

A novel ResNet-based model structure and its applications in machine health monitoring

Jian Duan¹ , Tielin Shi¹, Hongdi Zhou² , Jianping Xuan¹ and Shuhua Wang³

Abstract

Machine health monitoring has become increasingly important in modern manufacturers because of its ability to reduce downtime of the machine and cut down the production cost. Enormous signals acquired from machinery are capable of reflecting current working conditions by in-depth analysis with various data-driven methods. Hand-crafted feature extraction and representation from the traditional methods are essential but daunting tasks, and these methods may not be suitable for these massive data. Compared with traditional methods, deep learning ones are able to extract the best feature combination during model training without any artificial intervention, which makes it easier, more efficient, and more effective to monitor machine health, but the training cost and training time hamper its application. The short-time Fourier transform is adopted as the data preprocessing method to cut down the training cost and boost the training procedure. Inspired by the great achievements of ResNet, the new optimized model based on ResNet has been proposed with layer-by-layer dimension reduction of the feature maps. The proposed model is also able to avoid information loss in the conventional pooling layer. All the potential candidate model blocks are introduced and compared, and the best one is selected as the final one. Repeated model block layers are adapted for the best feature combinations, followed by a two-layer full connection layer for the final targets. The proposed method is validated by conducting experiments on bearing fault diagnosis and tool wear prediction dataset. The final results show that the proposed model achieves the best accuracy rate in the classification task and the lowest root mean squared error in the prediction task.

Keywords

ResNet, convolution neural network, machine health monitoring, bearing, tool Wear

1. Introduction

In modern manufacturing factories, it is of great value to reduce the downtime of machinery and production costs (Duan et al., 2018; Hoang and Kang, 2019; Kong et al., 2019). Machine health monitoring systems are able to master the status of the machines and then help to remind alert maintenance once the machines work improperly and provide proper maintenance strategies. Therefore, these systems have attracted more and more attention (Cerrada et al., 2018; Lenz et al., 2018; Zhao et al., 2019).

The key of the system is to grasp real-time state signals of important components in the machine together with finding and selecting the most appropriate signal analysis methods and machine learning models (Lenz et al., 2018; Yan et al., 2017). With rapid technological development of advanced sensing, big data transmission, the real-time storage, big data analysis, and deep learning algorithm, it has become possible to complete real-time condition monitoring of machines by acquiring signals under numerous

complicated working conditions of some important components of the machines (Yan et al., 2017; Yin et al., 2015; Zhao et al., 2019). As the signals should be collected completely and effectively, how to analyze these massive data and select enough features would be the next obstacle. For any potential target method, high accuracy or low prediction error is the primary requirements (Javed et al., 2018). To solve these urgent problems, there are plenty of

¹School of Mechanical Science and Technology, Huazhong University of Science and Technology, China

²School of Mechanical Engineering, Hubei University of Technology, China

³Foxconn Industrial Internet Company Co., Ltd., China

Received: 8 August 2019; accepted: 25 May 2020

Corresponding author:

Tielin Shi, School of Mechanical Science and Technology, Huazhong University of Science and Technology, 1037 Luoyu Road, Wuhan, Hubei 430074, China.

Email: tlshi@hust.edu.cn

data-driven solutions, which can roughly be classified as traditional methods and deep learning methods.

Traditional methods have three main steps analyzing signals and applying models: (1) signal preprocessing, (2) feature extraction and selection, and (3) model building and application (Javed et al., 2018; Khan and Yairi, 2018; Lei et al., 2016), as shown in Figure 1. Signal preprocessing mainly involves data cleaning, signal denoising, outlier deleting, and so on. First, signal denoising is a common trick to avoid the interference of noise and improving the quality of the signals. Many signal denoising methods have been developed over the years, such as local mean decomposition (Yu and Lv, 2017), singular value decomposition (SVD) (Zhao and Jia, 2017), and empirical wavelet transform (Chegini et al., 2019). Second, features are extracted for the sake of more completely representing the signals with less data. Time domain, frequency domain, time–frequency domain, and entropic features are popular features (Kong et al., 2018; Li et al., 2020; Zhou et al., 2017). Features are selected to remove these insensitive ones through dimension reduction methods, such as principal component analysis, linear discriminate analysis, and their variations (Kong et al., 2017, 2019; Zhao and Jia, 2018). Finally, these sensitive features will be connected to the targets by machine learning methods. Zhou et al. (2017) used kernel entropy component analysis to preserve the Renyi entropy of the signals, and a support vector machine (SVM) was built to recognize bearing states, the results were better than those of the genetic algorithm. Merainani et al. (2018) combined the Hilbert empirical wavelet transform and SVD to acquire fault feature vectors, and a self-organizing map neural network was built for fault diagnosis. Kong et al. (2017) used kernel principal component analysis (KPCA) to obtain sensitive features and proposed a support vector regression (SVR) model, and further experiments showed good prediction results of the model even when the sample was small. Zhao and Jia (2018) proposed global–local margin Fisher analysis to handle 43 features and form a low-dimensional feature

subset; they also used the Euclidean-weighted K-nearest neighbor for bearing fault diagnosis. Cuka and Kim (2017) extracted three features from each of four important signals, and the feature set was used for building a fuzzy inference system to evaluate tool wear online. Javed et al. (2018) selected four main features from force signals and proposed an ensemble of summation wavelet extreme learning machine models to predict tool life. Seera et al. (2017) adapted the conventional power spectrum and the sample entropy as the final features, and a hybrid intelligent model consisting of a fuzzy min–max (FMM) neural network and a random forest (RF) model, called FMM-RF, was designed for ball bearing fault classification. This hybrid model achieved a high accuracy performance.

Traditional methods have achieved impressive results in some cases (Abellan-Nebot and Subirón, 2010; Hoang and Kang, 2019). However, there still exist some unavoidable shortcomings in these studies (Lei et al., 2016; Qiao et al., 2018; Zhao et al., 2019). On the one hand, all these features are handcrafted, which requires plenty of prior domain knowledge and expert experience. This is a laboratory work, but results might differ greatly if the selected features share less information. On the other hand, traditional methods are usually trained and tested on small datasets. However, signals that reflect machine conditions usually yield massive samples. The effectiveness and generalization of these methods still deserve further study.

Recently, these problems are more or less overcome by introducing deep learning models for machine health monitoring (Khan and Yairi, 2018; Zhang et al., 2019; Zhao et al., 2019). Among these models, the convolution neural network (CNN) is one of the most popular types for its simple structure but powerful feature extraction ability (Zhang et al., 2019). For the application of deep learning models, there are mainly two main procedures after signal acquisition: signal preprocessing and model building and training (Khan and Yairi, 2018). Because deep learning models are able to avoid the influence of environmental noise, signal denoising seems to become not essential.

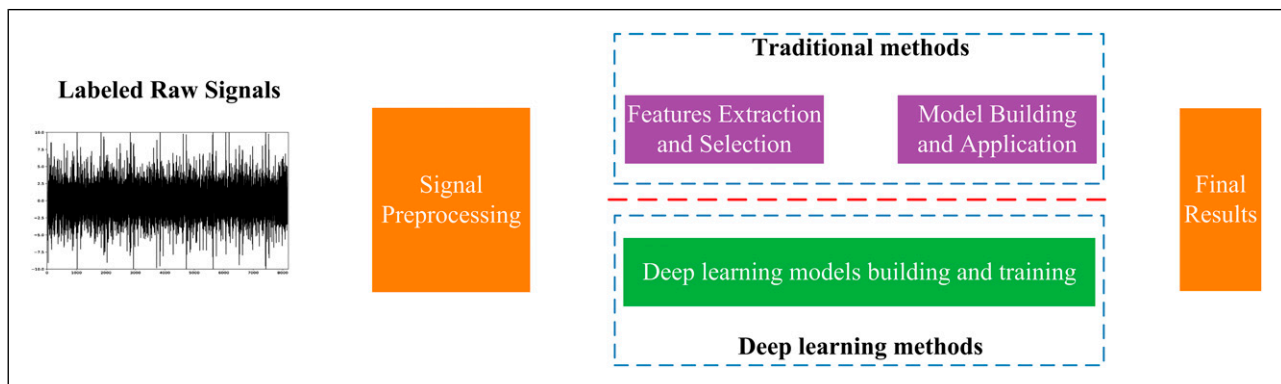


Figure 1. Procedure for signal analysis and model application.

Signals can also be transferred to 2D matrices or even the raw 1D array as the input of the models (Wang et al., 2019b; Zhang et al., 2018).

