



Optimal Modifications in CNN for Bearing Fault Classification and Adaptation Across Different Working Conditions

Diwang Ruan¹ · Feifan Zhang² · Luxi Zhang² · Jianping Yan³

Received: 23 November 2022 / Revised: 24 May 2023 / Accepted: 30 July 2023 / Published online: 1 September 2023
© Krishtel eMaging Solutions Private Limited 2023

Abstract

Introduction Convolutional neural network (CNN) has been widely used in bearing fault diagnosis and many satisfying results have been reported. As a typical CNN network, the LeNet-5 was improved from three aspects to further enhance its diagnosis performance in this paper.

Methods Firstly, eight hyperparameters were optimized by particle swarm optimization within the predefined discrete parameter value sets. Secondly, envelope spectrum and feature vector were adopted as replacements for the original signal input. The feature vector consisted of 157 manually extracted features from time and frequency domains. Thirdly, support vector machine, decision tree and random forest were applied to replace the default fully connected layer. An overall evaluation method was also proposed in terms of classification accuracy, stability, robustness to noise and computing efficiency.

Experiments Based on the Case Western Reserve University bearing dataset, two experimental cases were designed from four different working loads. In case 1, the training and test datasets of each load were individually collected from the corresponding working load. Based on the overall evaluation method introduced, the optimal modification methods were identified in terms of hyperparameters, input type and fully connected layers. The contributions of modification to CNN in the performance improvement were quantitatively discussed and compared. In case 2, the optimized CNN was trained with the dataset from one working load and tested with the other three different working loads, which resulted in a sharp reduction of accuracy. To address this problem, multi-convolutional layer, data augmentation and signal concatenation were proposed and adopted individually as well as collaboratively to improve the CNN's ability in the working condition adaptation.

Conclusion Experimental results confirmed that all the three approaches effectively enhanced the CNN's performance. The combination of two or three approaches has better performance than the individual one.

Keywords Bearing fault diagnosis · Convolutional neural network · Data augmentation · Signal concatenation · Working condition adaptation

This research is supported by CSC doctoral scholarship (201806250024) and Zhejiang Lab's International Talent Fund for Young Professionals.

✉ Diwang Ruan
diwang.ruan@campus.tu-berlin.de

Feifan Zhang
zhangfeifan20@gmail.com

Luxi Zhang
luxi.zhang@campus.tu-berlin.de

Jianping Yan
jianping.yan@zju.edu.cn

Introduction

With the rapid development of modern industries, there is an increasing demand for higher safety and reliability of intelligence and integration mechanical systems. The prognostic and health management (PHM) technology, as a promising

¹ Chair of Electronic Measurement and Diagnostic Technology, TU Berlin, 10587 Berlin, Germany

² School of Electronic Engineering and Computer Science, TU Berlin, 10587 Berlin, Germany

³ School of Aeronautics and Astronautics, Zhejiang University, Hangzhou 310027, China

approach to meet the above demands, has been receiving increasing research attention in recent years [1]. As a fundamental support component in rotating machines, the performance of rolling bearings directly affects the reliability of equipment. Its failures may result in huge damage, economic loss and human safety. Therefore, reliable fault diagnosis and predictive maintenance of bearings are meaningful and practical. The machine learning and deep learning [2, 3] as typical data-driven methods for effective intelligent fault diagnosis have been established and attracting more and more attention from both academia and industry. Among the various deep learning networks, CNN has been the most used due to its powerful ability in feature extraction and nonlinear mapping, many satisfying results have been achieved [4, 5]. Most published researches only focus on the performance improvement [6, 7], but very few works study the impacts of hyperparameters, feature extraction or structure modification. To study these impacts is the first task of this research. To date, most published works for the CNN-based fault diagnosis are based on such prerequisite that the training and test datasets are from the same distribution. For example, the training and test datasets are required to come from the same bearing test bench under the same working condition [8, 9]. If the test dataset does not appear in the training phase, the CNN's performance on test dataset is usually sharply reduced [10]. However, in practical industrial applications, CNN certainly has to deal with the measurement data that has never appeared in the training process [11]. For example, a bearing test bench usually works under various conditions and probably even under a transient cycle [12, 13]. Only limited data under certain conditions are available in the training process. Therefore, when trained with the data from one condition but tested under other different conditions, how to guarantee the CNN's performance has become a hurdle. This defines the second motivation for this research.

To address the two research gaps mentioned above, the LeNet-5 is chosen as a CNN benchmark in this study. The influence of hyperparameter, input and fully connected layer on the performance of CNN is discussed, and the optimal modifications from these three aspects are identified. Firstly, the particle swarm optimization (PSO) is applied to optimize the hyperparameter of CNN. Secondly, the different features from both time and frequency domains are extracted and fed into CNN rather than the commonly used original signal. Thirdly, the fully connected layer is replaced by machine learning methods to further improve the CNN's accuracy. An overall evaluation method is proposed to determine the best modification in terms of classification accuracy, stability, robustness to noise and computing efficiency. The optimized CNN is compared with the traditional one. With respect to the working condition adaptation, three different methods, namely the multi-convolutional layers, data augmentation and signal concatenation, are proposed to

enhance the CNN's performance. These three approaches are explored individually as well as collaboratively, and their performances are compared and discussed in details. The experimental data from the Case Western Reserve University is used as the data source. Validation results confirm the effectiveness of the proposed methods.

