Strojniški vestnik - Journal of Mechanical Engineering 64(2018)7-8, 443-452
© 2018 Journal of Mechanical Engineering. All rights reserved.
DOI:10.5545/sv-jme.2018.5249
Original Scientific Paper
Received for review: 2018-01-30
Received revised form: 2018-05-08
Accepted for publication: 2018-05-23

An Improved Bearing Fault Diagnosis Method using One-Dimensional CNN and LSTM

Honghu Pan – Xingxi He* – Sai Tang – Fanming Meng
Chongqing University, The State Key Laboratory of Mechanical Transmissions, China

As one of the most critical components in rotating machinery, bearing fault diagnosis has attracted many researchers' attention. The traditional methods for bearing fault diagnosis normally requires three steps, including data pre-processing, feature extraction and pattern classification, which require much expertise and experience. This paper takes advantage of deep learning algorithms and proposes an improved bearing fault diagnosis method based on a convolutional neural network (CNN) and a long-short-term memory (LSTM) recurrent neural network whose input is the raw sampling signal without any pre-processing or traditional feature extraction. The CNN is frequently used in image classification as it could extract features automatically from high-dimensional data, while LSTM is most applied in speech recognition as it considers time coherence. This paper combined one-dimensional CNN and LSTM into one unified structure by using the CNN's output as input to the LSTM to identify the bearing fault types. First, a part of raw bearing signal data is used as the training dataset in the model, and the simulation ends when the number of iterations reaches a specific value. Second, the rest of the signal data was input in the trained model as the testing dataset to verify the effectiveness of the proposed method. The results show that the average accuracy rate in the testing dataset of this proposed method reaches more than 99 %, which outperforms other algorithms for bearing fault diagnosis.

Keywords: bearing fault diagnosis, CNN, LSTM

Highlights

- *An improved bearing fault diagnosis method based on deep learning algorithms is proposed in this paper.*
- *To take advantages of CNN and LSTM, this proposed model combined them into one structure by taking CNN's output as LSTM's input.*
- *This proposed model requires no traditional feature extraction, which is the most difficult step in traditional fault diagnosis methods.*
- *A comparison experiment with other deep learning-based models and traditional methods proved the effectiveness of this proposed method.*

0 INTRODUCTION

The bearing is the most important component in rotating machinery; its main function is to support the mechanical rotating body and reduce the friction coefficient during movement. However, continuous abrasion resulting from the relative motion between mating surfaces would cause the components' damage, and several studies have shown that the bearing fault is the major source in rotating machinery faults [1]. An effective fault diagnosis method could obtain the healthy condition of bearings and probe the fault patterns, which are also the most challenging tasks in fault diagnosis.

The traditional methods for bearing fault diagnosis using vibration signals mainly includes three steps: data pre-processing, feature extraction, and pattern classification. The features that are commonly extracted have been generated from the time domain [2], the frequency domain [3], or the time-frequency domain [4]. Next, the extracted features are fed into classifiers such as a support vector machine (SVM) [5] and [6], a decision tree [7], a BP neural network [8], etc.

The difficulty of traditional fault diagnosis methods lies in the selection of features. Any feature has its own limitations [9], e.g., the time domain feature could not detect the faulty component, the frequency domain feature is unable to identify the location of damage, the envelop analysis requires prior knowledge and professional experience and the wavelet tree feature requires pre-selection of the suitable mother wavelet and appropriate level of decomposition.

In recent years, deep learning algorithms have aroused the widespread attention of researchers as it could discover intricate structures in big data [10]. Compared with traditional machine learning algorithms, deep learning has made great progress in image recognition [11] and speech recognition [12]. Furthermore, a large number of academic achievements emerged in the field of bearing fault diagnosis using deep learning algorithms. Sun at al. [13] extracted the wavelet features and selected convolutional neural network (CNN) as the classifier, whose accuracy rate reaches 99.79 %; He et al. [14] proposed an unsupervised fault diagnosis based on a deep belief network (DBN), and it has been proven

*Corr. Author's Address: Chongqing University, The State Key Laboratory of Mechanical Transmissions, Chongqing 400044, China, xingxi@cqu.edu.cn

to outperform the back propagation neural network (BPNN) and support vector machine (SVM); Yin at al. [15] proposed an effective health assessment model by integrating Isomap into DBN with extracting time-domain, frequency-domain, wavelet packet features.

As illustrated in the literature [16], CNN is good at reducing frequency variations, and a long-short term memory (LSTM) recurrent neural network is appropriate for temporal modeling. This paper takes advantage of both CNN and LSTM and proposed an improved fault diagnosis method by combining a one-dimensional CNN and LSTM into one structure. By using this method, the limitations of traditional feature extraction can be avoided since the input of the model is the raw signal data and no traditional feature extraction is needed.

The rest of this paper is illustrated as follows: Section 1 introduces one-dimensional CNN and LSTM briefly and describes the method to combine them into one structure. Section 2 presents the bearing fault diagnosis and the results by using the proposed model. Section 3 shows the comparison results with other models. Section 4 draws the conclusion.

1 METHODS

The structure of the proposed method is shown in Fig. 1, which consists of five layers including the input layer, the convolutional layer, the pooling layer, the LSTM layer and the output layer.

