Recognition of rolling bearing running state based on genetic algorithm and convolutional neural network

Wanjie Lu¹, Hongpeng Mao¹, Fanhao Lin¹, Zilin Chen², Hua Fu² and Yaosong Xu²

Abstract
In this study, the GA-CNN model is proposed to realize the automatic recognition of rolling bearing running state. Firstly, to avoid the over-fitting and gradient dispersion in the training process of the CNN model, the BN layer and Dropout technology are introduced into the LeNet-5 model. Secondly, to obtain the automatic selection of hyperparameters in CNN model, a method of hyperparameter selection combined with genetic algorithm (GA) is proposed. In the proposed method, each hyperparameter is encoded as a chromosome, and each hyperparameter has a mapping relationship with the corresponding gene position on the chromosome. After the process of chromosome selection, crossover and variation, the fitness value is calculated to present the superiority of the current chromosome. The chromosomes with high fitness values are more likely to be selected in the next genetic iteration, that is, the optimal hyperparameters of the CNN model are obtained. Then, vibration signals from CWRU are used for the time-frequency analysis, and the obtained time-frequency image set is used to train and test the proposed GA-CNN model, and the accuracy of the proposed model can reach 99.85% on average, and the training speed is four times faster than the model LeNet-5. Finally, the result of the experiment on the laboratory test platform The experimental results confirm the superiority of the method and the transplantability of the optimization model.

Keywords
Genetic algorithm (GA), convolutional neural network (CNN), running state recognition, rolling bearing, hyperparameter optimization

Date received: 21 November 2021; accepted: 17 March 2022

Handling editor: Chenhui Liang

Introduction
Rolling bearing is an important part of modern mechanical equipment, and its running state directly affects the working performance of equipment. Statistically, about 30% of the failures of rotating machinery and equipment are caused by rolling bearings. Therefore, fault diagnosis of rolling bearing is of great significance for industrial production.

At present, common fault diagnosis methods include vibration analysis method, thermal analysis method,
and acoustic signal analysis method, etc. The thermal analysis method is often expensive and has poor timeliness. The acoustic signal analysis method can complete the diagnosis task quickly and at low cost. However, this method may be affected by environmental factors, and is suitable for fault diagnosis tasks in a specific environment. Unlike the above two analysis methods, the vibration analysis method can be analyzed in time domain, frequency domain, or time-frequency domain. The vibration signal contains more equipment status information. Especially for bearing components, the early characteristic information of high frequency region is very important for equipment diagnosis and maintenance, and the early fault main characteristic information can be obtained from the vibration signal. Therefore, this paper adopts the vibration analysis method to diagnose the bearing faults.

With the development of deep learning algorithms, it is feasible to perform fault diagnosis by the network model. The industrial field fault data can be collected and input into the network model; after the model training, the efficiency of fault diagnosis in the model will be greatly improved. In contrast, traditional diagnostic techniques have some shortcomings in both accuracy and speed of diagnosis. In recent years, deep learning has developed rapidly in the field of speech recognition and image recognition by unique feature extraction methods. Xiao et al. converted the original time-domain signals into two-dimensional images directly. After gray processing, these images were input into CNN combined with the AMSGrad algorithm to improve the accuracy of rolling bearing fault recognition. Kumar et al. proposed a novel convolutional neural network (NCNN), which added the sparsity to the original cost function of convolutional neural network, and a new triangular cross-entropy function was proposed to calculate the sparsity cost, so as to realize the effective identification of bearing defects under the condition of small samples. Zhang et al. trained the CNN model by using the two-dimensional time-frequency image set obtained from PWVD analysis. The results showed that the proposed model can identify and classify 15 running states of rolling bearings. However, during the establishment of the CNN model, the optimal parameters are selected through six groups of different numbers of convolution kernels. That is to say, manual selection and adjustment of parameters is a troublesome and time-consuming process.

Through the above multiple cases, it can be proved that CNN has high research value in the field of fault recognition, but it also has inherent shortcomings. The structure of the CNN model is complex, and over-fitting and gradient dispersion are prone to occur during the training process. Besides, there are many parameters in the model, and the parameter selection and adjustment highly rely on manual experience. Therefore, it is of practical significance to propose an adaptive optimization method for CNN hyperparameters. To this end, Li proposed a method for CNN hyperparameter optimization based on particle swarm optimization algorithm.