The remainder of this paper is organized as follows. “**Test bench**” describes the test bench description and data processing. “**Methodology of modification**” presents the methodology of PSO, feature extraction, modification of fully connected layer and four indicators for an overall evaluation. “**Case study and results analysis**” validates the modified CNN with two design cases. In case 1, the training and test datasets are collected from the same working condition, while in case 2, the CNN is verified with the test datasets that differ from the training one. “**Case study and results analysis**” introduces the proposed methods to solve the problem in the working condition adaptation. “**Conclusion**” concludes the whole paper.

Test Bench

Experimental Setup

The bearing experimental data used is taken from the CWRU bearing dataset center [14]. The test bench shown in Fig. 1 comprises of a motor, a torque transducer/encoder, a dynamometer, control electronics and test bearings that support the motor shaft. Accelerometers are attached to the housing to collect the vibration data. There are three failure types (ball fault, inner race fault and outer race fault) aside from normal bearing. Each failure type has three different fault diameters (0.007 in., 0.014 in. and 0.021 in.) and four different load states [0 HP (horsepower), 1 HP, 2 HP and 3 HP]. There are totally 10 types of different bearing conditions. The drive end fault data with a sampling frequency of 12 kHz is used to validate the proposed CNN described in the following text.

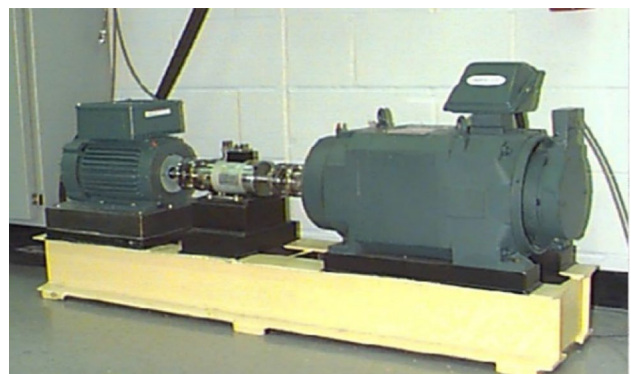


Fig. 1 CWRU bearing test bench [14]

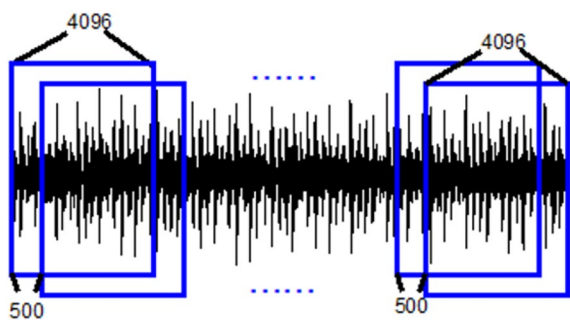


Fig. 2 Data segmentation with overlap

Table 1 Bearing data description

Bearing condition	Number of samples	Fault size	Training set	Test set	Labels
Normal	2401	–	1689	712	0
Ball fault	2822	0.007	629	309	1
		0.014	654	288	2
		0.021	599	243	3
Inner race fault	2822	0.007	664	278	4
		0.014	646	294	5
		0.021	662	278	6
Outer race fault	2589	0.007	660	281	7
		0.014	635	306	8
		0.021	505	202	9

Data Processing

The whole dataset needs to be rescaled into the range of $[-1, 1]$ and then split into small frames. Each frame is taken as a sample. Due to the limited data provided, the dataset is divided with overlap [15]. Supposed the total length of the original signal is L , the length of each small frame is l and the shift between two data frames is τ , then the number of data frames n can be computed as follows:

$$n = \text{floor} \left[\left(\frac{L - \tau}{l} \right) + 1 \right]. \tag{1}$$

In this work, l, τ are 4096 and 500 respectively. L differs from dataset to dataset. The approach is illustrated in Fig. 2. Subsequently, all the samples are split into the training and test datasets with a ratio of 7:3. The size of the training and test datasets under each label is summarized in Table 1.

