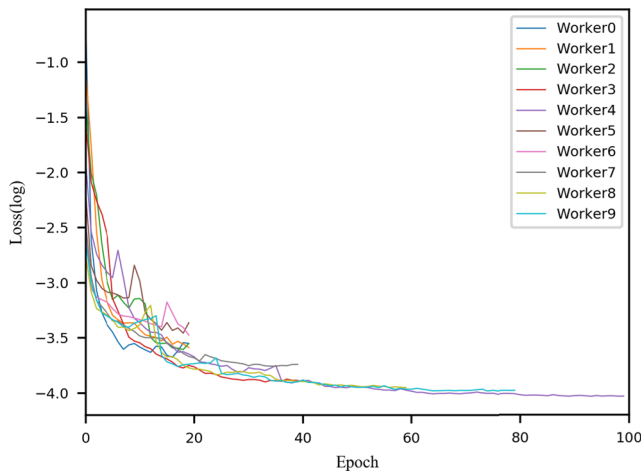


Fig. 14 Loss in a complete training process (ON-LSTM dimension = 32)

great performance in chatter detection task. Another thing is, LSTM network performs best when its dimension equals 64 since the value of chatter probability is closer either to 0 or 1. In this way, we can draw a conclusion that LSTM network with dimension of 32 is underfit while LSTM network with dimension of 128 is overfit.

Figure 12 shows the resultant surface of test set which holds the consistency with the predicted result. Although LSTM network performs good in chatter identification, it

works like a black box and we cannot tell why it performs good and how it analyzes signals.

3.3 Chatter detection based on ON-LSTM and PBT

In ON-LSTM training, we employ PBT to realize faster convergence of network parameters. Figure 13 shows the learning rate of each worker in a complete training process and Fig. 14 shows the training loss where the dimension of ON-LSTM layer is specified as 32 in both figures.

The three training strategies in PBT, *exploit*, *explore*, and *explode*, can be seen clearly from the plot. The y axis is expressed in log scale using natural logarithm since the learning rate in workers crosses a wide range especially at the initial state. In the first 20 epochs, workers with larger learning rates have better performance than those with smaller learning rates as shown in Fig. 14, so some workers with small learning rates change into large ones as shown in Fig. 13, which is called *exploit*. The reason is that at the beginning of training, large learning rates can help the network converge more quickly. After each epoch, every worker tries to find another learning rate near the current value except the changed worker. This operation is called *explore* and it can expand the learning rate range and benefit the training process. After 20 epochs of training, 5 workers with worse performance are deserted, which is called *explode*. Similarly, after 40 epochs 2 more workers