Among above layers, the convolutional layer and pooling layer are applied in the CNN model, which has been proven effective in image recognition [11]. The convolution operation changes the input data into smaller feature maps through convolutional kernels. The convolutional kernels and feature maps are usually two-dimensional as the input of CNN are two-dimensional figures. To meet the one-dimensional -characteristic of mechanical signals, this paper constructs a one-dimensional convolutional neural network, whose convolutional kernels and feature maps are all one-dimensional.

Suppose the input of a one-dimensional convolutional neural network is x, which belongs to $R^{n\times 1}$, where n is the length of the input data.

Then the output of the convolutional layer can be calculated as follows [17]:

$$y_{i,j,k} = f\left(\sum_{i=1}^{s} x_{i,k} * w_{j,i} + b_i\right), \qquad (1)$$

where the $y_{i,j,k}$ is the output of the convolutional layer, $1 \le i \le m$, m is the number of samples, $1 \le j \le p$, p is the length of the convolutional kernels, $1 \le k \le n$, f is the activation function, typically a hyperbolic tangent, relu, or sigmoid function; is the input data; * is the convolution operation; $x_{i,k}$ is the weight and b_i is the bias.

The pooling layer is the sub-sampling layer to reduce the size of feature maps and prevent the overfitting. The Max pooling method is frequently used in the pooling layer whose output is the

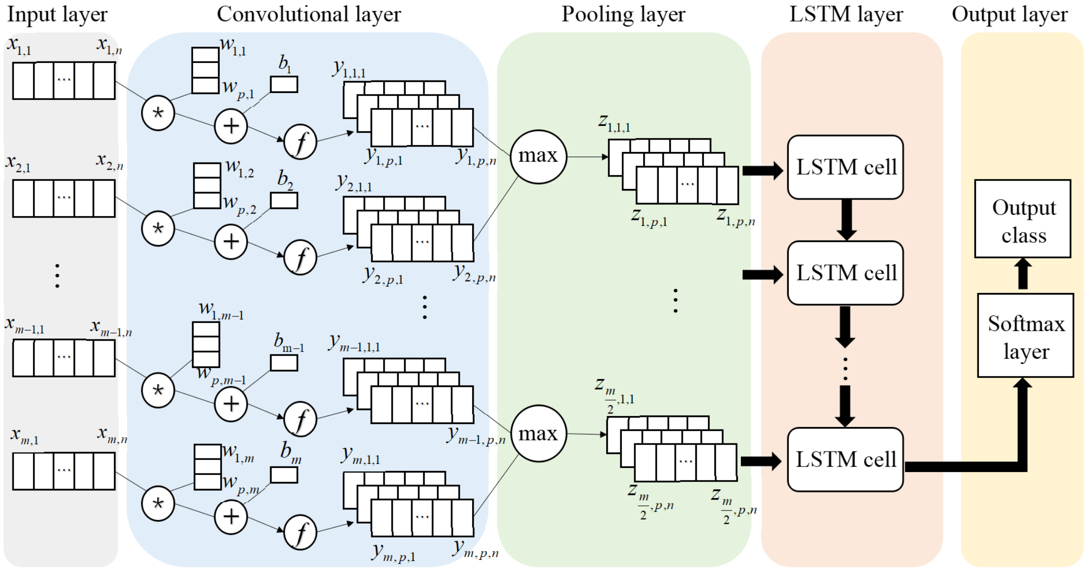

Fig. 1. *Combined structure of one-dimensional CNN and LSTM*

maximum of the previous feature maps, which can be expressed as follows [17]:

$$z_{l,j,k} = \max\left(x_{2i-1,j,k}, x_{2i,j,k}\right), \tag{2}$$

where $z_{l,j,k}$ is the output of the pooling layer, and $1 \leq l \leq m/2$.

The output of the one-dimensional CNN is taken as the input of the LSTM to reduce variance in time series. In order to solve the problem of gradient disappearance and gradient explosion in standard recurrent neural networks (RNN), Hochreiter and Schmidhuber [18] proposed the LSTM. The main difference between the LSTM and standard RNN is that the hidden units' structure of the standard RNN was replaced by LSTM cells. As shown in Fig. 2, an LSTM's cell consists of three gate structures, i.e., the forget gate, input gate and output gate and a cell structure.

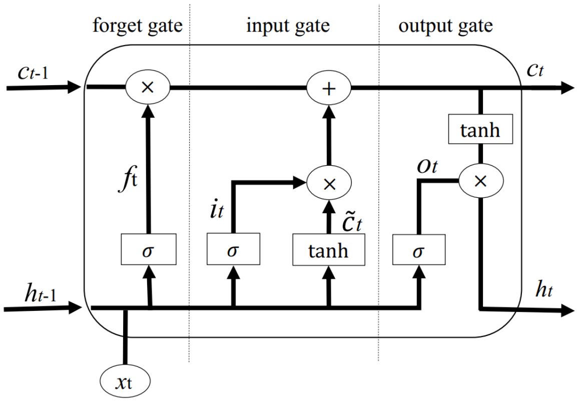

Fig. 2. *Structure of a cell of LSTM*

The output of the CNN layer is divided into $m/2$ segments, which means the input of the LSTM layer has $m/2$ time series.

The forget gate determines how much previous information could pass, whose output could be calculated as follows:

$$f_t = \sigma\left(w_{fz} z_t + w_{hf} h_{t-1} + b_f\right), \tag{3}$$

where the σ is a sigmoid function; w is weight; z_t is the current input, $1 \leq t \leq m/2$; h_{t-1} is the output of the previous cell; b_f is the bias.