Zhang et al. (2018) proposed convolution neural networks with training interference, and the raw temporal signals were treated as the model inputs. A series of experiments showed the higher accuracy of the proposed model even under noisy conditions or different workloads. Among the 2D matrix transformations based on a time–frequency analysis, two cases (Wang et al., 2019b) were conducted with several time–frequency feature maps as the input of the CNN model. The results show that the continuous wavelet transform (CWT) and Fourier synchronous compression shared the best results, but the short-time Fourier transform (STFT) and Winger–Ville distribution showed great performance in some cases. Aghazadeh et al. (2018) proposed a novel input of the CNN model. The raw signals are decomposed by wavelet packet decomposition, and several features are extracted from the decomposition results; this procedure is followed by the spectral subtraction method. The final results are treated as the input of the CNN model. Experiments were conducted to suggest the accuracy improvement with the method. Xu et al. (2019) proposed a novel RF ensemble learning method based on three CNN models. Among these CNN models, a CWT was applied to the raw vibrations, and the time–frequency feature maps were used as the input. Further experiments validated the performance of the proposed model. Udmale et al. (2018) used the spectral kurtosis (SK) as the input of the CNN, and the experiments demonstrated the superior classification performance of this method even under different operating conditions. Cao et al. (2019) decomposed the vibration signals with a discrete wavelet transfer, and the reconstructed results were concatenated into 2D matrices as the input of a CNN model. Experiments were conducted, and the tool wear recognition results demonstrated the effectiveness of the proposed method. Wang et al. (2018) used the cutting force signals of the machining process as the CNN model to predict tool wear, the following case showed the huge space for the development of the CNN model.

For deep learning models, cases have been validated to have vaster potential than the traditional models. Models must be increasing deeper to adapt increasingly complicated conditions. Recently, the ResNet network block has become increasingly popular as an effective solution for the degradation problem of deeper models (He et al., 2016a). Wen et al. (2020) transferred the ResNet-50 model and applied it on three different datasets, and the model outperformed other compared models. Wang et al. (2019a) proposed a ResNet-based unified neural network structure for bearing fault diagnosis. However, these original ResNet network structures, that is “the shortcut connection,” can be optimized to reduce the model parameters by reducing the number of dimensions of the feature maps layer by layer,

which is helpful to avoid overfitting without losing any information.

The main contributions of this article are summarized as follows:

1. STFT has been introduced to process the raw non-stationary signals, and the parameters in STFT will be discussed in detail. Because the raw time–frequency maps may not be able to be analyzed directly by the deep learning models, the cubic spline interpolation has been used to fill the gap.
2. Based on ResNet, a series of similar networks are proposed and validated. These models are capable of reducing the feature maps without dropping out local information. Further experiments have proven the superior performance than the original ResNet models. To prevent overfitting, some useful tricks are also introduced, including data augmentation tricks, activation functions, parameter optimization, and the training process stopping tricks.
3. The parameters of the proposed models are discussed in detail classification and prediction contexts. And comprehensive experiments on bearing condition classification and tool wear prediction problems have been conducted to validate the performance of the proposed model. The proposed model still has good results even under more noisy environments without training on noisy dataset.

This article is organized as follows. In Section 2, some improvements to the proposed model and some training tricks are presented in detail. In Section 3, the model structures and parameters will be selected, and two different experiments on the classification and prediction problems will be conducted to validate the effectiveness and efficiency of the proposed model. Finally, some conclusions are provided in Section 4.

2. Methodology

In this section, explanations will be presented for data preprocessing procedure, the proposed model block, and some tricks during the training.

2.1. Data preprocessing

Deep learning has achieved outstanding results in plenty of cases, and the testing time is endurable in practice. However, the training cost for deep learning is too high, and the model training time is too long, which hamper its application in commercial and industrial scenario. Hence, the method without any data preprocessing is hardly applicable.

In general, there are also 3 steps processing the raw signals in advance, that is data augmentation, time–frequency spectrum analysis, and cubic spline interpolation of the

time–frequency maps. The whole procedure is shown in Figure 2.

Data augmentation is a simple but efficient method to prevent models from overfitting and is helpful for improving the final results. For 1D signal data, a common data augmentation solution called fixed-window slicing is applied on the source.

Frequency spectrum maps created with the fast Fourier transfer provide another and equivalent method for the signals. However, this method is not suitable for non-stationary signals, and the information from the time spectrum must be ignored. In this case, time–frequency analysis is able to solve this problem. Among all time–frequency analysis methods, STFT is a simple but reliable and powerful method (Wang et al., 2019b) and is selected to obtain time–frequency relationship in this article. Therefore, STFT is adopted as the data preprocessing method, and the method is promising for industrial applications. Mathematically, the STFT algorithm can be described as follows

$$\text{STFT}_X^{(\omega)}(t, f) = \int_t [x(t) \cdot \omega^*(t - t')] \cdot e^{-j2\pi ft} dt \quad (1)$$

At the same time, the window length N in STFT is an important parameter for the time and frequency resolution. Once N is bigger, the frequency resolution will be better, but the time resolution will be worse. On the contrary, the time resolution will be better, but the frequency resolution will be worse if N is smaller. Hence, the value N will be discussed further in the following case study. In general, the time–frequency spectrum matrices are not selected as the input of the CNN because the length and width are not the same. As

a result, cubic spline interpolation is introduced to reconstruct the matrices. By a normalization operation, the values in the training and testing datasets will have the same scale, which is helpful for model training. In this article, min–max normalization is adopted, which can be defined as follows

$$\hat{x} = \frac{x - \min\{x\}}{\max\{x\} - \min\{x\}} \quad (2)$$

For labels in the prediction tasks, the normalization will be conducted empirically as follows

$$\hat{y} = \frac{y}{100} \quad (3)$$

2.2. Proposed network block for CNN based on “residual unit”

With the rapid expansion of signal data from machines, it has become possible to collect increasingly more information under different working conditions by deepening and widening our deep learning models. Signals collected from machine tools during manufacturing are indeed massive in volume but simple in type. However, less effort has been made to establish the relationship between signals and the machine states, such as the bearing health state, and the tool wear condition. At the same time, signals under abnormal operation conditions of the machine tools are not that easy to acquire. In other words, the samples are almost always low valued and not that numerous. To avoid disappointment, many popular training tricks may have little effect on these models without requiring a larger dataset.

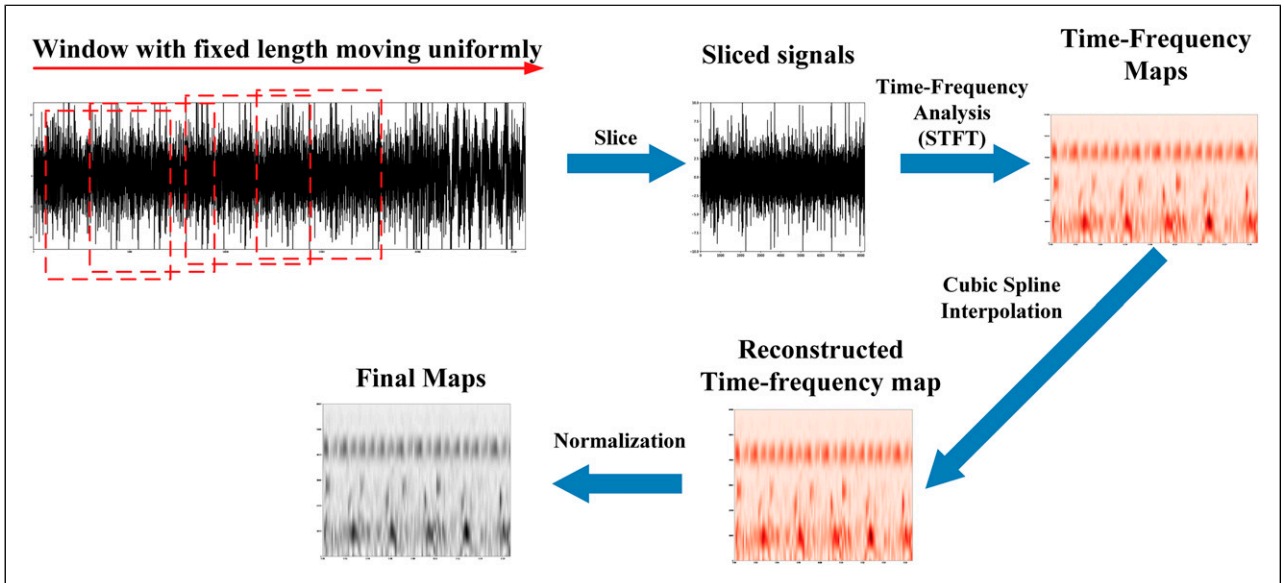


Figure 2. Preprocessing procedure of the raw signals.

The ideal models must share fewer model parameters but can achieve a higher accuracy or less error deviation.