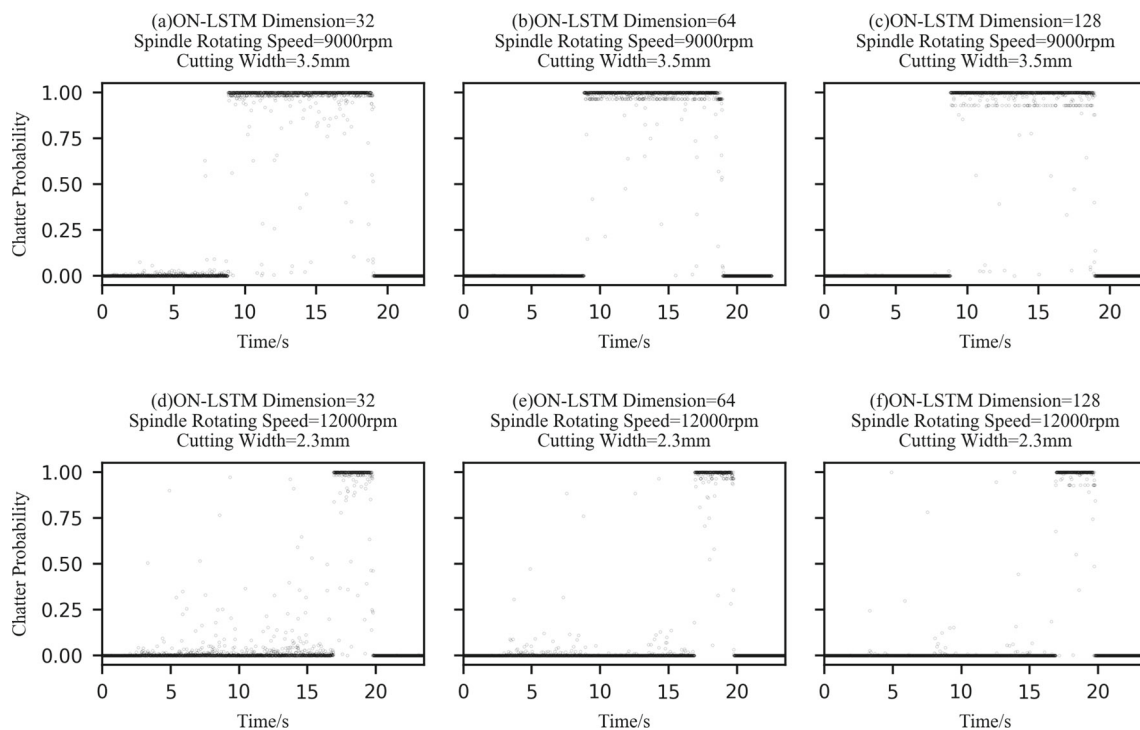


Fig. 15 Predicted chatter probability by ON-LSTM

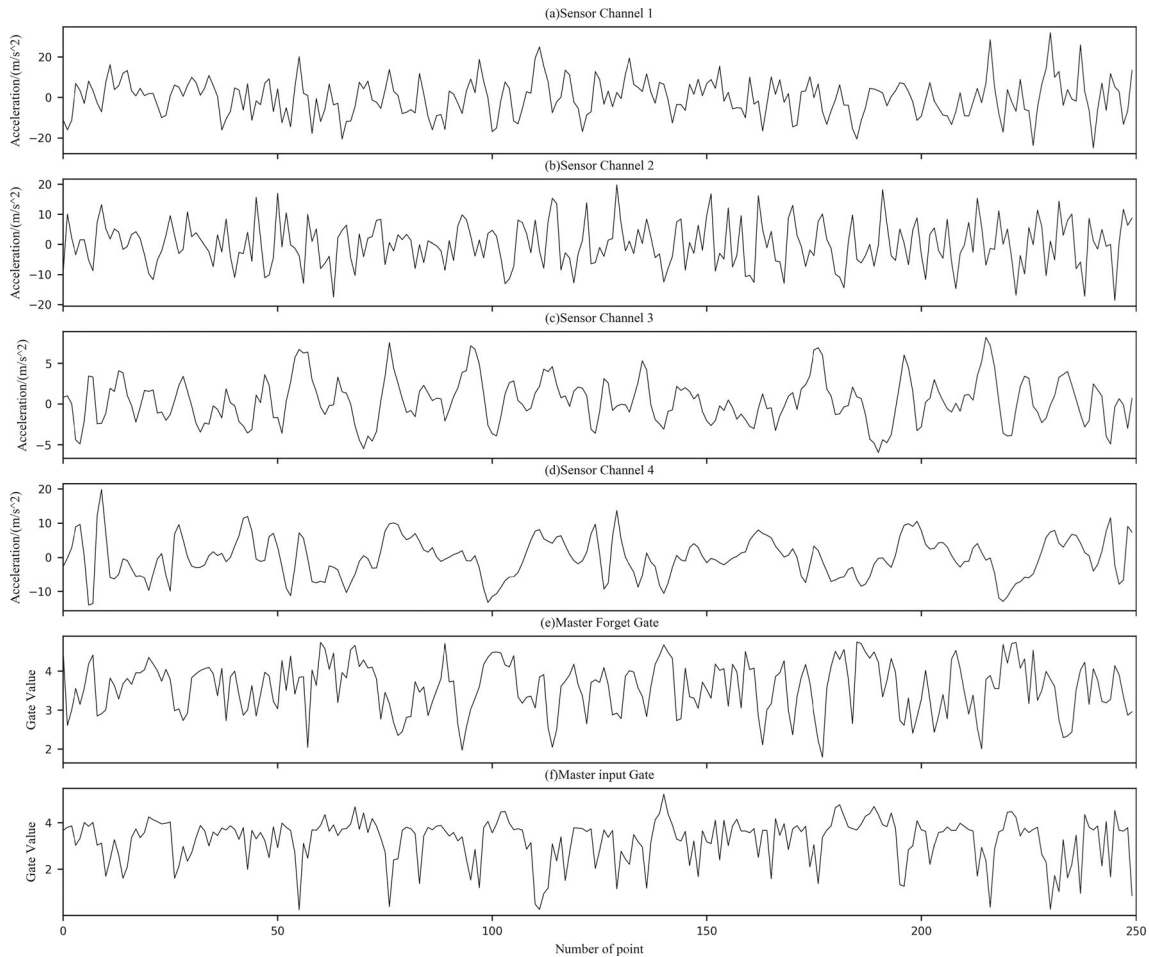


Fig. 16 Signals from sensors and master gates in one segment (spindle rotating speed = 9000 rpm, cutting width = 3.5 mm)

are deserted, after 60 epochs 1 more worker is deserted and after 80 epochs still 1 more worker is deserted. When the training is at epoch 80, only 1 worker survives.