In this study, based on the preprocessing of vibration signals, a genetic algorithm (GA)-based CNN hyperparameter optimization method is proposed to realize the intelligent fault diagnosis of rolling bearings. The proposed GA-CNN model is also verified by sample sets, the results show that this model has high accuracy and training speed. This method can effectively find the best parameters of the model and deploy them on the intelligent operation and maintenance platform. With the accumulation of industrial field data, the network model is continuously improved to achieve an accurate diagnosis.

Optimization of CNN structure

Convolutional neural network

The convolutional neural network (CNN) is a multi-layer neural network with advantages of local connection and weight sharing. Its incomplete connection and weight sharing network structure reduces the complexity of the network model and reduces the number of weights. Typical convolutional neural networks are usually composed of different numbers of convolutional layers, pooling layers, full connection layers, and other layers.

The core of the CNN convolution layer is the convolution kernel, and the principle is the same as the filter. The mathematical model of convolution operation is as follows:

\[
X^l_j = f \left( \sum_{i \in M_j} X^{l-1}_i \cdot W^{l}_{ij} + b^l_j \right)
\]

where 
- \(X^{l-1}_i\) and \(X^l_j\) are the input and output of the \(l^{th}\) layer neurons;
- \(W^{l}_{ij}\) and \(b^l_j\) represent the weight between the \(i^{th}\) neuron in the \(l^{th}\) layer and the \(j^{th}\) neuron in the \((l+1)^{th}\) layer, paranoid term and the number of network layers;
- \(i\) and \(j\) are two adjacent neurons; \(f\) is the activation function.
The pooling layer of the CNN is also known as the down-sampling layer, which is used to reduce the dimension of feature graph. The mathematical model of pooling operation is as follows:

$$p^{(i,j)} = \max_{(j-1)w + 1 \leq j \leq jw} \{ a^{(i,j)} \}$$

where $$p^{(i,j)}$$ represents the output characteristic value after the pooling operation; $$a^{(i,j)}$$ represents the activation value of the $$i^{th}$$ neuron in the $$l^{th}$$ feature map of layer; $$w$$ is the width of the window for pooling.

The CNN full connection layer classifies the extracted features and expands the output of the pooling layer of the previous layer as a one-dimensional vector. The mathematical model of full connection operation is as follows:

$$y^{(l+1)}(j) = \sum_{i=1}^{n} W^l_{ij} x^{(l)}(i) + b^l_j$$

where $$y^{(l+1)}(j)$$ represents the output value of the $$j^{th}$$ neuron at $$(l+1)$$ layer; $$x^{(l)}(i)$$ is the output value of the $$i^{th}$$ neuron in the $$l^{th}$$ layer.

Optimization of model structure

There are many classical models of convolutional neural networks. From the LeNet-5, AlexNet, VGGNet, and ResNet, the number and complexity of network layers increase with the increasing demand for feature extraction capability. In addition, due to the limited amount of data, over-fitting and gradient dispersion are prone to occur in the training process of the CNN model. To this end, the overlapping sampling method is used, and the batch normalization (BN) layer and Dropout processing technology are introduced in this study to avoid the above problems.

The addition of the BN layer can effectively prevent gradient dispersion and gradient explosion, and greatly improve the training speed of network.

During each CNN training, Dropout processing can make some neurons fail randomly, increasing the diversity of the model. As a result, the over-fitting of the CNN model is effectively avoided and the robustness is improved. The visualization of Dropout technology is shown in Figure 1. Each circle represents a neuron, where the red dotted line is inactivated and the dotted arrow represents there is no nerve conduction.

The hidden layer structure of the optimized CNN model is C1-BN2-S3-C4-BN5-S6-F7(Dropout)-F8(Dropout). That is to say, a BN layer is added after each convolutional layer and Dropout processing is performed after the full connection layer. Figure 2 shows the optimized CNN structure.