The input gate determines the new information that could be saved in the cell, which could be calculated as:

$$i_t = \sigma\left(w_{zi} z_t + w_{hi} h_{t-1} + b_i\right), \tag{4}$$

$$\tilde{C}_t = \tanh\left(w_{xc} z_t + w_{hc} h_{t-1} + b_i\right). \tag{5}$$

The output gate determines what information to output from the cell state, whose output can be expressed as follows:

$$c_t = c_{t-1} f_t + i_t \tilde{C}_t, \tag{6}$$

$$o_t = \sigma\left(w_{zo} z_t + w_{ho} h_{t-1} + b_o\right), \tag{7}$$

$$h_t = o_t \times \tanh\left(c_t\right). \tag{8}$$

Behind the LSTM layer is the softmax layer for classification, which could be calculated as follows:

$$\text{softmax}\left(y_i\right) = \frac{e^{u_i}}{\sum e^{u_i}}, \tag{9}$$

where u_i is the i^{th} output of the former layer.

The output class label can be obtained after the softmax layer, which was compared with the true label of the experimental data. The backpropagation (BP) algorithm [19] was introduced to train the model, which could adjust the weights and biases by comparing the output class label with the true label and propagating the output layer's error back through the network to minimize the loss function L, which is calculated as follows:

$$L = -\frac{1}{m} \sum \left[u \ln u' + \left(1-u\right) \ln\left(1-u'\right)\right], \tag{10}$$

where m is the number of samples; u is the true label, and u' is the output class.

In the process of back propagation, the value of weight and bias are adjusted continuously until the number of iterations reaches the specific value. The parameters adjustment could be expressed as follows:

$$w_t = w_{t-1} - \varepsilon \frac{\partial L}{\partial w}, \tag{11}$$

$$b_t = b_{t-1} - \varepsilon \frac{\partial L}{\partial b}, \tag{12}$$

where ε is the learning rate, which determines the updating speed of parameters; w_t, b_t represent the value of weight and bias in t^{th} iteration; w_{t-1}, b_{t-1} represent the value of weight and bias in $(t-1)^{\text{th}}$ iteration.

The training processing was presented in Fig. 3, the output class was compared with the true label of the sample, and parameters were adjusted based on the above BP algorithm. After several iterations, the value of loss function became tiny, which means the parameters have met the samples' characteristics.

To accelerate the training process and prevent the local optima, the mini-batch gradient descent algorithm [12] is applied, in which the batch size of training samples is selected for iteration.

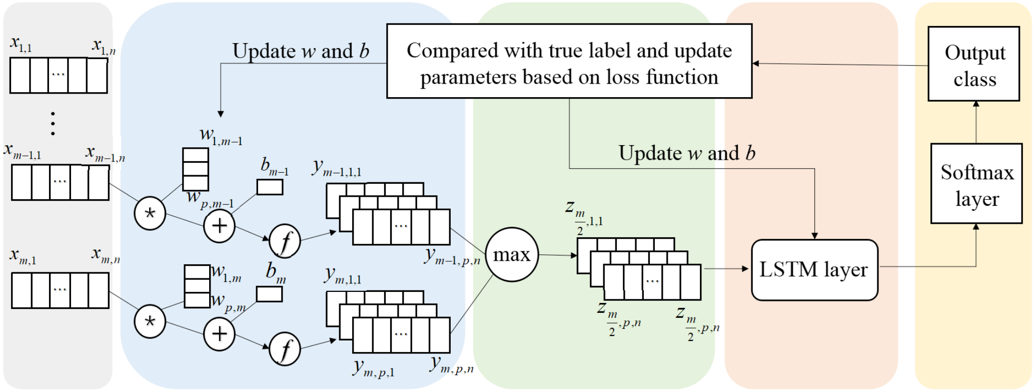
Update w and b
Compared with true label and update parameters based on loss function
Output class
Update w and b
Softmax layer
max
LSTM layer

Fig. 3. *The training processing*

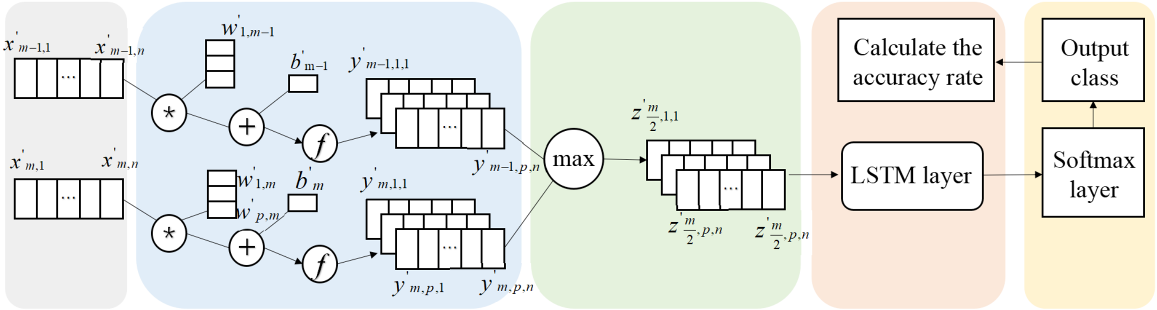

Fig. 4. *The testing processing*

After training processing finished, testing datasets is about to be input into the trained model, which was presented in Fig. 4. Parameters for testing datasets use what was updated in the last iteration of training processing.