From Fig. 13, the learning rates are large at first and then fall down gradually as the effort of the strategies in PBT. This accords with intuition. For an untrained network, the parameters inside it are far from the optimum value, so they need a relatively large learning rate to converge quickly. As the parameters getting trained, they become closer to the optimum value and if a large learning rate is still used, they may just jump over the optimum value and reach another hillside causing non convergence. Therefore, as training goes on, we need a smaller learning rate.

The predicted chatter probability by ON-LSTM is shown in Fig. 15. The results can reflect the cutting condition clearly where the cutting condition is normal at first, then turns to chatter and at last back to normal again. The computation time is important in chatter identification, since chatter usually develops within 0.1 s and the computation time should less than time length of one sample for practical application. In our experiment, the length of input data is

250 while the inference time for one sample is 3.72×10^{-4} s. As the sampling frequency in our experiments is 24,000 Hz, the sampling time for one sample is 1.04×10^{-2} s. Therefore, the computation time is much shorter than the sampling time and also short enough for chatter identification.

Although LSTM and ON-LSTM have great performance on chatter detection, we still have no idea how these recurrent neural networks work and why they have the ability for such tasks. To dive in and get interpretability of the performance of trained ON-LSTM, we estimate the signal hierarchy by probability of master gate value.

Different from the vanilla LSTM, the calculation of current cell state is by the master forget gate and the master input gate. To comprehend the working mechanism inside ON-LSTM, we transform the two parts of Eq. 13 into another form separately in Eqs. 22 and 23.

$$\begin{aligned}
 z_t \odot i_m^t &= z_t \odot i_t \odot \omega_t + z_t \odot i_m^t - z_t \odot \omega_t \\
 &= z_t \odot i_t \odot \omega_t + z_t \odot (i_m^t - \omega_t)
 \end{aligned}
 \tag{22}$$