Methodology of Modification

As outlined in Fig. 3, the CNN modifications are achieved from three approaches. Firstly, eight hyperparameters are optimized by PSO within the predefined discrete value sets. Then, three kinds of input (original signal, envelope spectrum and feature vector) are fed into the CNN individually, and their performances are compared to each other. Finally, the default fully connected layer in the CNN are replaced by the machine learning methods that have stronger ability in classification. The optimal modifications in each step are identified and then combined to build an optimal CNN structure, which is further

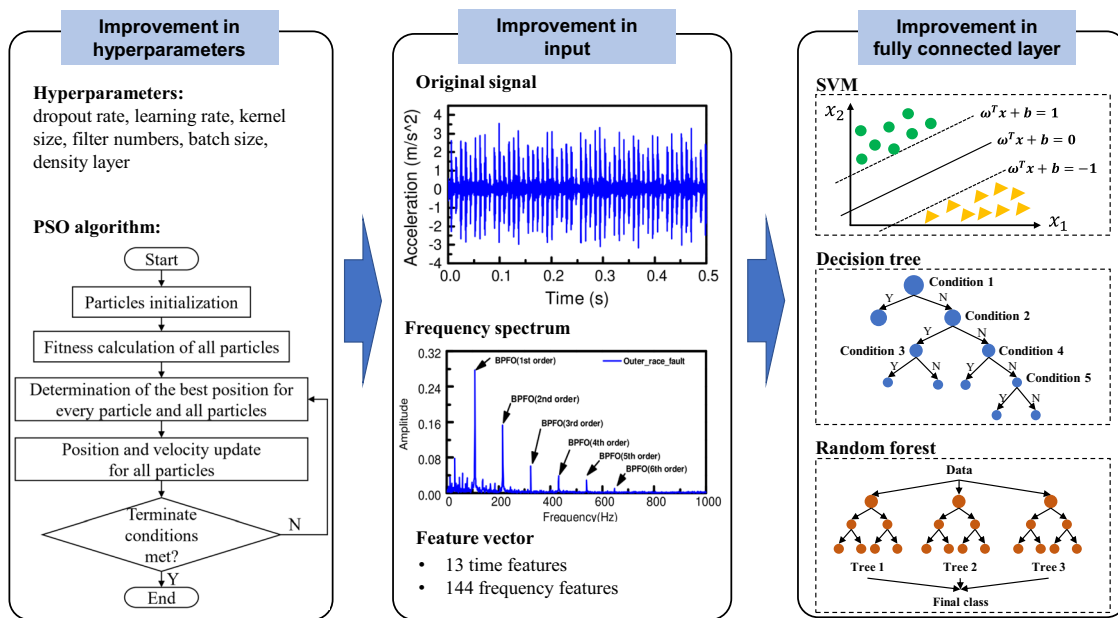


Fig. 3 Improvements of original LeNet-5 from three approaches

verified under two cases for the bearing fault diagnosis. The modification details are described in the following subsections.

Introduction of LeNet-5

LeNet-5 is one of the most frequently used CNN structures and has achieved great success in the automatic handwritten digit classification [16]. It consists of two sets of convolutional and pooling layers, one flattening convolutional layer, two fully-connected layers and one softmax classifier [17]. As shown in Fig. 4, the LeNet-5 structure is slightly modified and adopted as a representative of CNN in this study.

Particle Swarm Optimization

Besides the network structure, the hyperparameters have much influence on the CNN's performance. Therefore, PSO is used to find the best set of hyperparameters like dropout rate, learning rate, kernel size, number of filters, batch size and size of density layer. In the PSO algorithm, N individual particles search within a D -dimensional space to find the best solution to fitness function. Each particle has its own velocity and position that are updated after each iteration. The update of velocity and position of each particle i at the $(k + 1)$ th iteration is influenced by the global best position of all particles g^{best} as well as the best position of each particle p^{best} . This can be described by the following equations [18]:

$$v_{ij}(k+1) = \omega v_{ij}(k) + c_1 r_1 [p_{ij}^{\text{best}}(k) - x_{ij}(k)] + c_2 r_2 [g_j^{\text{best}}(k) - x_{ij}(k)], \quad (2)$$

$$x_{ij}(k+1) = x_{ij}(k) + v_{ij}(k+1), \quad (3)$$

where k denotes the number of iteration, v_{ij} the velocity of the i th particle in the j th dimension, x_{ij} the position of the i th particle in the j th dimension. p^{best} is an $i \times j$ matrix, whilst

p_{ij}^{best} is the best position of the i th particle in the j th dimension. g^{best} is a j -dimensional vector, while g_j^{best} is the global best position of all particles in the j th dimension. ω is the inertia coefficient, c_1 and c_2 are the learning factors, r_1 and r_2 are random numbers generated from the uniform distribution in range of $[0, 1]$ [19].

Eight hyperparameters in CNN are optimized by the PSO algorithm. The fitness function is set as the accuracy in test set, with the target to find the maximum of fitness function. To achieve a trade-off between accuracy and efficiency for optimization, the search space of each hyperparameter is restricted to a much smaller but reasonable range based on both theory and experience [20, 21], as demonstrated in Table 2. The PSO parameters are summarized in Table 3.