To address this problem, He et al. (2016a) optimized the network blocks known as “shortcut connection” in highway network and proposed “residual unit.” By adopting the “bottleneck” design, these structures are quite suitable for deeper models and have been developed or used in many other models, such as DenseNet (Huang et al., 2017), Inception ResNet (Szegedy et al., 2017), Xception (Chollet, 2017), etc.

The original architecture of “residual unit” with “bottleneck” design for deeper networks is depicted in Figure 3(a). If the numbers of feature maps of the input and the output are not the same, either Conv 1 * 1 layer or the pooling layer will be chosen to connect different dimensions, which is shown in Figure 3(b). The “residual unit” can be expressed in the following form

$$y = H(x) = R(x) + F(x) \quad (4)$$

where x is the input of the layer, $R(x)$ and $F(x)$ are the functions of the residual mapping part and the identity/project mapping part, respectively, $R(x) = x$ if the mapping is identical, and $R(x) = Wx$ if the mapping is a projection.

We assume that it is easier to fit a low, fluctuating value than a high, stable one. It is a better choice to force the residual mapping layers $F(x)$ to fit the optimal relation $H(x) - R(x)$, which is usually fluctuating, than to fit a new function. Further study (He et al., 2016b) shows the superior

performance of the preactivated structure of ResNet, the structure of which is depicted in Figure 3(c).

Based on the preactivated structure, this article proposes a series of optimized model structures, which are shown in Figure 4, and further experiments will be conducted to determine the best one. Compared with the original preactivated structure, the proposed structure is designed to reduce the dimensions of the feature maps, which is helpful for avoiding overfitting. Therefore, the feature maps will be reduced to 1×1 by structure stacking. Notably, the information loss between the output and the input can also be avoided by adapting the optimized “shortcut” layer.

2.3. Proposed intelligent method

The overall structure of our proposed method is depicted in Figure 5. The structures consist of two parts: stacked optimizer structure blocks and two full connection (FC) layers. Stacked structure blocks reduce the size of the feature maps, which is vital for achieving lower hardware requirements and fewer parameters to be trained. The size of the feature maps will be reduced to 1×1 for the extreme dimensions of the feature maps. As a result, the depth of these blocks will be set according to the input of the model, and three FC layers will be used for realizing the final target, that is classification or prediction.

All layer activation functions except the final layer will be set to an exponential linear unit (ELU) (Clevert et al.,

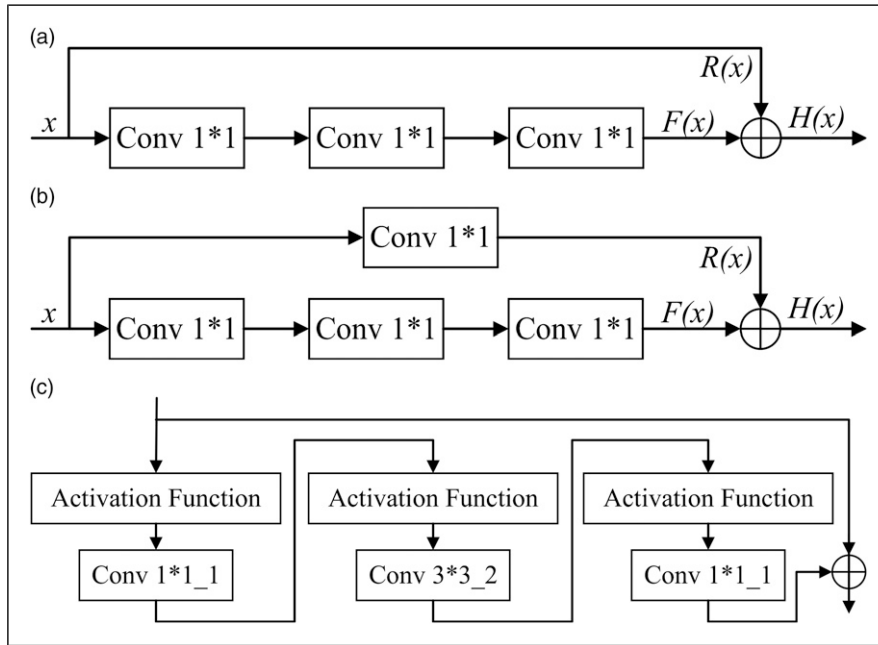


Figure 3. Original residual network block architecture (a), the refined block architecture for different dimensions of the input and the output (b), and the original preactive structure (c). Conv $N \times N_M$ denotes a convolution kernel dimension of $N \times N$, and the stride is M . Conv $N \times N$ only denotes $N \times N$ convolution kernel dimensions. \oplus denotes element-wise addition.

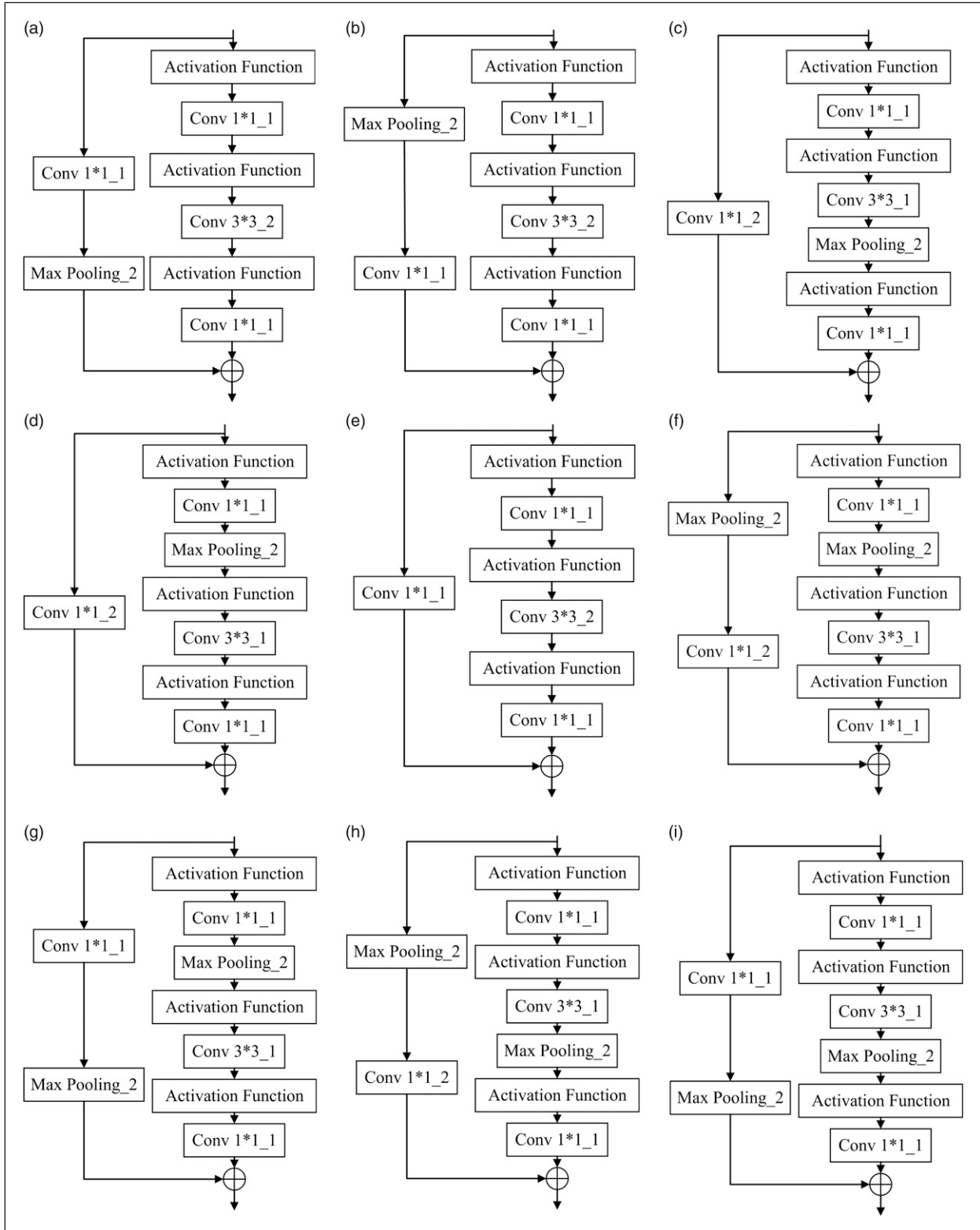


Figure 4. Series of similar proposed network.