CNN hyperparameter optimization based on GA

Genetic algorithm

Genetic Algorithm (GA) is a random search method that simulates natural selection and population genetic. Just like the survival of the fittest rule in nature, GA inherits high-quality genes to offspring through eliminating inferior genes and inheriting high-quality genes, so as to obtain the optimal offspring genes repeatedly. The algorithm has good global optimization performance.

Compared with manual parameter selection and adjustment, genetic algorithm is faster and more accurate. The GA model can be described as follow:

$$GA = (F, P, M, Pn, Pc, Pm, T)$$

where $$F$$, $$P$$ and $$M$$ represent the individual fitness function, the initial population, and the population quantity; $$Pn$$, $$Pc$$, $$Pm$$, and $$T$$ represent the selection operator, the crossover operator, the mutation
operator, and the number of evolution generations respectively.

Hyperparameters optimization process of CNN

Before the CNN hyper-parameter optimization, hyperparameters are encoded in a form similar to human chromosomes by binary coding. In other words, each hyperparameter has a mapping relationship with the corresponding gene loci on the chromosome. Each chromosome represents a CNN hyperparameter combination. After many times of genetic evolution, the chromosome containing the optimal gene is obtained.

The hyperparameter optimization process based on GA is as follows:

Step 1: Population initialization. A random value is generated as the initial value within the value range of all hyperparameters that need to be optimized. When all hyperparameters are searched in this operation, the population initialization is finished.

Step 2: Population evaluation. In this step, the fitness value of the individuals in the population to the objective function is calculated. In this study, population evaluation refers to the accuracy of the bearing state recognition of the CNN model constructed by the combination of different hyperparameters. The fitness value is defined as follows:

\[
\text{Fit} = f(X_1, X_2, X_3, X_4 \ldots X_9)
\]  

where \(f\) is the target function; \(\text{Fit}\) is the fitness value corresponding to an individual; \(X_i\) is the independent variable of the fitness value function.

Step 3: Operator selection. In this step, the common roulette model is adopted, and individuals with higher fitness values in the population are more likely to be selected. The probability of being selected is marked as \(P_n\), and the selection operator can be expressed as follows:

\[
P_n = \frac{\text{Fit}(i)}{\sum_{i=1}^{M} \text{Fit}(i)}
\]

where \(M\) is the quantity of population; \(\text{Fit}(i)\) is the fitness values of individuals in the population.

Step 4: Operator crossover. The individuals selected in the previous step are crossed according to a specific operator probability \(P_c\) to determine whether the chromosomes cross and generate new individuals.

Step 5: Operator mutation. Each chromosome is randomly mutated at the same gene locus according to the preset mutation probability \(P_m\).

Step 6: Update chromosomes. The fitness value of the new chromosome is recalculated.

Step 7: Stop judging. In this step, whether evolution generation \(T\) reaches the preset maximum is judged. If not, return to Step 3 and enter the next iteration.

Step 8: The optimal hyperparameter solution is obtained, and the CNN model can be built with optimized hyperparameters. The validity of the model can be verified by the follow-up experiment.

The flow of the genetic algorithm is shown in Figure 3.

Recognition steps of rolling bearing running state

Figure 4 shows the identification method of the rolling bearing running state based on the GA-CNN model.

Step 1: Classify the running states of rolling bearings and obtain the original vibration signal data under different running states.

Step 2: Adopt the time-frequency analysis method of short-time Fourier Transform (STFT) to establish a time-frequency graph sample set, including a training set and a test set.

Step 3: Optimize the structure of the CNN model.
Step 4: Determine the hyperparameters to be optimized in the CNN, and conduct automatic optimization of parameters through the GA algorithm.

Step 5: Build the CNN model with optimized hyperparameters and conduct the CNN model training and testing.

Step 6: Compare with the classical LeNet-5 model and verify the effectiveness of the proposed method.

Experimental verification

Classification of rolling bearing running state and acquisition of original data

The data from the Bearing Data Center of Case Western Reserve University (CWRU) contains bearing data under multiple operating conditions and various operating conditions, and the amount of data is relatively sufficient, which is very suitable for the state recognition of bearing operating based on deep learning. Therefore, the data sets used in this study are all from the Bearing Data Center of CWRU in the United States. Figure 5 shows the experimental platform. The bearing data of the drive end of the platform is selected as experimental data. The bearing model is a deep groove ball bearing SKF6205, the load state is HP1, the motor speed is 1772 r/min, and the sampling frequency of the signal acquisition device is 12 kHz.