Feature Extraction

Although CNN has quite strong ability in extracting hidden features from data [22], many researchers still prefer to process the input before fed into CNN, enabling CNN to achieve higher performance [23–25]. For comparison with the original input, two kinds of extracted feature are proposed. The first one is the envelope spectrum, the other one is the feature vector consisting of 157 features extracted from time and frequency domains. The detailed process is presented hereafter.

Envelop Spectrum

Concerning features in frequency domain for the fault diagnosis, the envelope spectrum has been verified to be highly effective [26]. Theoretically, two steps are necessary to get this for the input signal: first is to obtain the signal envelop and then to conduct the frequency transformation. The Hilbert transform is adopted to capture the signal envelope. To this end, the input is decomposed by the empirical mode decomposition (EMD) into a finite

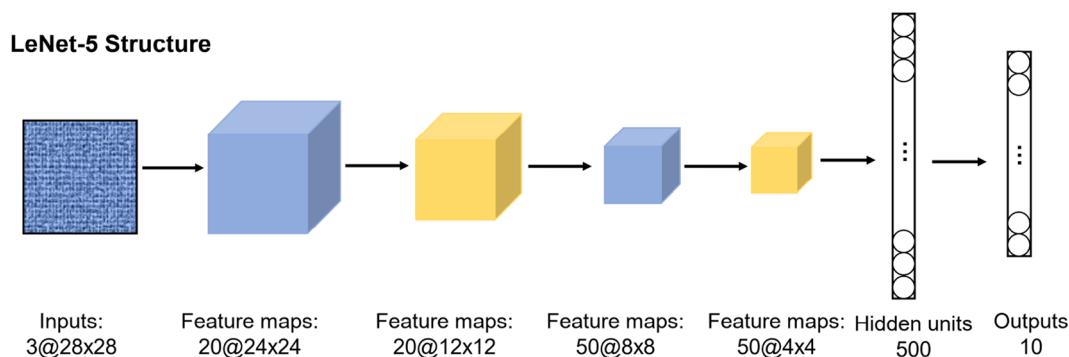


Fig. 4 LeNet-5 structure

Table 2 Hyperparameter range of CNN

Hyperparameters	Range
Drop rate	0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85
Learning rate	0.0001, 0.00015, 0.0002, 0.00025, 0.0003, 0.00035, 0.0004, 0.00045, 0.0005 0.00055, 0.0006, 0.00065, 0.0007, 0.00075, 0.0008, 0.00085
Kernel size of the 1st layer	3×3, 5×5, 7×7
Kernel size of the 2nd layer	3×3, 5×5, 7×7
Number of filters in the 1st layer	16, 32, 64, 128, 256
Number of filters in the 2nd layer	16, 32, 64, 128, 256
Batch size	50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 100, 105, 110, 115, 120, 125
Dense layer (flatten)	128, 256, 512, 1024, 2048

Table 3 Parameters of PSO

Parameter	Value
Number of particles	20
c_1	0.4
c_2	0.6
ω	0.2
Number of iterations	30

set of small components, namely intrinsic mode functions (IMFs). The IMFs are the complete and orthogonal basis of initial signal [27, 28]. Once they are obtained, the analytic signal $z_i(t)$ of the i th IMF component is acquired [27]:

$$z_i(t) = c_i(t) + jH[c_i(t)], \tag{4}$$

where $c_i(t)$ is the i th component of IMF and $H[c_i(t)]$ is the Hilbert transform:

$$H[c_i(t)] = \frac{1}{\pi i} \int_{-\infty}^{+\infty} \frac{c_i(\tau)}{t - \tau} d\tau. \tag{5}$$

After that, the envelope of signal $X(t)$ can be obtained by computing the absolute value of $z_i(t)$ [27]:

$$a_i(t) = \sqrt{c_i^2(t) + H^2(c_i(t))}. \tag{6}$$

Besides the envelope calculation, the EMD is applied for denoising. Generally, the measurement noise mainly comes from the high frequency, and the fault characteristic spectrum in the low-frequency band is sufficient. Therefore, the first component IMF1 is ignored. The other components are summed up to achieve the noise reduction. Figure 5 shows the IMFs of the vibration signal measured from a bearing with an inner race fault.

Once the high-frequency noise has been filtered and the signal envelop has been captured, the Fourier transform is applied to calculate the spectrum. Figure 6 shows the Fourier transform results, with (a) for the original