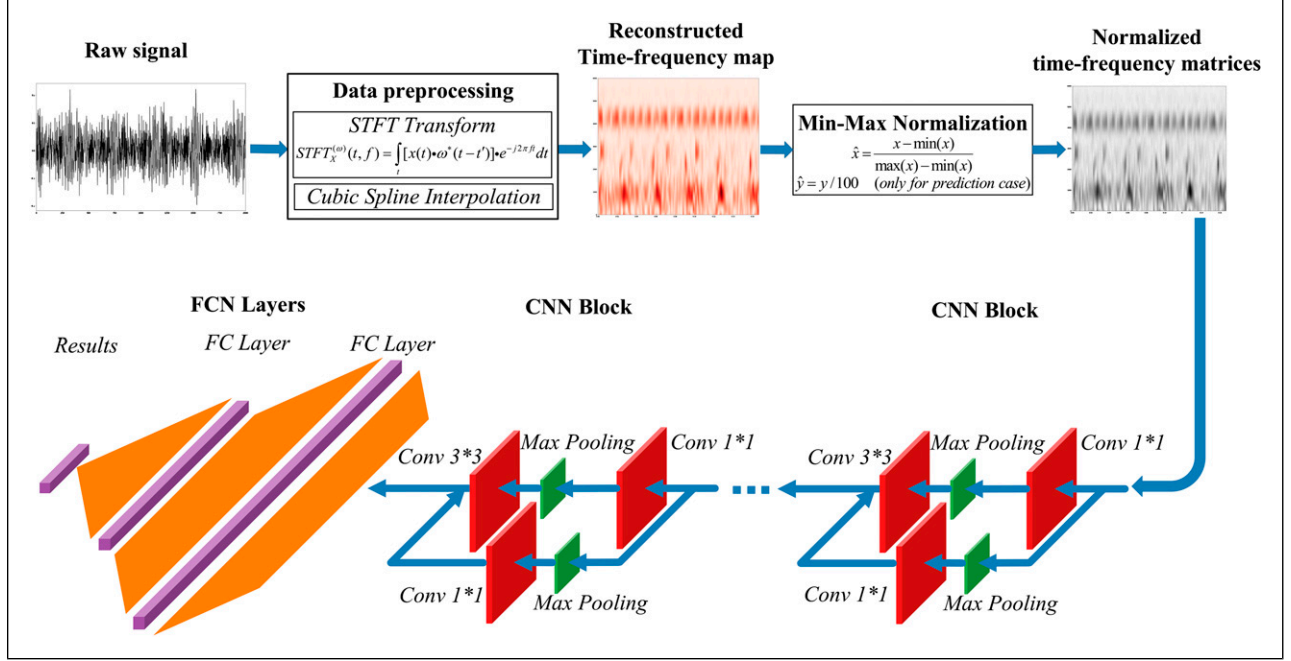


Figure 5. Structures of the proposed methods.

2015), the optimized variance in the *ReLU*. *ReLU* and *ELU* can be depicted mathematically as follows

$$ReLU = \begin{cases} x, & x > 0; \\ 0, & x \leq 0 \end{cases} \quad (5)$$

$$ELU = \begin{cases} x, & x > 0; \\ e^x - 1, & x \leq 0 \end{cases} \quad (6)$$

ReLU is quite simple; however, the neuron will “die” permanently once the output of the neuron is 0. *ELU* is able to prevent the “dead neuron” problem. In addition, *ELU* is able to accelerate convergence because the average result is close to 0.

Dropout trick is an efficient method to prevent the deep learning model from overfitting. In particular, by randomly ignoring some neurons with a probability p and blocking the connections of the FC layers in the proposed model, the models are forced to extract more information from the input and be less sensitive to the changes in values of some specific neurons. For the application in FC layers, dropout trick can be described as follows

$$\begin{cases} r_j^{(l)} \sim \text{Bernoulli}(p); \\ \tilde{\mathbf{y}}^{(l)} = \mathbf{r}^{(l)} * \mathbf{y}^{(l)}; \\ \mathbf{y}^{(l+1)} = f(\mathbf{W}^{(l+1)} \tilde{\mathbf{y}}^{(l)} + \mathbf{b}^{(l+1)}) \end{cases} \quad (7)$$

where the ignoring probability p is the dropout rate, $r_j^{(l)}$ refers to the Bernoulli distribution result under a dropout rate of p , $*$ denotes the element-wise product, and f is the

activation function of the layers. With the dropout trick, the original output of the layers $\mathbf{y}^{(l)}$ will be the final output of the dropout layer $\tilde{\mathbf{y}}^{(l)}$. In this article, the dropout rate of all the models is set to 0.5 according to experience.

For the classification task, the *softmax* function will be used to transform the results of the output layer to the probability distribution. The *softmax* function can be described as follows

$$\mathbf{Y} = \text{softmax}(\mathbf{WX} + \mathbf{b}) = \frac{e^{w_i x_i + b_i}}{\sum_k e^{w_k x_k + b_k}} \quad (8)$$

where \mathbf{W} , \mathbf{X} , and \mathbf{b} denote the input matrix, weight matrix, and bias array separately and x_i , w_i , and b_i denote the input matrix, weight matrix, and bias array of the i th of the input, respectively.

For the prediction task, no activation functions need to be applied, which means that

$$\mathbf{Y} = \mathbf{WX} + \mathbf{b} \quad (9)$$

The parameter optimizer of the model is the Nesterov accelerated gradient (NAG) (Ruder, 2016). Compared with the original mini-batch gradient descent method, the NAG optimizer is able to prevent the gradients from changing too slowly or too quickly by combining the direction of the previous gradients and the current accumulated gradients.

Early stopping tricks (Prechelt, 1998) are an efficient method to avoid the model from overfitting. The flow chart of the general training and validating procedure with early stopping is shown in Figure 6. If the validation accuracy results are no longer increasing after N criterion steps, the

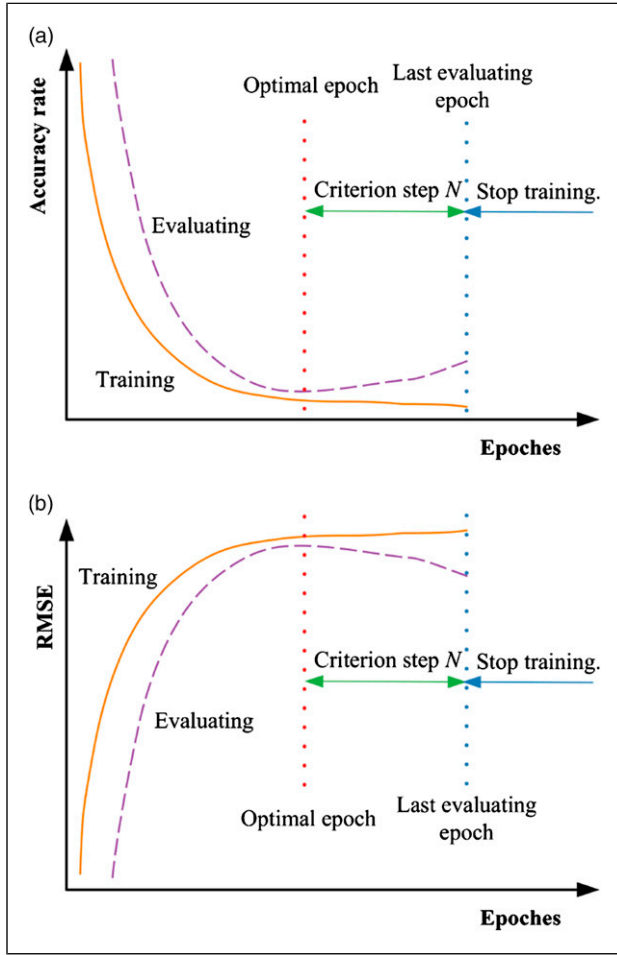


Figure 6. General procedure of the early stopping trick. (a) Classification conditions, and (b) prediction conditions.

model will stop training and the structure and current parameters of the model will be saved as the final choice. However, a criterion may not be examined in every mini-batch, which will slow down the training procedure. In this article, a criterion will be evaluated every 50 mini batches.

3. Case study and results

In this section, our proposed intelligent method will be applied and validated with two famous datasets, namely, the Case Western Reserve University motor bearing dataset and the tool wear dataset provided by the American PHM Society. The former is a typical classification case, and the main criterion is the accuracy rate, whereas the latter is a typical prediction case, and the root mean squared error (RMSE) will be used as the criterion. The accuracy rate and RMSE are defined as follows

$$\text{Accuracy rate} = \frac{T}{N} \quad (10)$$

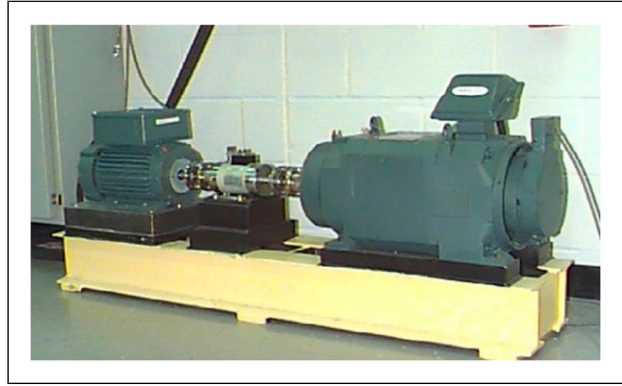


Figure 7. Test platform of the Case Western Reserve University motor bearing dataset.