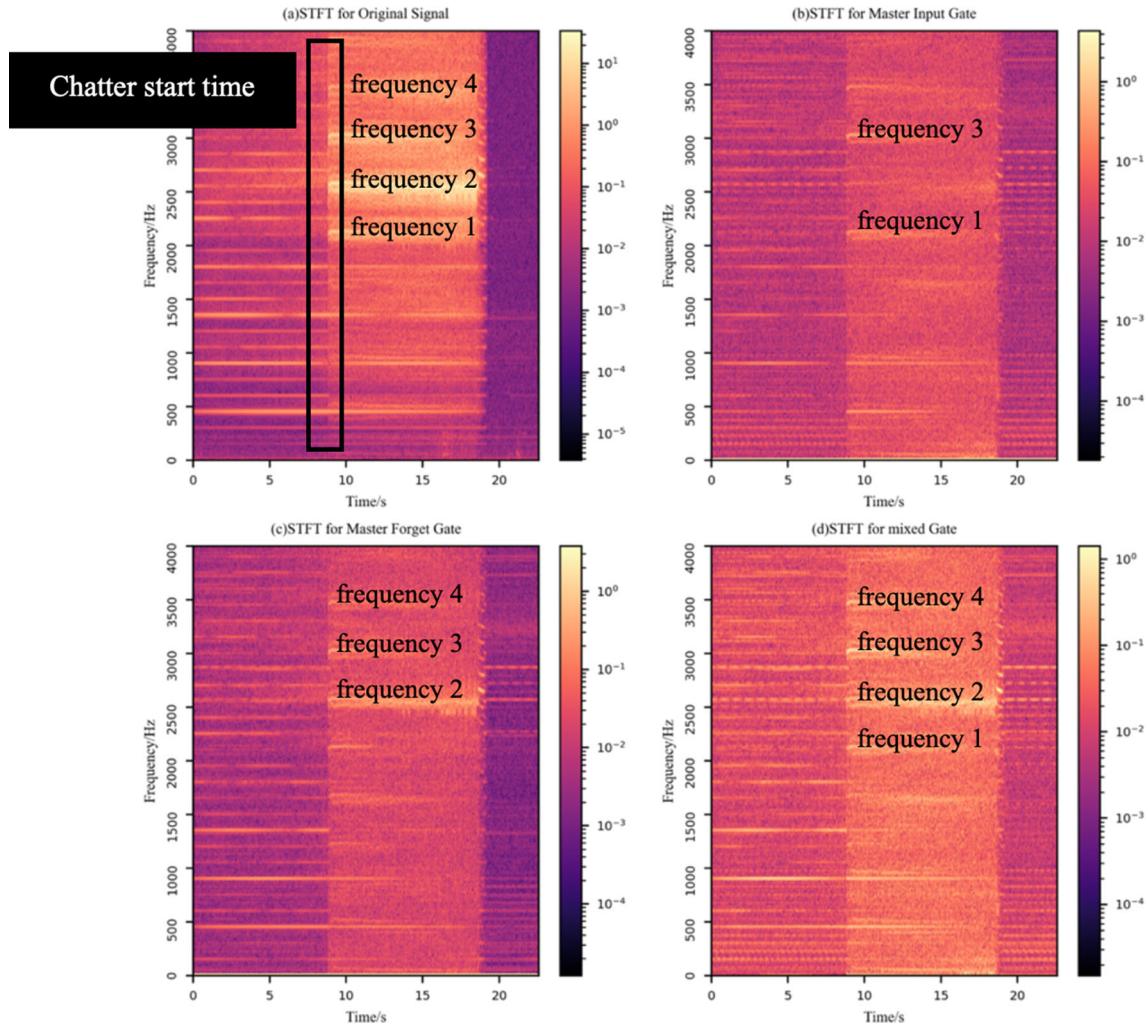


Fig. 17 Short-time Fourier transform results (spindle rotating speed = 9000 rpm, cutting width = 3.5 mm, ON-LSTM dimension = 32)

$$\begin{aligned}
 c_{t-1} \odot f_m^t &= c_{t-1} \odot f_t \odot \omega_t + c_{t-1} \odot f_m^t - c_{t-1} \odot \omega_t \\
 &= c_{t-1} \odot f_t \odot \omega_t + c_{t-1} \odot (f_m^t - \omega_t)
 \end{aligned}
 \tag{23}$$

By the transformation above, the two parts in Eq. 13 are divided into two subparts separately. The first part denotes the incoming information while the second part indicates the inherited information, and each part is a sum of two items. The first item is a element-wise multiplication of z_t , i_t and ω_t . Different from vanilla LSTM, ω_t is added to the multiplication besides z_t and i_t . ω_t means the overlapping part of the master input gate and the master forget gate, and it can apply both limitations from them. In this way, only information whose hierarchical level is both higher than current forget level and lower than current input level is flowed into current cell states. In the second item, z_t is multiplied by the deference between i_m^t and ω_t , it accepts information from input directly and blocks information according to the master input gate and the overlap. From the

above, the hierarchical structure of input data is analyzed by the master gates. It is similar in the second part of Eq. 13.

From i_m^t and f_m^t , we can learn the hierarchical level at current time point. However, what we intend to get is the hierarchical level of whole time series, so a way of calculating the wave of hierarchical level is needed. We estimate the value of current hierarchy by the expectation in Eqs. 18 and 19 in the manuscript. We mark the hierarchical level of master gates by numbers and calculate the expectation of the master forget gate and the master input gate separately. By combing the expectations, the gate value series are obtained.

Figure 16 shows the original signals from 4 sensors and the estimated hierarchical value in master forget gate and master input gate at the same time. In ON-LSTM, master forget gate and master input gate are used to learn the latent structure of the signals. Larger value of master forget gate means that the current cell should forget more information at that point and larger value of master input gate means