To validate the algorithm's efficiency, the accuracy rate A_r was introduced, which could be calculated with the following formula:

$$A_r = \frac{N_r}{N_t} \times 100\,[\%], \quad (13)$$

where N_r is the number of correctly predicted samples, and N_t is the number of total samples.

The flow chart of this whole proposed model is shown in Fig. 5, where N is the max iteration epoch. Firstly, vibration signal was collected in test stand and then divided into a training dataset and a testing dataset; Secondly, the training dataset was used to train the model, and optimal parameters would be acquired after several iterations; Thirdly, the testing dataset was input to the trained model to obtain its predicted class, so the accuracy rate could be calculated based on Eq. (13).

2 BEARING FAULT DIAGNOSIS

2.1 Experimental Setup and Data Acquisition

In order to validate the proposed method, the bearing vibration data from the Case Western Reserve University (CWRU) Bearing Data Centre is applied [20]. As shown in Fig. 6, the test stand consists of a motor, a torque transducer, a dynamometer and the control electronics (not shown). The SKF bearings were used in this experiment, whose fault was introduced by electro-discharge machining. The vibration data was collected for four health conditions, i.e., the normal condition, the ball fault, the inner race fault and the outer race fault, and the fault diameters are 0.18 mm, 0.36 mm, and 0.53 mm, respectively.

The data used in this paper were collected from the drive end of the test bench. The motor provides an output power of 2.2 kW, the sampling frequency is 48 kHz and the sampling time for each dataset is approximately 10 s. The rotating speed of the shaft is 1725 r/min, which means about 1670 data points will be collected for one revolution. The first 48,430 points of each dataset are chosen and divided into 290 samples, and therefore each sample includes

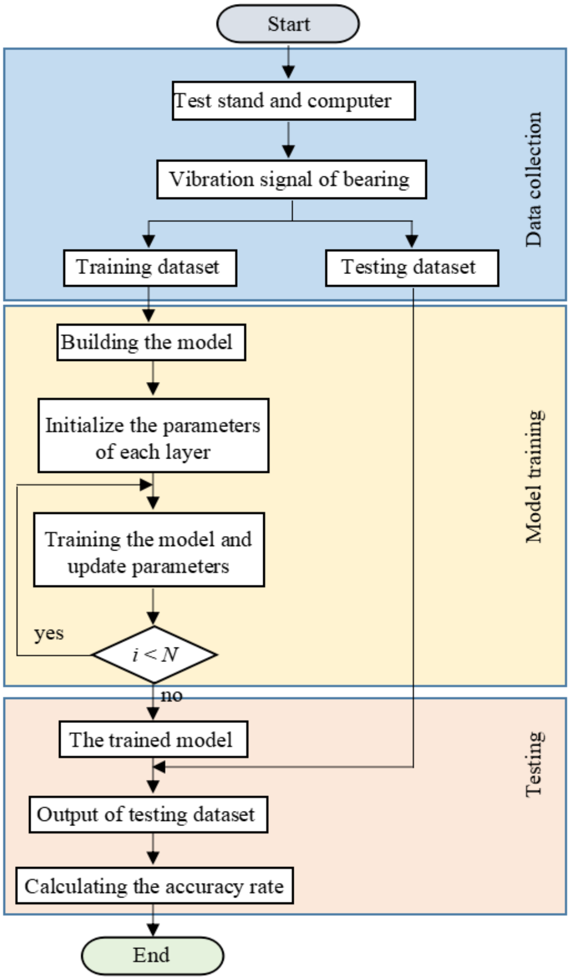
Start
Test stand and computer
Vibration signal of bearing
Training dataset
Testing dataset
Data collection
Building the model
Initialize the parameters of each layer
Training the model and update parameters
yes
$i < N$
no
Model training
The trained model
Output of testing dataset
Calculating the accuracy rate
Testing
End

Fig. 5. *Flowchart of the proposed model*

1670 points collected in one revolution. To reduce the impact of equipment fluctuations and to ensure the points in each sample are collected within the same revolution, the first 35 points and the last 35 points in each sample are discarded; therefore, every sample includes only 1600 points.

Each dataset represents a state of bearing health conditions and contains 290 samples, in which 240 samples are selected randomly as the training dataset, and the rest are used as the test set. As shown in Table 1, the proposed model processed a total of ten datasets, including one normal state and three fault types, i.e. the ball fault, the inner race fault and the outer race fault while each fault type has three variations by size. Fig. 7 shows the vibration signals of ten health conditions as referred in the Table 1, and it is hard to classify them just by intuition.

Table 1. *Description of dataset*

Health condition	Fault size [mm]	Training dataset	Testing dataset	Class label
a) normal	–	240	50	0
b) ball fault	0.18	240	50	1
c) ball fault	0.36	240	50	2
d) ball fault	0.53	240	50	3
e) inner race fault	0.18	240	50	4
f) inner race fault	0.36	240	50	5
g) inner race fault	0.53	240	50	6
h) outer race fault	0.18	240	50	7
i) outer race fault	0.36	240	50	8
j) outer race fault	0.53	240	50	9

2.2 Training Results

The model was illustrated in Section 2 in detail. The input data has three dimensions; the first dimension is the number of samples; the second dimension represents time steps, and the third dimension is a default value. The original training dataset with the dimension of 2400×1600 is changed into $2400 \times 80 \times 20$, while for the testing dataset the dimension is changed to $500 \times 80 \times 20$.