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^n (\hat{y}_i - y_i)^2} \quad (11)$$

where N is the total number of dataset, T is the number of correctly classified datasets, and \hat{y}_i and y_i are the prediction tool wear values and the corresponding real values, respectively. Obviously, an ideal model should reach a high accuracy rate or a low RMSE. To prove the performance of our proposed method, other deep learning models and traditional methods will also be used for performance comparison. All the experiments will be conducted for 3 times, and the mean value will be regarded as the final result of the experiments for better reproducibility.

All the models are written in Python 3.6, Tensorflow-gpu 1.5, and Scikit-learn 0.20.1 and run on Ubuntu mate 18.04 LTS with two Intel Xeon E5-2620v4 CPUs and one NVIDIA GTX TITAN XP graphic cards with 12 GB graphic memory and 3840 NVIDIA CUDA cores. TensorFlow 1.5 provided by Google is used as the models backend except for the SVM. And the SVM is built with Scikit-learn 0.20.1.

Case 1. Bearing fault classification

In this section, typical classification problems will be discussed in detail. All the models will be trained and tested on the Case Western Reserve University motor bearing dataset (Case Western Reserved University, 2019), and the test platform is shown in Figure 7.

Data description signals will be collected on the driving end at 12 kHz sampling frequency. The bearing used in this dataset is a 6203-2RS JEM SKF, which is a deep groove ball bearing. There are three different fault types, namely, inner race faults, outer race faults, and ball faults, and the normal condition. Each type of fault contains three different diameters, that is 0.007 inches, 0.014 inches, and 0.021 inches. All 10 experimental conditions were conducted under different loads varying among 1 hp, 2 hp, and 3 hp. There

are 3000 samples in the training dataset and 450 samples in the testing dataset after data augmentation.

Considering that environmental noise is quite strong in real production lines but may not be that strong in the experiment, the models will be trained by the original training data and will be tested by testing data with added Gaussian white noise to simulate the noise interference under real industrial conditions. The signal–noise ratio (SNR) is introduced to reflect the level of the environmental noise. The models should be able to catch more characteristics from the raw training data and be more robust without being overfitted. The testing part with a length of 50,000 will be cut from the end of the vibration signal and 3 dB Gaussian white noise will be added; and the remaining part is set aside as the training data without any additional noise. The training part and testing part will be preprocessed with the method described above, and the segment length is 2048. The window length N in the STFT will be discussed in detail.

The final size of the reconstructed time–frequency representation matrices is set to 64×64 . All these parameters mentioned above are selected empirically.

Parameter selection in the proposed method First, the best model block will be selected from structures depicted in Figure 4. The parameter updating method is NAG, and the Nesterov rate is set to 0.9. Temporarily, the window length N is set to 128, and the neuron number of the FC hidden layer is set to 128. “Early stopping” trick configurations will be adopted, and the criterion step N is set to 500. The dropout rate is set to 0.5. The SNR of the testing dataset will be 3 dB.

The final results are shown in Figure 8. Obviously, the accuracy rate varies greatly among the different

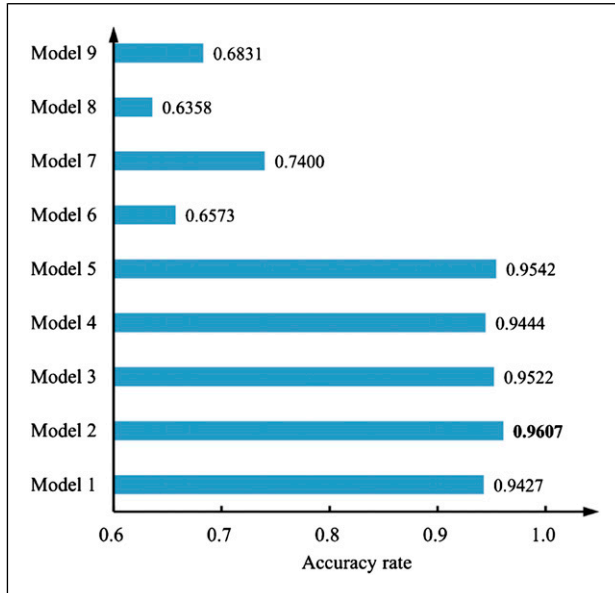


Figure 8. Results of models with a series of similar optimizer structures.

structures, and the two network structures perform better achieving an accuracy rate of 0.9607. Hence, this structure will be adopted as the final proposed model structure.

Then, the window length N in STFT is discussed in depth. The window length N will affect the time and frequency spectrum resolution and determine the result of reconstructed time–frequency map. The NAG and “early stopping” configurations remain the same. Temporarily, the neuron number of the FC hidden layer is set to 128.

The final results are shown in Figure 9(a). Obviously, the model performs best when the window length N is 128. Hence, the window length N will be set to 128 in this section.

Finally, the neuron number of the FC hidden layer should be determined. The NAG and “early stopping” configurations still remains the same. The final results are shown in Figure 9(b). Obviously, the model performs best when the neuron number is 128. Hence, the neuron number of the FC hidden layer will be set to 128 in this section.

Model comparison and results To validate performance on the accuracy rate of our proposed method, the following models are chosen for comparison:

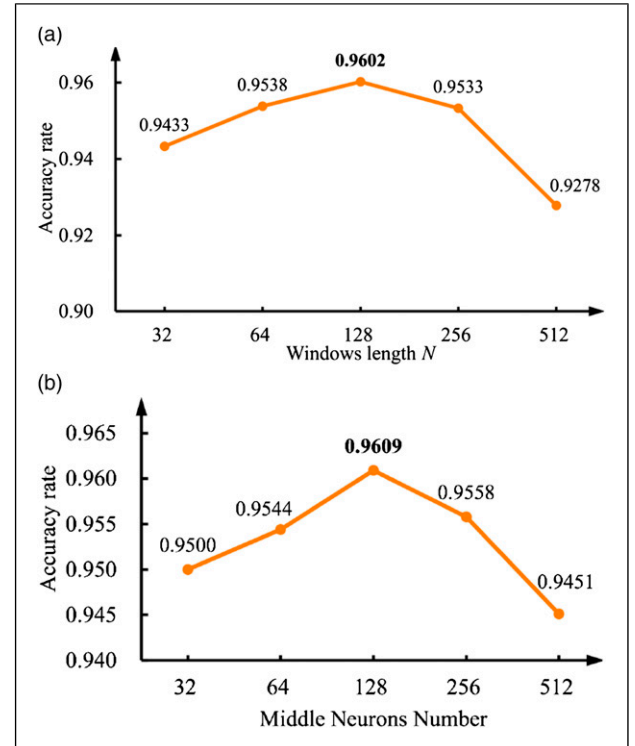


Figure 9. Experimental results of the classification model parameter selection. (a) The results of the models with different window lengths N in short-time Fourier transform of the proposed classification method, and (b) the results of the models with different neuron numbers in the full connection hidden layer of the proposed method.

Table 1. Kernel number in the network configurations of the compared classification methods.

Kernel number or list	Proposed model	ResNet-like model	Multilayer perception model
Layer or block 0	[16, 64, 64]	[16, 64, 64]	4096
Layer or block 1	[16, 64, 64]	[16, 64, 64]	—
Layer or block 2	[32, 128, 128]	[32, 128, 128]	1024
Layer or block 3	[32, 128, 128]	[32, 128, 128]	—
Layer or block 4	[64, 256, 256]	[64, 256, 256]	512
Layer or block 5	[64, 256, 256]	[64, 256, 256]	—
Hidden layer in the FC layers	128	128	128

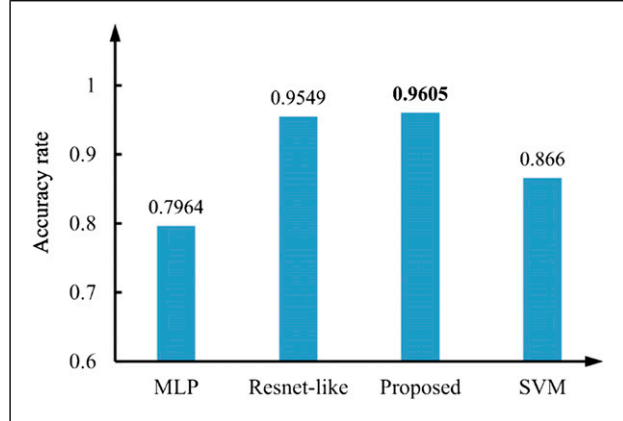
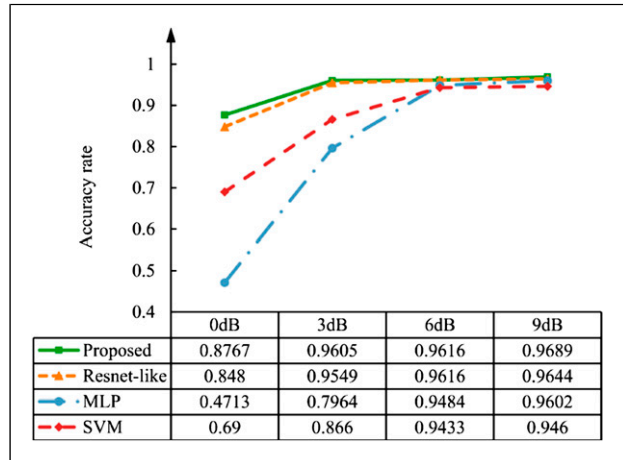
- The proposed method,
- a ResNet-like model,
- a multilayer perception (MLP) model with flattened array input, and
- an SVM model with flattened array input.