Due to the limited amount of data in the Bearing Data Center of CWRU, overlapping sampling is adopted in this study to ensure the sample size of model training and test.

The selected data set contains four common operating states of rolling bearings, namely normal condition (NC), inner ring failure (IF), outer ring failure (OF), and rolling element failure (BF). The damage lengths of the three failure states are divided into 0.18, 0.36, and 0.54 mm. In addition, the damage directions of outer ring failure of the bearing are divided into three directions of 6 o’clock, 3 o’clock, and 12 o’clock. Finally, there are 16 types of bearing running states, as shown in Table 1. A total of 300 experimental samples are selected for each state, each sample is composed of 1024 data points, and the sample set is divided into the training set and the test set at a ratio of 5:1.

Establishment of rolling bearing data set

In this study, the Short-Time Fourier transform (STFT) analysis method is adopted in the production method of the time-frequency image sample set. The basic idea of the STFT is to segment the non-stationary signal into countless short-time sequences, and then perform each segment of the time sequence. The Fourier transform formula is as follows:

$$STFT_x(t, \omega) = \int_{-\infty}^{\infty} x(t) r(t - \tau) e^{-j\omega \tau} d\tau$$ (7)

where $x(t)$ represents the original signal; $r(t)$ represents a window function.

The generated time-frequency diagram is trimmed to remove the coordinate information and retain only time-frequency diagram information. Due to the limited space, the time-frequency images of four types of
bearing running states are displayed corresponding to NC, IF, OF, and BF states, as shown in Figure 6(a)–(d). Although the color image has richer feature information, a large amount of data processing is also required. Therefore, the image is grayed, and the imresize function is used to reconstruct the image size into a $28 \times 28$ grayscale image. Eventually, the final sample of the time-frequency image sample set is obtained, as shown in Figure 7. The original time-frequency images corresponding to NC, IF, OF and BF are shown in Figure 7.

### Experimental process and results analysis

#### Setting of experimental parameters

In the genetic algorithm, the population number $M$ is 50, the genetic evolution generation number $T$ is 6, the crossover probability $P_c$ is 0.3, and the mutation probability $P_m$ is 0.1.

Table 2 shows the CNN hyperparameters to be optimized and their searching range. The size and number of convolution kernels have a direct impact on the performance of the CNN model. At the same time, the activation function type of the activation layer, the type

### Table 1. Running states of bearing.

| Number | Working condition       | Damage size/mm | Number of training (testing) samples |
|--------|-------------------------|----------------|-------------------------------------|
| 1      | Normal                  | -              | 250 (50)                            |
| 2      | Inner ring failure      | 0.18 mm        | 250 (50)                            |
| 3      | Inner ring failure      | 0.36 mm        | 250 (50)                            |
| 4      | Inner ring failure      | 0.54 mm        | 250 (50)                            |
| 5      | Inner ring failure      | 0.72 mm        | 250 (50)                            |
| 6      | Outer ring failure (6 o'clock) | 0.18 mm | 250 (50)                            |
| 7      | Outer ring failure (3 o'clock) | 0.18 mm | 250 (50)                            |
| 8      | Outer ring failure (12 o'clock) | 0.18 mm | 250 (50)                            |
| 9      | Outer ring failure (6 o'clock) | 0.36 mm | 250 (50)                            |
| 10     | Outer ring failure (6 o'clock) | 0.54 mm | 250 (50)                            |
| 11     | Outer ring failure (3 o'clock) | 0.54 mm | 250 (50)                            |
| 12     | Outer ring failure (12 o'clock) | 0.54 mm | 250 (50)                            |
| 13     | Rolling body failure    | 0.18 mm        | 250 (50)                            |
| 14     | Rolling body failure    | 0.36 mm        | 250 (50)                            |
| 15     | Rolling body failure    | 0.54 mm        | 250 (50)                            |
| 16     | Rolling body failure    | 0.72 mm        | 250 (50)                            |
of the optimizer, the size of the learning rate, the size of the batch, and the loss rate of the dropout layer are all important to the training progress of the CNN model. Therefore, the above hyperparameters are selected to be optimized. The searching range of each hyperparameter is expanded as far as possible on the basis of mainstream settings to seek the optimal solution.