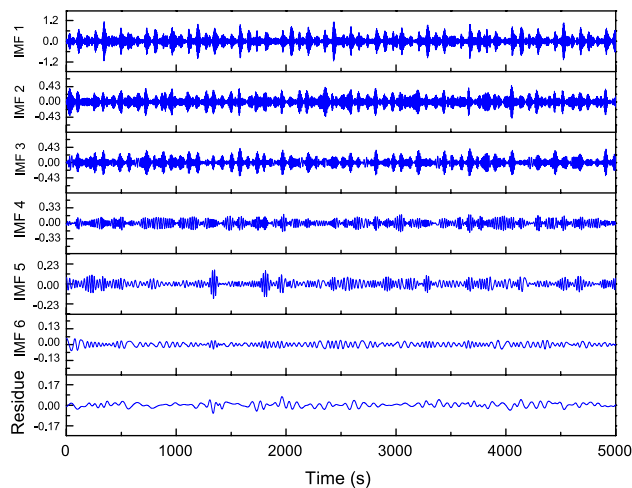


Fig. 5 IMFs of vibration signal of a bearing with inner race fault

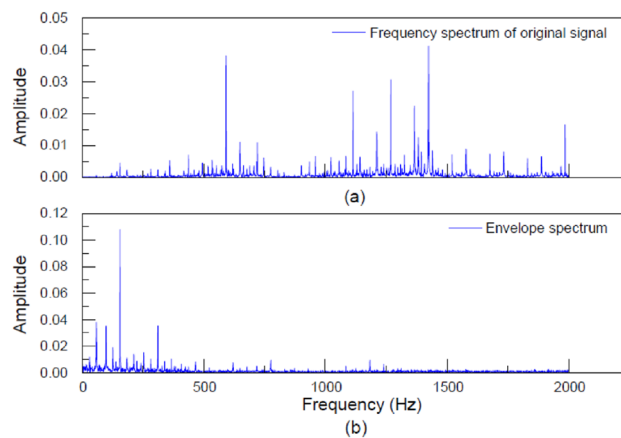


Fig. 6 **a** Frequency spectrum of original signal, **b** frequency spectrum of envelope signal

vibration signal without any signal processing and (b) for the envelop signal obtained by the EMD and Hilbert transform. It can be seen that the signal after the EMD and

Hilbert transform contains less high-frequency components. This indicates that the high-frequency components have been significantly reduced by the EMD, while the characteristic frequencies of fault bearings are kept. The spectrum in Fig. 6b is the so-called envelop spectrum and will be used as a kind of input.

Feature Vector

Though the envelope spectrum contains much information for fault diagnosis, the input size has to be big enough to guarantee the frequency resolution, which inevitably leads to high computing load. Thus, the feature vector consisting of manually extracted features from time and frequency domains is proposed.

Time-Domain Features

When a bearing starts to degrade, the time-domain acceleration response will gradually present the non-stationary and non-Gaussian dynamics [29]. Therefore, 13 indexes in total are selected from time domain. The kurtosis and skewness characterize the non-Gaussian dynamics. The impulse factor, crest factor, standard deviation and shape factor identify the non-stationary characteristics. A summary of all these features and their corresponding formulas is given in Table 4 [30].

Frequency-Domain Features

In frequency domain, when a defect occurs on bearing, peaks will appear at the fault characteristic frequencies (FCFs) of the corresponding component. For example, a bearing with the outer race fault produces the peaks at the ball pass frequency of outer race (BPFO) and its harmonics at frequency spectrum. The same is for the inner race and ball faults, corresponding to the ball pass frequency of inner race (BPFI) and the ball spin frequency (BSF), respectively. These FCFs contain much information of bearing condition and therefore are taken to extract the frequency-domain features. The theoretical FCFs can be calculated as following [31]:

$$\text{BSF} = \frac{Df}{d} \left[1 - \left(\frac{d}{D} \cos \theta \right)^2 \right], \quad (7)$$

$$\text{BPFI} = \frac{nf}{2} \left(1 + \frac{d}{D} \cos \theta \right), \quad (8)$$

$$\text{BPFO} = \frac{nf}{2} \left(1 - \frac{d}{D} \cos \theta \right), \quad (9)$$

where f is the shaft rotation frequency, n , d , D and θ are four geometric parameters of bearing, and their specifications are detailed in Table 5.

Table 4 Time-domain features

Feature	Formulation
Mean	$\hat{\mu} = \frac{1}{N} \sum_{i=1}^N X(i)$
Standard deviation	$\left(\frac{1}{N-1} \sum_{i=1}^N (X(i) - \hat{\mu})^2 \right)^{\frac{1}{2}}$
RMS	$\left(\frac{1}{N} \sum_{i=1}^N [X(i)]^2 \right)^{\frac{1}{2}}$
Skewness	$\frac{1}{N} \sum_{i=1}^N \left(\frac{X(i) - \hat{\mu}}{\hat{\sigma}} \right)^3$
Kurtosis	$\frac{1}{N} \sum_{i=1}^N \left(\frac{X(i) - \hat{\mu}}{\hat{\sigma}} \right)^4$
Crest factor	$\frac{\max X }{\text{RMS}}$
Impulse factor	$\frac{\max X }{\frac{1}{N} \sum_{i=1}^N X(i) }$
Shape factor	$\frac{\text{RMS}}{\frac{1}{N} \sum_{i=1}^N X(i) }$
Median	amplitude $\left(\frac{N+1}{2} \right)$
Range	$\max(X) - \min(X)$
Variance	$\frac{1}{N-1} \sum_{i=1}^N (X(i) - \hat{\mu})^2$
Maximum	$\max(X)$
Minimum	$\min(X)$