Because the output maps of the proposed structure blocks should be reduced to 1×1 , the proposed method consists of 6 stacked blocks and three FC layers. ResNet-like model shares the same layers and the parameters of each model block with our proposed model. The MLP model consists of 4 FC layers. The kernel number or the list of each block or layer parameters for these models are depicted in Table 1. The network configurations, such as the batch size, activation function, optimizer function, and the “early stopping” configurations, are kept the same as in the above proposed model selection experiments. Because the SVM and the MLP are unable to process the 2D array directly, these 2D data will be flattened to a 1D array first. A SVM with Gaussian kernels is then chosen to compare the accuracy rate of the results with that of the proposed model. The penalty parameter C is set to 10, and parameter gamma of the kernel is set to 0.1. All of these parameters are determined by previous grid search experiment. The testing dataset with 3 dB Gaussian white noise is used for comparison.

The results of the different models are shown in Figure 10. Obviously, the accuracy rate of the compared models reaches the best. Compared with the other models, the proposed model performs best with 96.05% accuracy rate.

As we mentioned above, environmental noise is inevitable in the real world. An additional experiment is designed to show the robustness of these models by training and validating the models with a series of datasets adding different SNR of Gaussian white noise, that is 0 dB, 3 dB, 6 dB, and 9 dB. The final results are shown in Figure 11.

The experimental results show that the proposed model achieves the highest accuracy rate compared with the other models. In general, the deep learning model performs better than the shallow learning models, especially when the environmental noise is heavy. Because of the ability to

**Figure 10.** Results of the model comparisons.**Figure 11.** Test results of the models under different environmental noise level.

reduce the feature maps without losing local information, the proposed model performs better than the original ResNet model, even under noisy environmental conditions. In particular, the accuracy rate of the proposed model still achieves nearly 90% under 0 dB Gaussian white noise conditions, although the models are trained by the dataset on less noisy conditions, while the accuracy rate of the MLP is below 50%. However, the SVM performs better than the

MLP even under heavy noise environment, which means that an improper model might also lead to terrible results. In addition, the average classification time for the proposed model is 0.0037s per time, which is endurable in practice. In summary, the proposed model shows better potential for the application in industrial conditions.

Case 2. Tool wear prediction

In this section, typical prediction problems are discussed in detail. All the models will be trained and tested under the tool wear dataset provided by the American PHM Society

(American PHM Society, 2019; Li et al., 2009). The test platform is shown in Figure 12.

Data description This dataset contains the whole milling procedure of three individual cutter records with the same experimental setup, as shown in Table 2. Tri-axis acceleration signals are collected on spindle with Kistler piezo accelerator at 50 kHz sampling frequency by NI cDAQ PCI-1200. The z-axis acceleration signal will be selected for tool wear prediction. Each cutting dataset shares 315 individual data acquisition sample files, and the data augmentation methods above are applied on every sample in all 3 datasets. After augmentation, each cutting dataset contains 9450 samples.

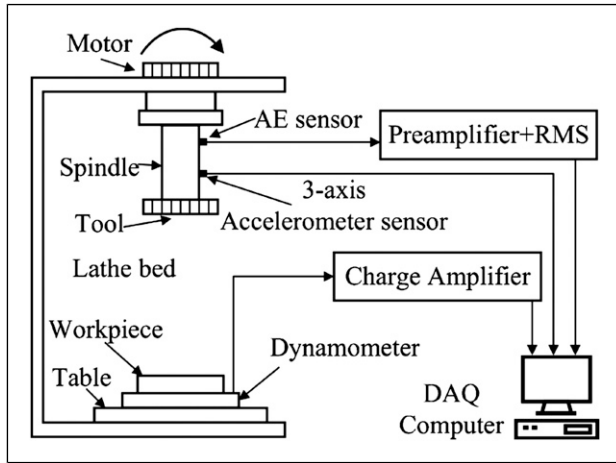


Figure 12. Test platform of the American PHM Society tool wear dataset.

Table 2. Experimental setup in the American PHM Society tool wear dataset.

Cutting factor	Value
Spindle speed	10400 r/min
Y depth of cut (radial)	0.125 mm
Z depth of cut (axial)	0.2 mm
Feed rate	1555 mm/min

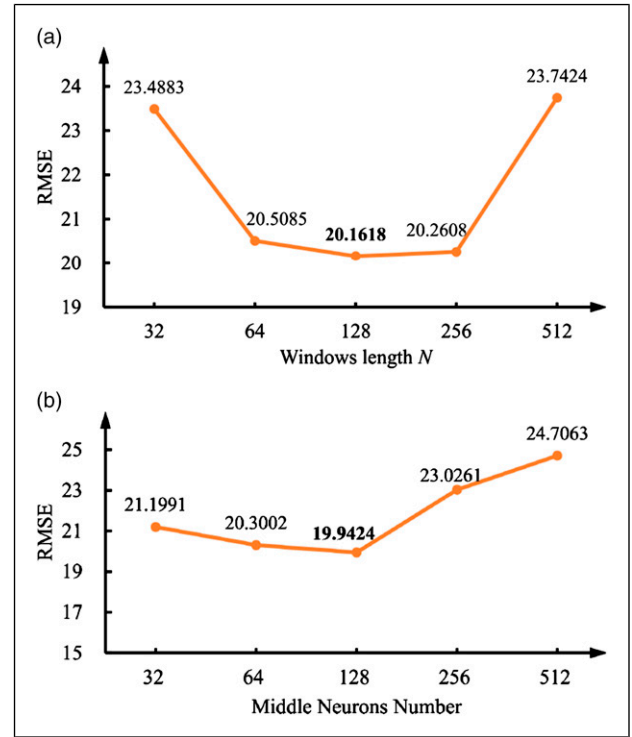


Figure 13. Experimental results of prediction model parameter selection. (a) The results of the models with different window lengths N in STFT of the proposed method, and (b) the results of the models with different neuron numbers in the full connection hidden layer of the proposed method.

Table 3. Kernel number in the network configurations of the compared prediction models.

Kernel number or list	Proposed model	ResNet-like model	Multilayer perception model
Layer or block 0	[32, 128, 128]	[32, 128, 128]	16,384
Layer or block 1	[32, 128, 128]	[32, 128, 128]	—
Layer or block 2	[64, 256, 256]	[64, 256, 256]	4096
Layer or block 3	[64, 256, 256]	[64, 256, 256]	—
Layer or block 4	[128, 512, 512]	[128, 512, 512]	1024
Layer or block 5	[128, 512, 512]	[128, 512, 512]	—
Layer or block 6	[256, 1024, 1024]	[256, 1024, 1024]	256
Hidden layer in the full connection layer	128	128	128

Tool wear was measured offline by Leica MZ12 microscope. The segment length is 4096, and the original data will be enlarged 30 times to form a dataset. The window length N in STFT will also be discussed in detail. The final size of the time–frequency representation reconstructed matrices is empirically set to 128×128 .

Parameter selection in the proposed method the window length N of the STFT is discussed at the beginning. The NAG and “early stopping” tricks keep the same as the configurations in Case 1, and the criterion step N is set to 800. Temporarily, the neuron number of the FC hidden layer is set to 128. The dropout rate is set to 0.5.