The structure of this proposed model was shown in Fig. 1. A convolutional layer is set behind the input layer, whose kernel length is chosen to be 32 and kernel channel to be 64, then the dimension of this layer's output is $2400 \times 80 \times 32$ or $500 \times 80 \times 32$. Behind the convolutional layer is the pooling layer, the pooling length and stride are both chosen to be 2, which means the dimension of this layer's output is $2400 \times 40 \times 32$ or $500 \times 40 \times 32$. Following the pooling layer, a 'relu' activation function is introduced to increase the nonlinear properties, and 0.2 dropout is introduced to prevent overfitting. The fourth layer is the LSTM layer with 128 cells, whose output dimension is 2400×128 or 500×128. The last layer is the output layer with a softmax classifier, and the data turns into 2400×10 or 500×10.

Settings of the model were shown in Table 2., and the Adam optimizer [21] is chosen to minimize the loss function. This proposed model is developed based on Python and implemented in an open resource library (keras). All experiments are performed on a computer with 4 GB GPU whose type is GTX 1050 Ti.

One important task for deep learning models is the adjustment of hyperparameters, and this paper takes the batch size and learning rate as hyperparameters. As mentioned in Section 1, the mini-batch gradient descent algorithm was used to minimize the loss function, and then the batch size defines the number

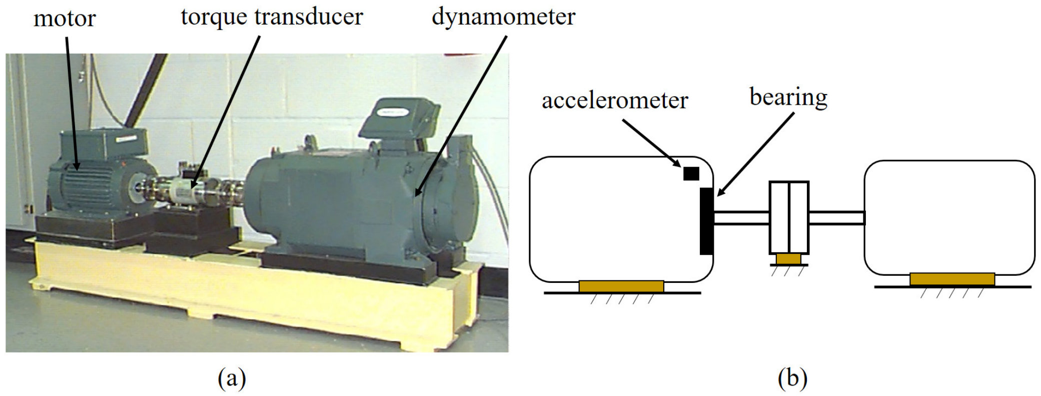

Fig. 6. *Test stand; a) a photo, and b) a scheme*

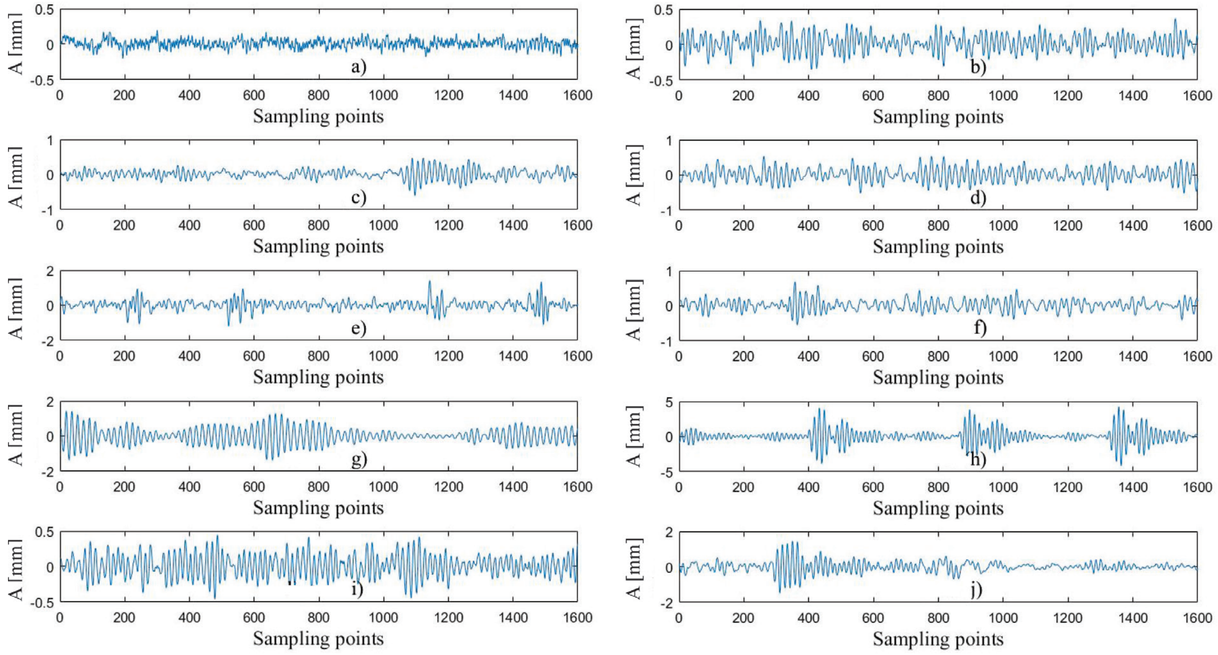

Fig. 7. *Vibration signal for different health conditions (as referred in Table 1)*
the y-axis represents the amplitude of signal and x-axis is the number of sampling points