**Analysis of results.** After the genetic algorithm optimization experiment, the obtained data are shown in Table 3. The top three hyperparameter combinations with the highest identification accuracy is listed as follows.

There are three GA-CNN models combined with hyperparameters achieving high recognition accuracy for rolling bearing running state. It indicates that the proposed method is effective for the automatic selection of model hyperparameters. As shown in Table 3, the convolutional kernel of $6 \times 6$ is not selected for the top three models, because the large size of the convolutional kernel can induce noise and reduce the accuracy of classification. In addition, two activation functions with better performance, Relu and Tanh, are screened out through hyperparameter optimization. It can be seen that the first two hyperparameter combinations with the highest recognition accuracy are the results of the sixth generation of evolution, and the accuracy of the CNN model recognition will be improved with the increasing number of evolution generations.

As shown in Figure 8, the abscissa is the number of population evolution generations, and the ordinate is the recognition accuracy. The blue prismatic data points represent the average recognition accuracy of the current evolutionary generation, and the red square data points represent the highest recognition accuracy of the current evolutionary generation. It can be seen that with the continuous evolution of the population, the average recognition accuracy of the CNN model increases greatly. It indicates that the accuracy of the CNN model recognition will be improved with the increasing number of evolution generations.

**Analysis of comparative experiment results.** To verify the effectiveness of the proposed method, the performance of the GA-CNN model is compared with that of the LeNet-5 model. The network parameters of both models are set as follows: The maximum number of iterations is set to 50, BatchSize to 8, the learning rate to 0.001, and Dropout to 0.3. As shown in Figure 9, the blue diamond markers and curves represent the recognition accuracy of the training set and test set of the GA-CNN model; the red square markers and curves represent the recognition accuracy of the training set and test set of the LeNet model, respectively. Similar to

| Table 2. Optimization parameter set. |
|-------------------------------------|
| Hyperparameter                      | Parameter selection range |
|-------------------------------------|
| C1 convolution kernel size          | $3 \times 3, 4 \times 4, 5 \times 5$ |
| C1 convolution kernel number        | 8, 16, 32, 64, 128          |
| C4 convolution kernel size          | $3 \times 3, 4 \times 4, 5 \times 5$ |
| C4 convolution kernel number        | 8, 16, 32, 64, 128          |
| Activation function type            | Relu, Tanh, Sigmoid, Linear |
| Optimizer type                      | SGD, Adam, RMSprop, Adadelta |
| Learning rate                       | 0.0001–0.005                |
| Dropout                             | 0.1–0.5                     |
| Batch size                          | 8, 16, 32, 64               |

| Table 3. Hyperparameter combinations with highest accuracy. |
|-------------------------------------------------------------|
| Hyperparameter                  | Top 1 | Top 2 | Top 3 |
|---------------------------------|-------|-------|-------|
| C1 convolution kernel size      | $5 \times 5$ | $5 \times 5$ | $4 \times 4$ |
| C1 convolution kernel number    | 128   | 128   | 128   |
| C4 convolution kernel size      | $3 \times 3$ | $4 \times 4$ | $3 \times 3$ |
| C4 convolution kernel number    | 32    | 64    | 64    |
| Activation function type        | Relu  | Tanh  | Relu  |
| Optimizer type                  | Adam  | Adam  | Adam  |
| Learning rate                   | 0.001 | 0.0005| 0.0005|
| Dropout                         | 0.3   | 0.2   | 0.3   |
| Batch size                      | 8     | 8     | 8     |
| Accuracy                        | 100   | 99.87 | 99.50 |
| Evolution generations           | 6     | 6     | 5     |

![Figure 8. Recognition accuracy curve.](image)
the markings in Figures 9 and 10 represents the change curve of the loss function during the training and testing of the two models. Through the comparison, it can be seen that the proposed GA-CNN model has higher recognition accuracy and faster loss function convergence than those of the LeNet-5 model.

Figure 11 shows the confusion matrix plot for the experimental test set. The predicted fault type label is taken as the abscissa, and the actual fault type label is taken as the ordinate.