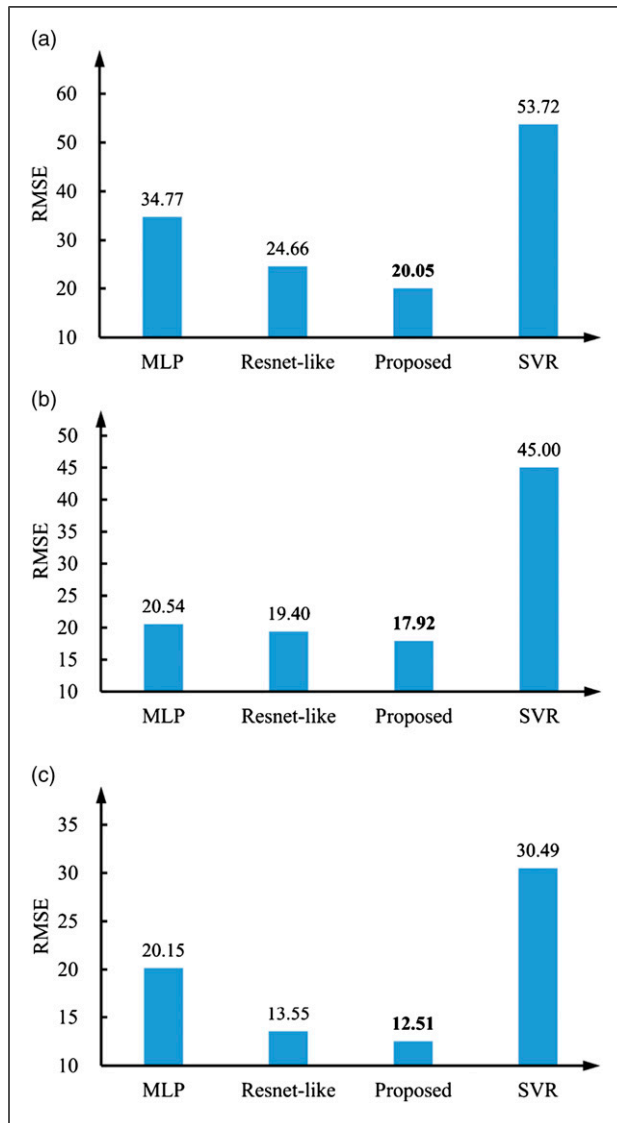


Figure 14. Tool wear predicted results of model comparison. (a) Cutter 1 and cutter 2 as the training dataset, (b) cutter 1 and cutter 3 as the training dataset, and (c) cutter 2 and cutter 3 as the training dataset.

The final results are shown in Figure 13(a). Obviously, the results are optimal if the window length is 128. Hence, the window length N will be set to 128 in this section.

Finally, the neuron number of the FC hidden layer should be decided. The NAG and “early stopping” tricks still keep the same.

The final results are shown in Figure 13(b). Obviously, the results are optimal if the neuron number in the FC hidden layer is 128. Hence, the neuron number in the FC hidden layer will be set to 128 in this section.

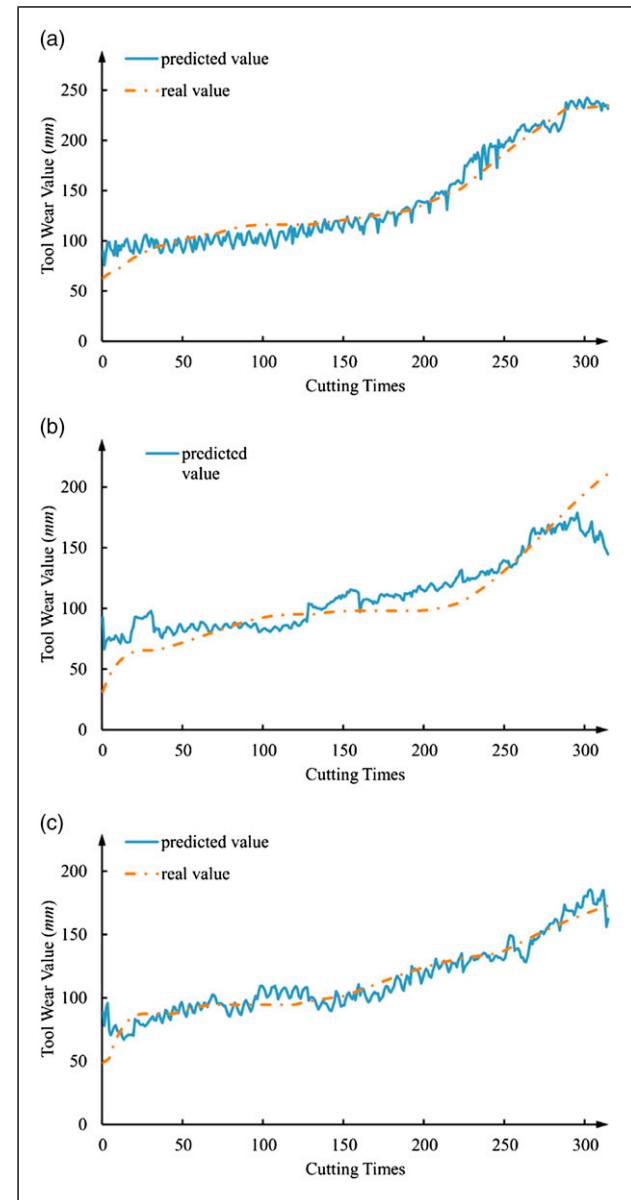


Figure 15. Predicted and the actual tool wear of all three cutter datasets. (a) Cutter 3 prediction and actual tool wear curves, (b) cutter 2 prediction and actual tool wear curves, and (c) cutter 1 prediction and actual tool wear curves.

Model comparison and results to validate the performance of our proposed method, the following corresponding models will be compared:

- The proposed model,
- a ResNet-like model,
- an MLP model with flattened array input, and
- an SVR model with flattened array input.

Because the output maps of the proposed structure blocks should be reduced to 1×1 , the proposed method consists of seven stacked blocks and three FC layers. The ResNet-like model shares with the same layers and the parameters of each model block with our proposed model. The MLP model consists of four FC layer. The kernel number or list of each block or layer parameters for these models are shown in Table 3. The network configurations, such as the batch size, activation function, optimizer function, and the “early stopping” tricks, are kept the same as in the above proposed model selection experiments. Because the SVR and MLP are unable to process the 2D array directly, these 2D data will be flattened to 1D array first. An SVR with Gaussian kernels is then chosen to compare the accuracy rate of the results with the proposed one, and the penalty parameter C is set to 1 and parameter gamma of the kernel is set to 100. All of these parameters are determined by previous grid search experiment. Two cutting dataset will be selected as the training dataset and the remaining one will be set aside as the testing dataset. Therefore, there are three kinds of experiment combinations. The prediction results of all experiment combinations are shown in Figure 14(a)–(c).

In general, the prediction problem seems to be more difficult, complicated, and challenging than classification one. The experimental results show that deep learning models still achieve better RMSE values than the shadow models with only z-axis acceleration signal. Among these experiments, the MLP outperforms SVR at all times. The proposed model still keeps the advances compared with other models in three cutter tool wear prediction. The prediction and the actual tool wear of all three cutter datasets are shown in Figure 15(a)–(c). In addition, the average prediction time for the proposed model is 0.0038 s, 0.0040 s, and 0.0040 s per time of all three experiments, respectively, which are also endurable in practice. The results have illustrated that the proposed model is able to predict the main tool wear trend quite well, which has proven the impressive performance of the proposed model.

4. Conclusions

A novel ResNet-based CNN model has been proposed and validated on datasets of machine health monitoring conditions. STFT is used to analyze the nonstationary signals because it is easier for STFT to be realized, and the results

remain good. The time–frequency representation results are processed models followed by cubic spline interpolation. Some tricks during model training are also explained. Then, novel models based on ResNet were proposed. A series of potential models based on ResNet have been proposed and validated. These models are designed to reduce the feature map dimensions without the loss of local information. Finally, the model parameters are determined at the beginning of each case, and the subsequent experiments validate the outstanding performance of the proposed model. Regardless of the classification or the prediction context, the proposed model still achieved great performance even when trained by a dataset on less noisy conditions, which shows the great potential of the application of the proposed model in the industrial conditions.

In the future, subsequent works might mainly focus on the further optimization of the proposed model structure and the applications of the proposed model trained by less noisy dataset on industrial conditions. At the same time, we will begin to pay more attention to questions such as bearing condition prognostic problems and tool remaining useful life prediction.

Acknowledgements

We would like to thank the Case Western Reserve University Bearing Data Center for providing the bearing vibration data and American PHM Society for providing the tool wear dataset.

Declaration of conflicting interests


The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This paper was partially funded by Key-Area Research and Development Program of Guangdong Province (Grant No. 2020B090927002), the National Natural Science Foundation of China (Grant No. 51675204), the National Science and Technology Major Project of China (Grant No. 2018ZX04035002-002), the Natural Science Foundation of Hubei Province (Grant No. 2019CFB326) and the Scientific Research Foundation for Doctoral Program of Hubei University of Technology (BSQD2017003).