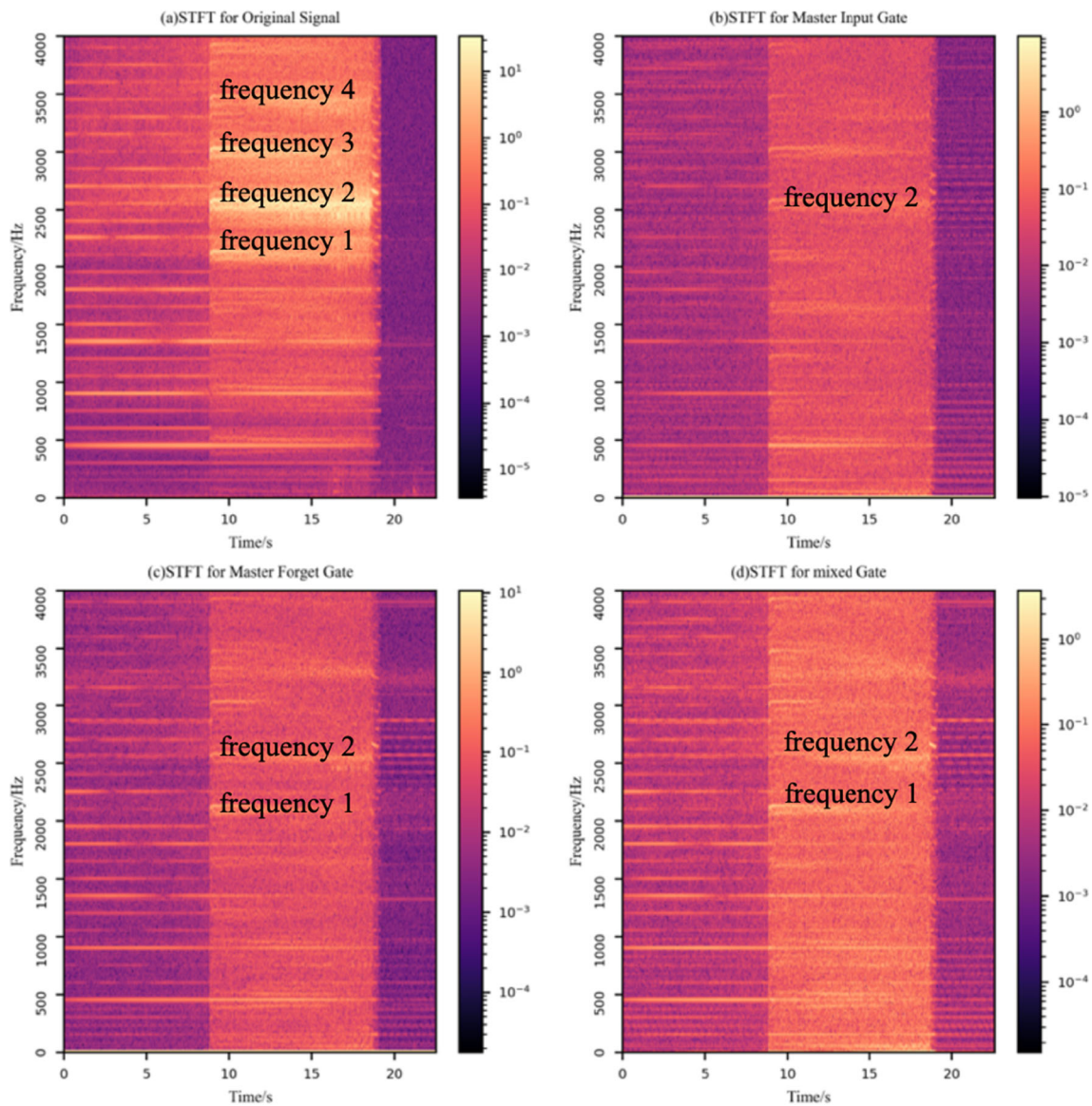


Fig. 18 Short-time Fourier transform results (spindle rotating speed = 9000 rpm, cutting width = 3.5 mm, ON-LSTM dimension = 64)

the current cell should remember more information at that point.

Some waves can be seen from the value of the two gates but the result is still obscure.

Since the length of one segment is too short, fast Fourier transform (FFT) works bad for these signals. Thanks to the flexible operation on sequences of recurrent neural networks, we can feed the whole signal into the trained ON-LSTM network to find latent structure. By doing this, we can get the master forget gate value and the master input gate value with the same length of the original signal. Still taking the first scheme as an example, where spindle rotating speed is 9000 rpm and the cutting width is 3.5 mm, STFT is applied to the signals in the 3rd sensor channel, the Master Forget Gate value, the Master Input Gate value

and the mixed Gate value. The transform results are shown in Figs. 17, 18, and 19, which can present the mechanism of ON-LSTM much clearer, and the dimension of the ON-LSTM layer is 32, 64, and 128.

In Figs. 17, 18, and 19, the spindle rotating speed is 9000 rpm, which means the cutting frequency is 450 Hz as our cutter has 3 edges. The cutting frequency is crucial in a milling process because generally it has the most energy and it is produced directly by the cutting force. This cutting frequency and its several multiplies can be seen clearly in every subplot and this means the master gates learn to trace the cutting force under both normal and chatter conditions. Since the cutting force will be extremely different when chatter occurs, this ability is very helpful for chatter identification.