Table 2. *Parameters for the proposed method*

Layer	Type	Parameter for layer	Other parameters

1	Input layer	Shape = [2400,80,20]	Epoch = 50
			Dropout = 0.2
			Activation = 'relu'
			Classifier = 'softmax'
			Optimizer = 'AdamOptimizer'

2	Convolutional layer	KC = 32, KL = 64	
3	Pooling layer	PL = 2, Stride = 2	
4	LSTM layer	NU = 128	
5	Output layer	OC = 10	

KC = kernal channel; KL = kernal length; PL = pooling length;
NU = number of units; OC = output channel

Table 3. *Accuracy rate in testing dataset for different configurations*

		Batch size				
		20	40	60	80	100
		Accuracy rate [%]				
Learning rate	0.0005	99.8	99.6	99.2	99.8	99.6
	0.001	99.4	99.8	99.8	99.4	99
	0.002	99.8	99.8	99.4	99	99.6
	0.004	99.6	99.6	98.6	100	99.6
	0.006	98.8	98.4	97.6	98.8	99.6

of samples to be processed in one batch. The learning rate defines the updating speed of some parameters such as weight and bias.

The accuracy rate in the testing dataset reflects the effectiveness of the model, which equals the number of correctly predicted samples divided by the total number of testing dataset samples. The results for different configurations are shown in Table 3; as can be seen, the best configuration with the batch size of 80 and the learning rate of 0.004 presents a completely correct prediction. Otherwise, the average accuracy rate is over 99 %, which proves the effectiveness of the proposed model.

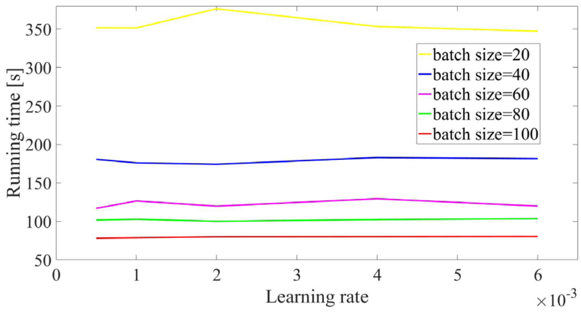

Fig. 8. *Running time for different configurations*

As shown in Table 3, the value of batch size has little effect on the prediction accuracy, while a large learning rate may lead to a decline in prediction accuracy because the parameters tend to oscillate instead of converge. Considering the similar accuracy rate for different configurations, the computing time is introduced for evaluation. As shown in Fig. 8, the main influencing factor of computing time is the batch size. The computing time for the batch size of 20 is four times more than that for the batch size of 100. Taking accuracy rate and computing time into consideration, the configuration with the batch size of 80 and the learning rate of 0.004 is recommended.

3 COMPARISONS

3.1 Comparison with Other Deep Learning Models

The comparisons between the proposed method and a DNN model, a CNN model and a LSTM model are presented in this section. The DNN model includes three layers with 256 neurons, 256 neurons, 10 neurons respectively and 0.2 dropout to prevent overfitting; otherwise, the 'relu' activation function is used to introduce non-linear properties. The configuration and parameters for the CNN model and the LSTM in this section use the same values in Table 2.

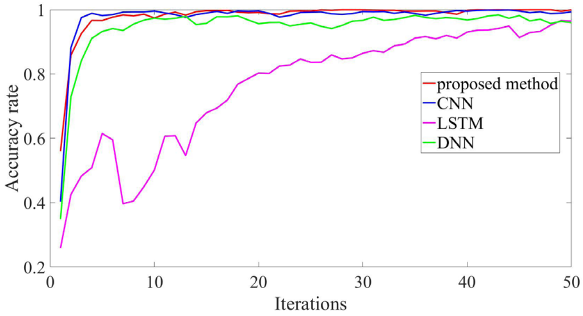

Fig. 9. *Accuracy rate in training dataset of models using deep learning algorithms*

The accuracy rate in the training dataset is used to evaluate the models; as shown in Fig. 9, the iteration speed of the LSTM model is the slowest, which requires nearly 50 iterations to achieve a 90 % accuracy rate. The CNN model performs as good as the proposed method in training dataset and better than the DNN.

Table 4 presents the accuracy rate in the testing dataset for different models, the three models for comparison achieve the accuracy rate of 98.8 %, 93.8 % and 80.8 %, respectively. Combined with Fig. 8 and Table 4, the accuracy rate of the DNN model in the testing dataset is much lower than that in training dataset due to the overfitting.

Table 4. *Accuracy rate in testing dataset of different models*

Methods	Proposed method	CNN	LSTM	DNN
Accuracy rate [%]	100	98.8	93.8	80.8

The conclusion can be drawn that the proposed model has a higher accuracy rate in the testing dataset than the CNN model, it iterates faster than LSTM model, and it is more effective in reducing overfitting than DNN model.

3.2 Comparison with Traditional Fault Diagnosis Methods

The traditional fault diagnosis methods require feature extraction before feeding data to the classifier. This paper extracted the time domain features, wavelet packet features and empirical mode decomposition (EMD) features to feed into a commonly used classifier decision tree. Table 5 presents the extracted time domain features, while wavelet packet features and EMD features are given in [22] and [23] respectively. The training and testing datasets use what mentioned in 2.1 (240 datasets for training, 50 datasets for testing).