Table 5 Parameter specifications for ball bearing

Number of rolling elements (n)	Ball diameter (d)	Pitch diameter (D)	Initial contact angle (θ)
9	0.3126 in.	1.537 in.	0°

Figure 7 gives an example of BSF, BPFO and BPFI from the first to sixth order. The characteristic frequencies can be affected by many factors such as the shaft speed, external load, friction coefficient, raceway groove curvature and defect size [32–34]. Therefore, it is often the case that there exists bias between the theoretical and actual FCFs (BPFO, BPFI, BSF). In addition, sidebands will inevitably appear in the acceleration spectrum [35]. For example, for the ball fault, peaks will occur at $k \times f_{\text{BSF}}$, $k \times f_{\text{FTF}}$, $k \times f_{\text{BSF}} \pm j \times f_{\text{FTF}}$, where $j, k = 1, 2, \dots, N$, f_{BSF} stands for the FCF of ball fault, and f_{FTF} means the fundamental train frequency [36]. This also explains the peaks before the first order of BSF in Fig. 7a. Even worse, some harmonics of the FCFs influenced by modulation of other vibrations may not be detected in the test bench [33]. Thus, to ensure the FCFs included into the frequency-domain features, 8 amplitude values around the theoretical FCFs, as shown in Fig. 8, are selected for each order, and the FCFs from the first to sixth order are

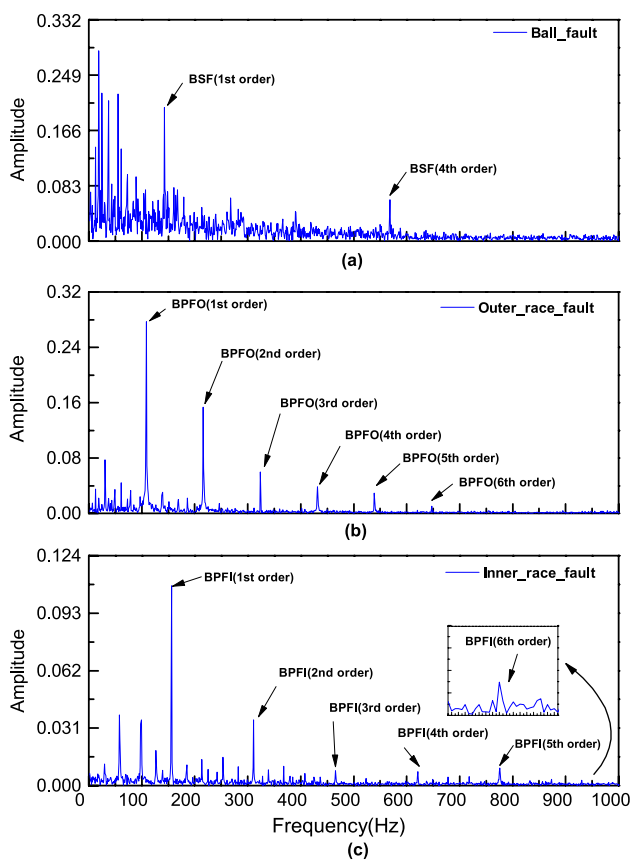


Fig. 7 Illustration of a BPF, b BPFO and c BPFI

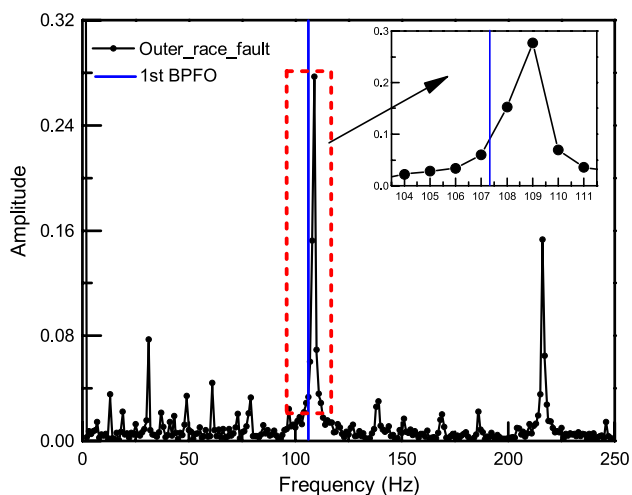


Fig. 8 Process to obtain frequency-domain features in feature vector

considered. This brings 48 amplitude values (8 values for each order from 1st to 6th) for each FCF. In this paper, 3

FCFs are considered (BPFI, BPFO, BSF), which brings 144 frequency-domain features in total.

Feature Vector Construction

After the manually extracted features from time and frequency domains are determined, the next step is to build a feature vector. It consists of 157 elements as $x = [x_1, x_2, \dots, x_N]$. The definition of each element is listed in Table 6.