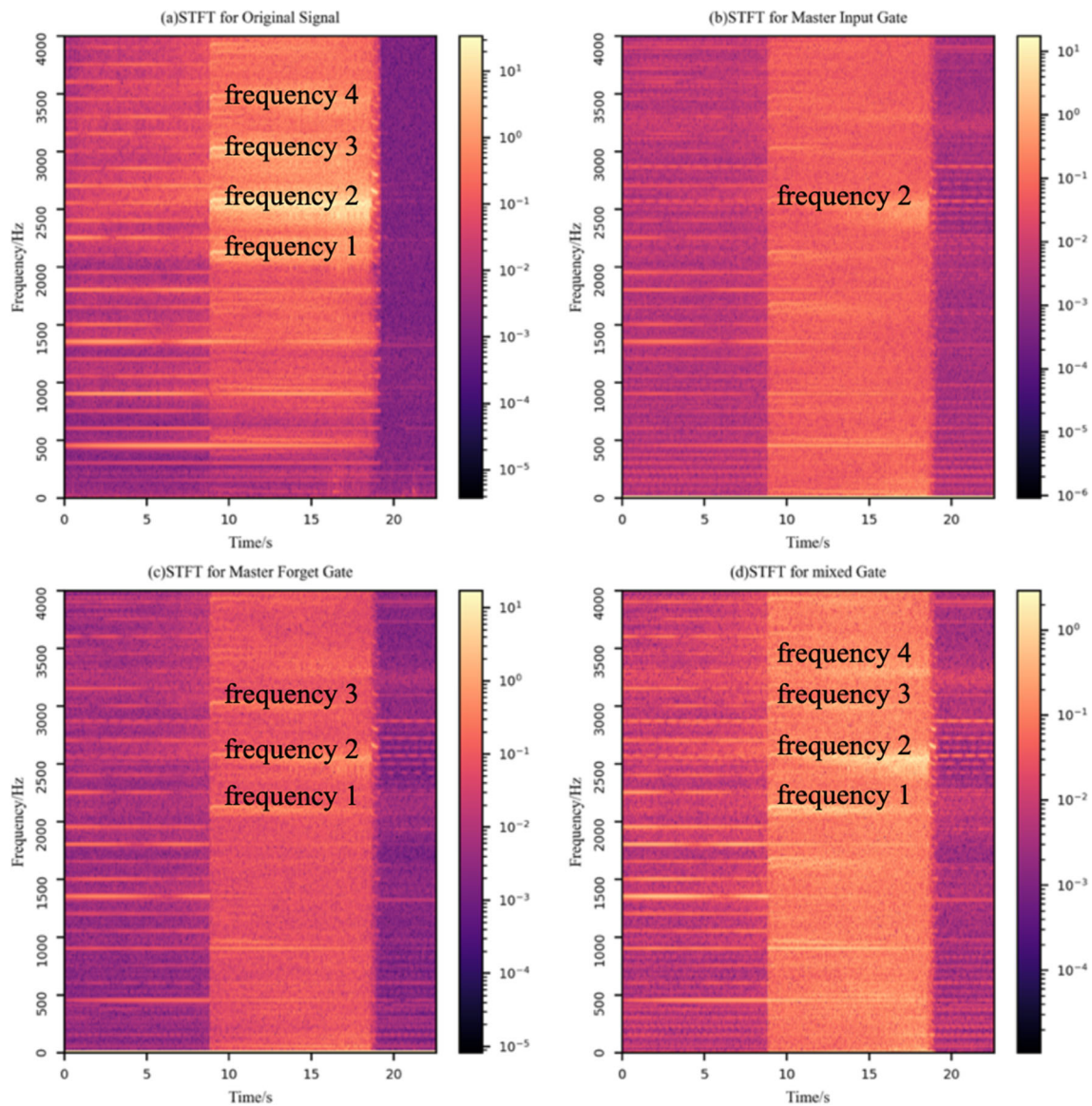


Fig. 19 Short-time Fourier transform results (spindle rotating speed = 9000 rpm, cutting width = 3.5 mm, ON-LSTM dimension = 128)

Besides this, the master gate also learns energy change when the condition turns to chatter. When chatter occurs, large amount of energy will burst because of the self-excited vibration. We can see in these frequency spectrums that the color turns lighter at certain time, which means the energy of the whole signal gets larger and this phenomenon indicates chatter. This ability of ON-LSTM network means it can learn the rapid increase of the magnitude of raw signals.

The most important thing is about the chatter frequency. The chatter frequency components only appear when chatter occurs and they have very complex mechanism which is still a hot research topic. The chatter frequency is critical in chatter identification because it is a unique frequency component and related directly to the forming mechanism of chatter. Usually, the chatter frequency is not

an accurate frequency but is several frequency bands. In the STFT result of the original signal, four new frequency components appear which we can see clearly. The first one is around 2100 Hz, the second one is around 2600 Hz, the third one is around 3000 Hz and the last one is around 3400 Hz. If the ON-LSTM network can learn these chatter frequencies, it obviously will have the ability to realize chatter identification, because these chatter frequencies are the intrinsic quality of chatter which has a relationship with the natural frequencies of the whole spindle system.