Table 5. *Time domain features*

Features	Formula
Mean	$\bar{X} = \frac{1}{N}\sum_{i=1}^{N} x_i$
Root mean square	$X_{rms} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}$
Variance	$D_x = \frac{1}{N-1}\sum_{i=1}^{N}\left(x_i - \bar{X}\right)^2$
Peak value	$x_p = \max\left(x_i\right)$
Peak to peak value	$X_{p-p} = \max\left(x_i\right) - \min\left(x_i\right)$
Kurtosis	$D_x = \frac{\frac{1}{N}\sum_{i=0}^{N}\left(x_i - \bar{X}\right)^4}{\left[\frac{1}{N}\sum_{i=0}^{N}\left(x_i - \bar{X}\right)^4\right]^2}$
Skewness	$S = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{x_i - \bar{X}}{\sigma_X}\right)^3$
Crest factor	$C_f = \frac{X_P}{X_{rms}}$
Impulse factor	$I_f = \frac{X_P}{\bar{X}}$

The performance of models using different features is given in Table 6; the accuracy rate is in the range between 75 % and 90 % by using traditional methods. It can be concluded that the training results by using different features vary greatly and the accuracy rate of the previous experiment is nearly 10 % higher than the latter one, which proves the choice of the feature is a difficult step in traditional methods. The same feature extracted in different ways may cause different results as well. The wavelet packet feature of db4 mother wavelet outperforms than that of db3 mother wavelet and haar mother wavelet. Therefore, the overall performance of the traditional methods is worse than the deep learning-based models.

Compared with traditional or previously-used techniques of bearing fault diagnosis, the advantages of using the deep learning networks mentioned in the paper lie in the following aspects: 1) Comparably fewer iterations are required, and the computing time is acceptable. Based on Fig. 8, with the batch size of 80, the simulation time takes only 100 seconds. 2) Even though the rate of accuracy is not exactly 100 %, the result is remarkably high and quite close to 100 % (with the batch size of 80, different learning rates result in the accuracy rate all above 99.4 %). 3) No feature extraction based on prior knowledge, diagnostic experiences or professional expertise is required, and the constructed deep learning networks can extract features automatically. Traditional feature extraction methods can be quite complicated, e.g., different features suit for different signals, for instance, Fourier transform suits stationary signals while wavelet transform suits non-stationary signals. Or those methods might be hard to fully reflect the fault characteristics. Automatic feature extraction can avoid these complexities and uncertainties.

4 CONCLUSIONS

In traditional methods for bearing fault diagnosis and detection, it's necessary to extract some features to describe the signal. However, different features suit different conditions, which requires much expertise and a priori knowledge. The application of deep learning algorithms helps solve the problem as the deep learning algorithms such as the DNN, CNN, and LSTM have been proved capable of discovering intricate structures in big data.

Table 6. *Performance of different methods*

Methods	Accuracy rate in testing dataset [%]
Time domain + decision tree	80.6
EMD + decision tree	77
Wavelet packet(mother wavelet: haar) + decision tree	79.8
Wavelet packet(mother wavelet: db3) + decision tree	86.6
Wavelet packet(mother wavelet: db4) + decision tree	87.8

This paper proposed an improved bearing fault diagnosis method by combining one-dimensional CNN and LSTM into one structure. Considering CNN's advantage in reducing frequency variance and LSTM's advantage in the temporal model, the output of CNN was taken as the input of LSTM. The raw signal data collected by sensors is divided into a training dataset and testing dataset. The training dataset was used to determine the inner parameters in the model. After that, the testing dataset was fed into the trained model to verify its effectiveness. The results show that the average accuracy rate in the testing dataset is over 99 %; moreover, the model in its best configuration presents a completely correct prediction.

Compared with other deep learning-based models, the proposed model has advantages: firstly, it achieves the highest prediction accuracy in the testing dataset; secondly, it iterates faster than the LSTM model; thirdly, the proposed model is more efficient to prevent overfitting than the DNN model. Compared with the traditional fault diagnosis methods, this proposed method has the following advantages: firstly, the traditional feature extraction is not required, which eliminates the interference of inappropriate features; secondly, the prediction accuracy of this proposed method is much higher than that of the traditional methods.

However, this proposed model has its own limitation: its main disadvantage is the large amount of computation required. As the improvement of computer's computation power, this proposed method can be extended to more complicated mechanical systems, such as gearboxes. Future research will focus on the further improvement of the algorithm and its applications in other field.

5 ACKNOWLEDGEMENTS

This work was supported by the Fundamental Research Funds for the Central Universities of P.R. China (No. 106112016CDJZR288805).

6 REFERENCES

[1] Vekteris, V., Trumpa, A., Turla, V., Šešok, N., Iljin, I. Moškin, V., Kilikevičius, A., Jakštas, A., Kleiza, J. (2017). An investigation into fault diagnosis in a rotor-bearing system with dampers used in centrifugal milk separators. *Transactions of FAMENA*, vol. 41, no. 2, p. 77-86, **DOI:10.21278/TOF.41207**.

[2] Liu, F., Qian, Q., Liu, F., Lu, S., He, Q., Zhao, J. (2017). Wayside bearing fault diagnosis based on envelope analysis paved with time-domain interpolation resampling and weighted-correlation-coefficient-guided stochastic resonance. *Shock and Vibration*, vol. 2017, art ID 3189135, **DOI:10.1155/2017/3189135**.