ORCID iDs

Jian Duan  <https://orcid.org/0000-0003-2493-9453>

Hongdi Zhou  <https://orcid.org/0000-0002-6618-3984>

References

- Abellan-Nebot JV and Subirón FR (2010) A review of machining monitoring systems based on artificial intelligence process models. *The International Journal of Advanced Manufacturing Technology* 47(1–4): 237–257.

- Aghazadeh F, Tahan A and Thomas M (2018) Tool condition monitoring using spectral subtraction and convolutional neural networks in milling process. *The International Journal of Advanced Manufacturing Technology* 98(9–12): 3217–3227.
- American PHM Society (2019) 2010 PHM society conference data challenge. Available at: <https://www.phmsociety.org/competition/phm/10> (last accessed on 12 July 2019).
- Cao X-C, Chen B-Q, Yao B, et al. (2019) Combining translation-invariant wavelet frames and convolutional neural network for intelligent tool wear state identification. *Computers in Industry* 106: 71–84.
- Case Western Reserve University (2019) Case western reserve university bearing data center website. Available at: <http://csegroups.case.edu/bearingdatacenter/home> (last accessed on 12 July 2019).
- Cerrada M, Sánchez R-V, Li C, et al. (2018) A review on data-driven fault severity assessment in rolling bearings. *Mechanical Systems and Signal Processing* 99: 169–196.
- Chegini SN, Bagheri A and Najafi F (2019) Application of a new ewt-based denoising technique in bearing fault diagnosis. *Measurement* 144: 275–297.
- Chollet F (2017) Xception: deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, Honolulu, USA, 21–26 July 2017, pp. 1251–1258. Piscataway, NJ: IEEE.
- Clevert DA, Unterthiner T and Hochreiter S (2015) Fast and accurate deep network learning by exponential linear units (elus). arXiv preprint arXiv:1511.07289.
- Cuka B and Kim D-W (2017) Fuzzy logic based tool condition monitoring for end-milling. *Robotics and Computer-Integrated Manufacturing* 47(C): 22–36.
- Duan J, Shi T, Duan J, et al. (2018) A narrowband envelope spectra fusion method for fault diagnosis of rolling element bearings. *Measurement Science and Technology* 29(12): 125106.
- He K, Zhang X, Ren S, et al. (2016a) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas, USA, 27–30 June 2016, pp. 770–778. Piscataway, NJ: IEEE.
- He K, Zhang X, Ren S, et al. (2016b) Identity mappings in deep residual networks. In: European conference on computer vision, Amsterdam, the Netherlands, 11–14 October 2016, pp. 630–645. Cham, Switzerland: Springer.
- Hoang D-T and Kang H-J (2019) A survey on deep learning based bearing fault diagnosis. *Neurocomputing* 335: 327–335.
- Huang G, Liu Z, Van Der Maaten L, et al. (2017) Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, Honolulu, USA, 21–26 July 2017, pp. 4700–4708. Piscataway, NJ: IEEE.
- Javed K, Gouriveau R, Li X, et al. (2018) Tool wear monitoring and prognostics challenges: a comparison of connectionist methods toward an adaptive ensemble model. *Journal of Intelligent Manufacturing* 29(8): 1873–1890.
- Khan S and Yairi T (2018) A review on the application of deep learning in system health management. *Mechanical Systems and Signal Processing* 107: 241–265.
- Kong D, Chen Y, Li N, et al. (2017) Tool wear monitoring based on kernel principal component analysis and v-support vector regression. *The International Journal of Advanced Manufacturing Technology* 89(1–4): 175–190.
- Kong D, Chen Y and Li N (2018) Gaussian process regression for tool wear prediction. *Mechanical systems and signal processing* 104: 556–574.
- Kong D, Chen Y, Li N, et al. (2019) Relevance vector machine for tool wear prediction. *Mechanical Systems and Signal Processing* 127: 573–594.
- Lei Y, Jia F, Lin J, et al. (2016) An intelligent fault diagnosis method using unsupervised feature learning towards mechanical big data. *IEEE Transactions on Industrial Electronics* 63(5): 3137–3147.
- Lenz J, Wuest T and Westkämper E (2018) Holistic approach to machine tool data analytics. *Journal of manufacturing systems* 48: 180–191.
- Li X, Lim B, Zhou J, et al. (2009) Fuzzy neural network modelling for tool wear estimation in dry milling operation. In: Annual conference of the prognostics and health management society, San Diego, USA, 27 September–1 October 2009, pp. 1–11. USA: The Prognostics and Health Management Society.
- Li Y, Wang X, Si S, et al. (2020) Entropy based fault classification using the case western reserve university data: a benchmark study. *IEEE Transactions on Reliability* 69(2): 754–767.
- Merainani B, Rahmoune C, Benazzouz D, et al. (2018) A novel gearbox fault feature extraction and classification using hilbert empirical wavelet transform, singular value decomposition, and som neural network. *Journal of Vibration and Control* 24(12): 2512–2531.
- Prechelt L (1998) Early stopping-but when? *Neural Networks: Tricks of the Trade*. Berlin, Heidelberg: Springer, 55–69.
- Qiao H, Wang T, Wang P, et al. (2018) A time-distributed spatiotemporal feature learning method for machine health monitoring with multi-sensor time series. *Sensors* 18(9): 2932.
- Ruder S (2016) An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747.
- Seera M, Wong MLD and Nandi AK (2017) Classification of ball bearing faults using a hybrid intelligent model. *Applied Soft Computing* 57: 427–435.
- Szegedy C, Ioffe S, Vanhoucke V, et al. (2017) Inception-v4, inception-resnet and the impact of residual connections on learning. In: 31st AAAI conference on artificial intelligence, San Francisco, USA, 4–9 February 2017, pp. 4278–4284. Palo Alto, CA: AAAI Press.
- Udmale SS, Patil SS, Phalle VM, et al. (2018) A bearing vibration data analysis based on spectral kurtosis and convnet. *Soft Computing* 23: 9341–9359.
- Wang C, Xie Y and Zhang D (2019a) Deep learning for bearing fault diagnosis under different working loads and non-fault location point. *Journal of Low Frequency Noise, Vibration and Active Control*. doi: [10.1177/1461348419889511](https://doi.org/10.1177/1461348419889511).
- Wang J, Mo Z, Zhang H, et al. (2019b) A deep learning method for bearing fault diagnosis based on time-frequency image. *IEEE Access* 7: 42373–42383.
- Wang Y, Dai W and Xiao J (2018) Detection for cutting tool wear based on convolution neural networks. In: 2018 12th international conference on reliability, maintainability, and safety (ICRMS), Shanghai, China, 17–19 October 2018, pp. 297–300. Piscataway, NJ: IEEE.
- Wen L, Li X and Gao L (2020) A transfer convolutional neural network for fault diagnosis based on resnet-50. *Neural Computing and Applications* 32: 6111–6124.

- Xu G, Liu M, Jiang Z, et al. (2019) Bearing fault diagnosis method based on deep convolutional neural network and random forest ensemble learning. *Sensors* 19(5): 1088.
- Yan J, Meng Y, Lu L, et al. (2017) Industrial big data in an industry 4.0 environment: challenges, schemes, and applications for predictive maintenance. *IEEE Access* 5: 23484–23491.
- Yin S, Li X, Gao H, et al. (2015) Data-based techniques focused on modern industry: an overview. *IEEE Transactions on Industrial Electronics* 62(1): 657–667.
- Yu J and Lv J (2017) Weak fault feature extraction of rolling bearings using local mean decomposition-based multilayer hybrid denoising. *IEEE Transactions on Instrumentation and Measurement* 66(12): 3148–3159.
- Zhang S, Zhang S, Wang B, et al. (2019) Machine learning and deep learning algorithms for bearing fault diagnostics—a comprehensive review. arXiv preprint arXiv:1901.08247.
- Zhang W, Li C, Peng G, et al. (2018) A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different working load. *Mechanical Systems and Signal Processing* 100: 439–453.
- Zhao M and Jia X (2017) A novel strategy for signal denoising using reweighted svd and its applications to weak fault feature enhancement of rotating machinery. *Mechanical Systems and Signal Processing* 94: 129–147.
- Zhao R, Yan R, Chen Z, et al. (2019) Deep learning and its applications to machine health monitoring. *Mechanical Systems and Signal Processing* 115: 213–237.
- Zhao X and Jia M (2018) Fault diagnosis of rolling bearing based on feature reduction with global-local margin fisher analysis. *Neurocomputing* 315: 447–464.
- Zhou H, Shi T, Liao G, et al. (2017) Weighted kernel entropy component analysis for fault diagnosis of rolling bearings. *Sensors* 17(3): 625.