The two master gates learn most of the chatter frequencies, and some of them are clear while some of them are obscure. The master input gate learn the least. It learns the first and the third chatter frequency in Fig. 17 but only learns one chatter frequency in Figs. 18 and 19, which are

both the second chatter frequency. The master forget gate works better, it learns 3 chatter frequency in Figs. 17 and 19 and 2 chatter frequencies in Fig. 18. The best learner is the mixed gate, it learns all 4 chatter frequencies in Figs. 17 and 19 but only 2 chatter frequencies in Fig. 18.

The reason why mixed gate is the best learner is also an interesting thing. The master input gate and the master forget gate do the job of sequencing the cell according to ordered information level. Input data is controlled by the master input gate, where low-level cells are easily influenced and high-level cells ban more information. The master forget gate manages the existing information, where information of low-level cells is abandoned frequently and high-level cells hold more stable information. The estimated master input gate and the estimated master forget gate are expectations of current master input gate and master forget gate of all cells. They represents the level of input information and existing information separately. The mixed gate value is obtained by estimated master input gate value minus estimated master forget gate value. By subtracting estimated master forget gate from estimated master input gate, we can extract a synthesis of both input and existing information level. Taking periodical signals for example, when a new period comes, we should forget and input more information in more cells at the same time. Therefore, overall consideration of both the master input gate and the master forget gate has more strength in revealing the chatter frequency.

For a long time, we regard recurrent neural network as a black box and do not know why it performs so good in sequence tasks. These STFT results can explain this problem to a certain extent. The results of ON-LSTM build a bridge between the neural networks and the fault mechanism of rotating machine, and the networks may even help human understand the fault mechanisms in the future.

4 Contribution

In this paper, two kinds of recurrent neural network, LSTM and ON-LSTM are first applied to chatter identification in high-speed milling. The performance is good and the combination of network and fault mechanism is detailed. The main contributions are listed:

1. To detect chatter at the very beginning, workpieces with a custom slope are used for cutting experiments. Large amounts of experiments are done under different spindle rotating speed and cutting width to obtain signals in both normal and chatter conditions with 4 accelerators. Signals are pre-classified according to the STFT results and the resultant workpiece surface.

2. A LSTM network is built for chatter identification with different internal dimensions and the sigmoid function is used for classification at the end. The signals are divided into quite small segments which means chatter detection can be realized only by a quite short time series. Two cutting processes are selected as test set and all LSTM networks have great performance on chatter identification task where the LSTM with dimension of 64 performs best.
3. An ON-LSTM network is built for chatter identification and to find the latent hierarchical structure of the signals. PBT with 3 strategies, which are *exploit*, *explore*, and a newly introduced one *explode*, is used for model training. The trained network performs well on test set. The STFT result of master gates shows great consistency with those of original signals and reveal the latent structure of the original signals, which gives an explanation of why recurrent neural network performs well.

Funding This work is supported by the National Science Foundation of China (51575423,11772244) and China Scholarship Council (201906280415).

References

1. Quintana G, Ciurana J (2011) Chatter in machining processes: a review. *Int J Mach Tools Manuf* 51(5):363–376. Elsevier
2. Cao H, Zhang X, Chen X (2017) The concept and progress of intelligent spindles: a review. *Int J Mach Tools Manuf* 112:21–52. Elsevier
3. Ding Y, Zhu L, Zhang X, Ding H (2010) A full-discretization method for prediction of milling stability. *Int J Mach Tools Manuf* 50(5):502–509. Elsevier
4. Zhang Z, Li H, Meng G, Tu X, Cheng C (2016) Chatter detection in milling process based on the energy entropy of VMD and WPD. *Int J Mach Tools Manuf* 108:106–112. Elsevier
5. Shi F, Cao H, Zhang X, Chen X (2019) A chatter mitigation technique in milling based on H_{∞} -ADDPMS and piezoelectric stack actuators. *Int J Adv Manuf Technol* 101(9–12):2233–2248. Springer
6. Xi S, Cao H, Zhang X, Chen X (2019) Zoom synchrosqueezing transform-based chatter identification in the milling process. *Int J Adv Manuf Technol* 101(5–8):1197–1213. Springer
7. Wan M, Liang XY, Yang Y, Zhang WH (2020) Suppressing vibrations in milling-trimming process of the plate-like workpiece by optimizing the location of vibration absorber. *J Mater Process Technol* 278:116499. Elsevier
8. Cao H, Yue Y, Chen X, Zhang X (2018) Chatter detection based on synchrosqueezing transform and statistical indicators in milling process. *Int J Adv Manuf Technol* 95(1–4):961–972. Springer
9. Altintas Y, Chan PK (1992) In-process detection and suppression of chatter in milling. *Int J Mach Tools Manuf* 32(3):329–347. Elsevier
10. Cao H, Yue Y, Chen X, Zhang X (2017) Chatter detection in milling process based on synchrosqueezing transform of sound signals. *Int J Adv Manuf Technol* 89(9–12):2747–2755. Springer