[3] Mckee, K.K., Forbes, G.L., Mazhar, I., Entwistle, R., Hodkiewicz, M., Howard, I. (2015). A vibration cavitation sensitivity parameter based on spectral and statistical methods. *Expert Systems with Applications*, vol. 42, no. 1, p. 67-78, **DOI:10.1016/j.eswa.2014.07.029**.

[4] Do, V.T., Nguyen, L. C. (2016). Adaptive empirical mode decomposition for bearing fault detection. *Strojniški vestnik - Journal of Mechanical Engineering*, vol. 62, no. 5, p. 281, 290, **DOI:10.5545/sv-jme.2015.3079**.

[5] Fernández-Francos, D., Martínez-Rego, D., Fontenla-Romero, O., Alonso-Betanzos, A. (2013). Automatic bearing fault diagnosis based on one-class v-SVM. *Computers & Industrial Engineering*, vol. 64 no. 1, p. 357-365, **DOI:10.1016/j.cie.2012.10.013**.

[6] Zhang, X., Liang, Y., Zhou, J., Zang, Y. (2015). A novel bearing fault diagnosis model integrated permutation entropy, ensemble empirical mode decomposition and optimized SVM. *Measurement*, vol. 69, p. 164-179, **DOI:10.1016/j.measurement.2015.03.017**.

[7] Aydin, I., Karakose, M., Akin, E. (2014). An approach for automated fault diagnosis based on a fuzzy decision tree and boundary analysis of a reconstructed phase space. *ISA Transactions*, vol. 53, no. 2, p. 220-229, **DOI:10.1016/j.isatra.2013.11.004**.

[8] Zhao, Z., Xu, Q., Jia, M. (2016). Improved shuffled frog leaping algorithm-based BP neural network and its application in bearing early fault diagnosis. *Neural Computing and Applications*, vol. 27, no. 2, p. 375-385, **DOI:10.1007/s00521-015-1850-y**.

[9] Boudiaf, A., Moussaoui, A., Dahane, A., Atoui, I. (2016). A comparative study of various methods of bearing faults diagnosis using the case Western Reserve University data. *Journal of Failure Analysis and Prevention*, vol. 16, no. 2, p. 271-284, **DOI:10.1007/s11668-016-0080-7**.

[10] Lecun, Y., Bengio, Y., Hinton, G. (2015). Deep learning. *Nature*, vol. 521, no. 7553, p. 436-444, **DOI:10.1038/nature14539**.

[11] He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, **DOI:10.1109/cvpr.2016.90**.

[12] Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V. (2012). Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. *IEEE Signal Processing Magazine*, vol. 29, no. 6, p. 82-97, **DOI:10.1109/msp.2012.2205597**.

[13] Sun, W., Yao, B., Zeng, N., Chen, B., He, Y., Cao, X., He, W. (2017). An intelligent gear fault diagnosis methodology using a complex wavelet enhanced convolutional neural network. *Materials*, vol. 10, no. 7, p. 790, **DOI:10.3390/ma10070790**.

[14] He, J., Yang, S., Gan, C. (2017). Unsupervised fault diagnosis of a gear transmission chain using a deep belief network. *Sensors*, vol. 17, no. 7, p. 1564, **DOI:10.3390/s17071564**.

[15] Yin, A., Lu, J., Dai, Z., Li, J., Ouyang, Q. (2016). Isomap and deep belief network-based machine health combined assessment model. *Strojniški vestnik – Journal of Mechanical Engineering*, vol. 62, no. 12, p. 740-750, **DOI:10.5545/sv-jme.2016.3694**.

[16] Sainath, T.N., Vinyals, O., Senior, A., Sak, H. (2015). Convolutional, long short-term memory, fully connected deep neural networks. *IEEE International Conference on Acoustics, Speech and Signal Processing*, p. 4580-4584, **DOI:10.1109/icassp.2015.7178838**.

[17] Li, H., Lin, Z., Shen, X., Brandt, J., Hua, G. (2015). A convolutional neural network cascade for face detection. *IEEE Conference on Computer Vision and Pattern Recognition*, **DOI:10.1109/cvpr.2015.7299170**.

[18] Hochreiter, S., Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, vol. 9, no. 8, p. 1735-1780, **DOI:10.1162/neco.1997.9.8.1735**.

[19] Horikawa, S., Furuhashi, T., Uchikawa, Y. (1992). On fuzzy modeling using fuzzy neural networks with the backpropagation algorithm. *IEEE Transactions on Neural Networks*, vol. 3, no. 5, p. 801-806, **DOI:10.1109/72.159069**.

[20] Loparo, K. (1998). Bearings Vibration Data Set, Cast Western Reverse University, from: *https://csegroups.case.edu/bearingdatacenter/home*, accessed on 1998-July-25

[21] Kingma, D., Ba, J. (2015). Adam: A metod for stochastic optimization. *Internal Conference of Learning Representation*, p. 1-15.

[22] Wang, Z., Zhang, Q., Xiong, J., Xiao, M., Sun, G., He, J. (2017). Fault diagnosis of a rolling bearing using wavelet packet denoising and random forests. *IEEE Sensors Journal*, vol. 17, no. 17, p. 5581-5588, **DOI:10.1109/jsen.2017.2726011**.

[23] Ali, J.B., Fnaiech, N., Saidi, L., Chebel-Morello, B., Fnaiech, F. (2015). Application of empirical mode decomposition and artificial neural network for automatic bearing fault diagnosis based on vibration signals. *Applied Acoustics*, vol. 89, p. 16-27, **DOI:10.1016/j.apacoust.2014.08.016**.