Modification of Fully Connected Layer

The fully connected layer in CNN is actually a mere classifier which outputs a probability for each label. Regarding classification, there are many other preferable machine learning algorithms like SVM and decision tree. Thus, it may be reasonable to replace the fully connected layer with other more promising methods. In this work, the fully connected layer is replaced by SVM, decision tree and random forest to investigate the impact of different classifiers on the accuracy of the bearing fault classification.

Definition of Four Indicators

In most published researches, only the algorithm accuracy have been considered, while the performance in other aspects have been ignored [37–39]. Apart from accuracy, a good algorithm should also perform stably and robustly, which means stable results under different measurements and keeping stable enough when noise is added. Computing efficiency should also be addressed. Therefore, accuracy, stability, robustness and

Table 6 Feature vector description

Number	Feature	Number	Feature
1	Mean	38–45	2nd BPFI
2	Standard deviation	46–53	2nd BPFO
3	Skewness	54–61	2nd BSF
4	Kurtosis	62–69	3rd BPFI
5	Max	70–77	3rd BPFO
6	Min	78–85	3rd BSF
7	Range	86–93	4th BPFI
8	Median	94–101	4th BPFO
9	Variance	102–109	4th BSF
10	RMS	110–117	5th BPFOI
11	Impulse factor	118–125	5th BPFO
12	Crest factor	126–133	5th BSF
13	Shape factor	134–141	6th BPFI
14–21	1st BPFI	142–149	6th BPFO
22–29	1st BPFO	150–157	6th BSF
30–37	1st BSF		

efficiency as four indicators are first proposed to evaluate the CNN performance. Their definitions are introduced as follows.

- **Accuracy** The accuracy under each time is defined as the percentage of samples that are correctly classified. CNN runs N times, the final accuracy is defined as the average accuracy under N times of simulations as follows:

$$\text{Accuracy} = \frac{\sum_{i=1}^N Y_i}{N}, \quad (10)$$

$$Y_i = \frac{y_i}{L}, \quad (11)$$

where y_i is the number of correctly classified samples, L is the number of the whole samples, and N is the times of simulation. The values of L under each label are summarized in Table 1. Without loss of generality, N is chosen as 20 in this study.

- **Stability** The stability is defined as the standard deviation of accuracy under the N times of simulation, with the formulation as follows:

$$\text{Stability} = \sqrt{\frac{\sum_{i=1}^N (Y_i - \text{Accuracy})^2}{N - 1}}. \quad (12)$$

- **Robustness** The robustness is proposed to evaluate the CNN's performance under noise. It is characterized by the accuracy when 20 dB signal-to-noise ratio (SNR) noise is added into the input and calculated by the following equation:

$$\text{Robustness} = \frac{\sum_{i=1}^N Y'_i}{N}, \quad (13)$$

where Y'_i is the accuracy of the i -th simulation under 20 dB SNR noise.

- **Efficiency** The consumed time T_i for a simulation is defined as the interval from the beginning of training to the end of validation. The efficiency is defined as the average consumed time of N simulations:

$$\text{Efficiency} = \frac{\sum_{i=1}^N T_i}{N}, \quad (14)$$

In our research, the algorithm is implemented at the Amazon Web Service platform with p2.xlarge instance.

Case Study and Result Analysis

In this section, two cases are designed to validate the modified CNN. In case 1, the training and test datasets come individually from the same distribution for all the four

working loads. The optimal modifications are determined in terms of hyperparameters, input and fully connected layer. In case 2, the optimized CNN is further verified under the working condition adaptation scenario, where the CNN is trained with the dataset from one working load and then tested with the datasets from the other three different working loads. Afterward, three methods are proposed to enhance the CNN's adaptation performance across different working conditions.

Case 1: Training and Test Datasets from the Same Working Load

Results of Three Modifications for CNN

Hyperparameters Optimization

As discussed above, the hyperparameters of CNN are determined by PSO. One thing that must be mentioned is that the optimization in this study is carried out in the order of input, hyperparameters, and fully connected layer. When optimizing the hyperparameters, the input type is fixed as the envelope spectrum. Figure 9 shows the optimization process of PSO over 30 iterations. It can be found that the PSO almost converges to the maximum accuracy after only five iterations, and the CNN's accuracy on fault classification increases from 98.9 to 99.81%. Table 7 shows the optimized hyperparameters that are used in CNN for the further modification exploration.