It can be seen from Figure 11 that the identification accuracy of the bearing state of label 9 is 94% (another 6% probability of being mistaken as label 13), and the identification accuracy of the other bearing states is 100%. After 10 experiments, the average recognition accuracy is 99.85%. Therefore, the proposed method for running state identification of rolling bearings is verified to be extremely effective.

Visualization of training process. To intuitively observe the training effect of each network layer of the GA-CNN model, the T-distributed stochastic neighborhood embedding (T-SNE) method is used to reduce the dimension of the features of some network layers and visualize them in the form of three-dimensional scatter plots, as shown in Figure 12.

Due to the limited space, the dimensionality reduction and visualization processing on the features of input layers, convolutional layers, and output layers are performed, as shown in Figure 12(a)–(d). The three coordinate axes respectively represent the first three dimensions after dimension reduction of features extracted from the current network layer. Each point represents a sample, and each color represents a bearing operating state, a total of 16 operating states. The specific correspondence between states and colors is shown in Figure 12(d).

Through the dimensionality reduction and the visualization processing of the input layer features, it can be seen that many sample points representing different bearing states overlap, resulting in the unidentifiable operating states, as shown in Figure 12(a).

After the dimensionality reduction and the visualization processing of features of convolutional layers, compared with the input layer, the clustering effect of
the sample points in the same state becomes more obvious, as shown in Figure 12(b). It indicates that after the feature extraction of the convolutional layer, the distinguishability of the features has been further enhanced.

Finally, the above method is also used for the output layer. Intuitively, sample points of different bearing states have a higher degree of discrimination, and there are no overlapping sample points. It suggests that the GA-CNN network model has a very good effect on bearing state recognition. As shown in Figure 12(c), the proposed method proposed in this study can effectively extract the feature information in the image and accurately identify the running state of the rolling bearing.

### Model transferability verification

In order to verify the superiority of the method and the portability of the optimization model in this paper, the mechanical transmission platform built by our team will be used for verification.

Bearings of different faults can be replaced on this platform to measure vibration signals of rolling bearings in different fault states (the bearing model is N205, which is different from the bearing SKF6205 in the Case Western Reserve University dataset). Three fault states are artificially created plus normal working state, so four kinds of bearing state data are collected through data acquisition system. The data acquisition system consists of NI cDAQ-9189 data acquisition card, NI-9234 sound and vibration input module, YD-186 piezoelectric sensor installed in the horizontal and vertical directions of the bearing housing. The connection of the test platform and the three kinds of fault bearing are shown in Figures 13–15:

Firstly, four kinds of bearing state data are collected through the data acquisition system, including normal state, inner ring fault state, outer ring fault state, and
rolling element fault state. Using the same data processing method as before in this paper, the vibration signals are converted into time-frequency image, and divided into training set and test set. Since this is a process of model migration, the parameters of the convolutional layer are locked, only the parameters of the fully connected layer are trained. This method can greatly speed up the training speed of the model while maintaining good recognition performance. Finally, the test set is input into the model, and the test set confusion matrix is obtained and shown as Figure 16. The recognition accuracy of the four running states can reach 100%.

Conclusions

(1) In this study, the structure of the classical CNN model is optimized by adding BN layers and Dropout technology to avoid over-fitting and improve the network training speed by at least four times.

(2) Genetic algorithm is used to automatically optimize 7 kinds of parameters in the CNN model, and the optimization range is the mainstream setting of corresponding parameters. This method can search the optimal hyperparameter more quickly and accurately, which greatly reduces the difficulty of CNN model design.

(3) The CNN model constructed by the optimized hyperparameters can be used to identify the running state of rolling bearings. The average recognition accuracy of the proposed CNN model can reach 99.85%, and its training speed is four times faster than the classical CNN model LeNet-5.

(4) The transferability of the model is verified, and the accuracy of bearing state recognition reaches 100%, which verifies that the optimized model has a good transferability. In the follow-up research work, our team will continue the further study of transfer learning and try to apply the proposed method to the fault diagnosis of gearboxes, motors, etc.

(5) A hyperparameter optimization method of deep learning model applied to bearing fault diagnosis is proposed, which can find the best parameters of the model effectively. In a specific application environment, the trained network model can be deployed on the intelligent cloud platform to accurately diagnose the current running status of the device.