11. Sun Y, Zhuang C, Xiong Z (2015) A scale factor-based interpolated DFT for chatter frequency estimation. *IEEE Trans Instrum Meas* 64(10):2666–2678. IEEE
12. Tansel I, Wang X, Chen P, Yenilmez A, Ozcelik B (2006) Transformations in machining. Part 2. Evaluation of machining quality and detection of chatter in turning by using s-transformation. *Int J Mach Tools Manuf* 46(1):43–50. Elsevier
13. Ma L, Melkote SN, Castle JB (2013) A model-based computationally efficient method for on-line detection of chatter in milling. *J Manuf Sci Eng* 135(3). American Society of Mechanical Engineers Digital Collection
14. Hynynen KM, Ratava J, Lindh T, Rikkonen M, Ryyänen V, Lohtander M, Varis J (2014) Chatter detection in turning processes using coherence of acceleration and audio signals. *J Manuf Sci Eng* 136(4). American Society of Mechanical Engineers Digital Collection
15. Cao H, Zhou K, Chen X (2015) Chatter identification in end milling process based on EEMD and nonlinear dimensionless indicators. *Int J Mach Tools Manuf* 92:52–59. Elsevier
16. Nair U, Krishna BM, Namboothiri VNN, Nampoori VPN (2010) Permutation entropy based real-time chatter detection using audio signal in turning process. *Int J Adv Manuf Technol* 46(1–4):61–68
17. Gradišek J, Baus A, Govekar E, Klocke F, Grabec I (2003) Automatic chatter detection in grinding. *Int J Mach Tools Manuf* 43(14):1397–1403. Elsevier
18. Pérez-Canales D, Álvarez-Ramírez J, Jáuregui-Correa JC, Vela-Martínez L, Herrera-Ruiz G (2011) Identification of dynamic instabilities in machining process using the approximate entropy method. *Int J Mach Tools Manuf* 51(6):556–564. Elsevier
19. Kuljanic E, Totis G, Sortino M (2009) Development of an intelligent multisensor chatter detection system in milling. *Mech Syst Signal Process* 23(5):1704–1718. Elsevier
20. Cao H, Zhou K, Chen X, Zhang X (2017) Early chatter detection in end milling based on multi-feature fusion and 3σ criterion. *Int J Adv Manuf Technol* 92(9–12):4387–4397. Springer
21. Devillez A, Dudzinski D (2007) Tool vibration detection with eddy current sensors in machining process and computation of stability lobes using fuzzy classifiers. *Mechanical Systems and Signal Processing* 21(1):441–456. Elsevier
22. Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural Computation* 9(8):1735–1780. MIT Press
23. Graves A, Schmidhuber J (2005) Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks* 18(5–6):602–610. Elsevier
24. Greff K, Srivastava RK, Koutník J, Steunebrink BR, Schmidhuber J (2016) LSTM: a search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems* 28(10):2222–2232. IEEE
25. Doetsch P, Kozielski M, Ney H (2014) Fast and robust training of recurrent neural networks for offline handwriting recognition. In: 2014 14th international conference on frontiers in handwriting recognition. IEEE, pp 279–284
26. Graves A (2013) Generating sequences with recurrent neural networks. arXiv:1308.0850
27. Zaremba W, Sutskever I, Vinyals O (2014) Recurrent neural network regularization. arXiv:1409.2329
28. Luong MT, Sutskever I, Le QV, Vinyals O, Zaremba W (2014) Addressing the rare word problem in neural machine translation. arXiv:1410.8206
29. Fan Y, Qian Y, Xie FL, Soong FK (2014) TTS synthesis with bidirectional LSTM based recurrent neural networks. In: Fifteenth annual conference of the international speech communication association
30. Shen Y, Tan S, Sordoni A, Courville A (2018) Ordered neurons: Integrating tree structures into recurrent neural networks. arXiv:1810.09536
31. Jaderberg M, Dalibard V, Osindero S, Czarnecki WM, Donahue J, Razavi A, Vinyals O, Green T, Dunning I, Simonyan K, et al. (2017) Population based training of neural networks. arXiv:1711.09846
32. Patel B, Mann B, Young K (2008) Uncharted islands of chatter instability in milling. *Int J Mach Tools Manuf* 48(1):124–134. Elsevier

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.