Declaration of conflicting interests

The author(s) 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
This work was financially supported by Basic Scientific Research project of Universities in Liaoning Province (LJKZ0352).

ORCID iD
Wanjie Lu https://orcid.org/0000-0003-4599-1920

References
1. Glowacz A, Tadeusiewicz R, Legutko S, et al. Fault diagnosis of angle grinders and electric impact drills using acoustic signals. Appl Acoust 2021; 179: 108070.
2. Li T, Duan L-X and Zhang D-N. Application of adaptive convolutional neural network in fault diagnosis of rotating machinery. Journal of Vibration and shock 2020; 39: 275–282.
3. Ming A. Feature extracting method in the rolling element bearing fault Diagnosis: Spectrum auto-correlation. J Mech Eng 2012; 48: 65–71.
4. Li S.-M., Guo H-D and Li D.-R. Summary of vibration signal processing methods. Chin J Sci Instrum 2013; 34: 1907–1915.
5. Pan P-W, Liang S-Y and Li J-T. Fault feature extraction of wind turbine rolling bearing based on complex wavelet multi-scale envelope analysis. Proc CSEE J 2015; 35: 4147–4152.
6. Hu A-J, Ma W-L and Tang G-J. Rolling bearing fault feature extraction method based on integrated empirical mode decomposition and kurtois criterion. Proc CSEE J 2012; 32: 106–111.
7. Zheng J. Improved Hilbert-Huang Transform and its applications to rolling bearing fault diagnosis. J Mech Eng 2015; 51: 138–145.
8. Lei Y-G, Jia F and Kong D.-T. Opportunities and challenges of machinery intelligent fault diagnosis in Big Data Era. J Mech Eng 2018; 54: 94–104.
9. Deng J, Xu X, Zhang Z, et al. Universum autoencoder-based domain adaptation for speech emotion recognition. IEEE Signal Process Letters J 2017; 24: 500–504.
10. Zhang L, Zuo W and Zhang D. LSDT latent sparse domain transfer learning for visual adaptation. IEEE Trans Image Process 2016; 25: 1177–1191.
11. Qu J-L, Yu L and Yuan T. An adaptive fault diagnosis algorithm for rolling bearings based on one-dimensional convolutional neural network. Chin J Sci Instrum 2018; 39: 134–143.
12. Chen G. Feature extraction and intelligent diagnosis of early faults of rolling bearings. Acta Aeronaut Astronaut Sinice J 2009; 30: 362–367.
13. Gong W-F, Chen H and Zhang Z-H. Research on Intelligent Fault diagnosis of rolling bearing based on improved convolutional neural network. Chin J Sci Instrum 2020; 33: 400–413.
14. Zan T, Wang H and Liu Z.-H. Rolling bearing fault diagnosis model based on multi-input layer convolutional neural network. J Vibr Shock 2020; 39: 142–149.
15. Chen B-J, Chen X-L and Shen B.-M. Application of CNN-LSTM deep neural network in fault diagnosis of rolling bearings. J Xi’an Jiaotong Univ 2021; 55: 28–36.
16. Xiao X, Wang J-X and Zhang Y-J. A two-dimensional convolutional neural network optimization method for bearing fault diagnosis. Proc CSEE J 2019; 39: 4558–4568.
17. Yuan J-H, Han T and Tang J. Intelligent fault diagnosis method of rolling bearing based on wavelet time-frequency diagram and CNN. Mach Des Res J 2017; 33: 93–97.
18. Kumar A, Vashishtha G, Gandhi CP, et al. Novel convolutional neural network (NCNN) for the diagnosis of bearing defects in rotary machinery. IEEE Trans Instrum Meas 2021; 70: 1–10.
19. Zhang A-A, Huang J-Y and Ji S-W. Bearing fault pattern recognition based on image classification of convolutional neural network. J Xi’an Jiaotong Univ 2020; 39: 165–171.
20. Li J. Research on optimization of convolutional neural network based on particle swarm algorithm. Comput Digit Eng J 2020; 48: 2452–2457.
21. https://engineering.case.edu/bearingdatacenter/download-data-file.