Comparison of the Three Kinds of Input Data

To study the impact of manual feature extraction on the performance of bearing fault classification, the CNN is trained with three kinds of input data: (1) original signal, (2) envelope spectrum and (3) feature vector extracted from time and frequency domains. CNN with each kind of input data

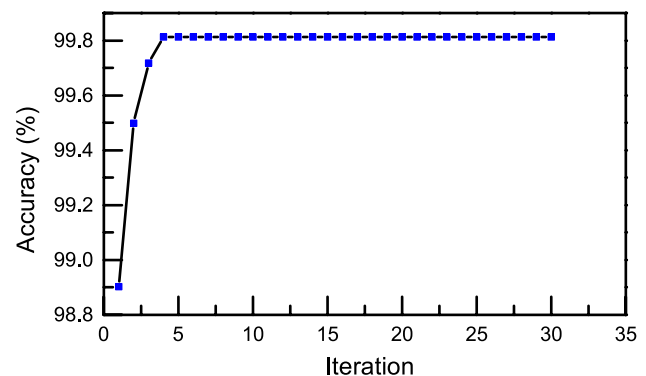


Fig. 9 Fitness value of each iteration

Table 7 hyperparameters used in CNN

Hyperparameter	Value
Drop rate	0.55
Learning rate	0.0006
Kernel size of the first layer	5×5
Kernel size of the second layer	3×3
Number of filters in the first layer	128
Number of filters in the second layer	256
Batch size	110
Density layer (flatten)	512

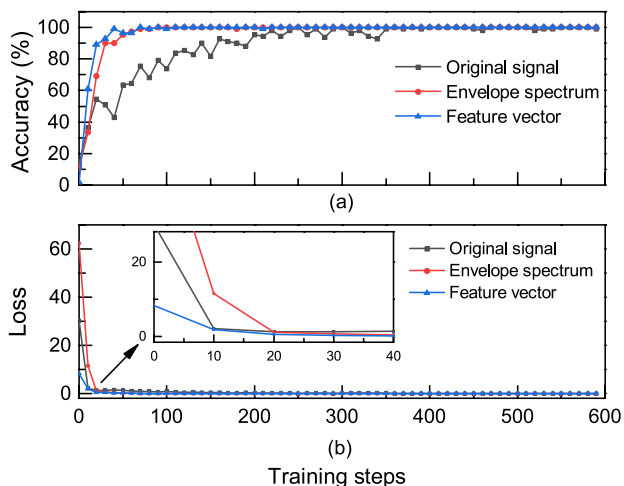


Fig. 10 Training accuracy and loss comparison of three different input data types

is trained for 20 times. Figure 10 shows the training accuracy and loss. As seen, the training accuracy of both the envelope spectrum and the feature vector need 90 steps to reach 100% for the first time and is then consistently around 100% after 100 steps. On the contrary, for CNN with the original signal as input, 250 steps and 360 steps are needed, respectively. This confirms that the CNN with the original signal as input has much lower training speed and needs much more steps to achieve the same results. When CNN is fed with the envelope spectrum and feature vector, despite the difference in training speed, the final training accuracy and loss are almost the same.

Figure 11 shows the accuracy boxplot of three different input data types for complete comparison. It is obvious that the envelope spectrum signal has a more concentrated 25%-75% range and higher mean of accuracy. The CNN with the feature vector as input also has quite good results, but is slightly lower than that with the envelope spectrum. Moreover, both of the proposed input data types have better performance than the input of original signal on all the four evaluation metrics.

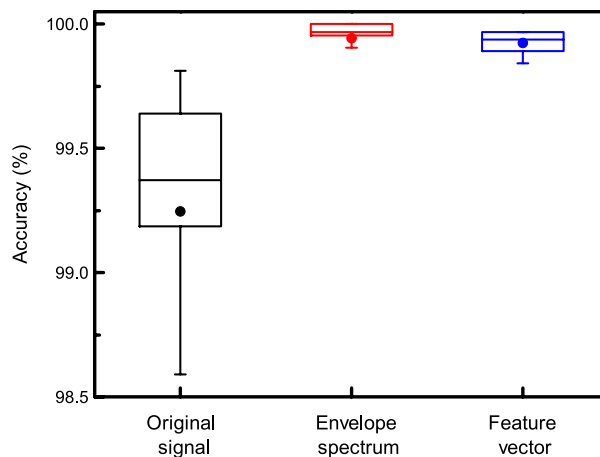


Fig. 11 Boxplot of three different input data types

An overall comparison among three input types over four evaluation metrics are summarized in Table 8. It can be found that CNN with the envelope spectrum as input performs the best in accuracy (99.94%) and robustness (99.69%), while CNN with the feature vector as input shares the best stability (0.04) and training speed (20 (s)). By contrast, CNN using the original signal as input shows the worst results in all the four indicators. Briefly, according to the above analysis, CNN with the envelope spectrum achieves the best test accuracy and robustness, the most concentrated 25–75% range and the highest median line. Therefore, the envelope spectrum is identified as the best input type and applied in the further experiments.

Modification for the Fully Connected Layer

To find the best modification candidate for the fully connected layer among SVM, decision tree and random forest, the performance of CNN is compared and discussed in this subsection. Figure 12 indicates that the replacement of fully connected layer by SVM and random forest improves the performance of CNN and reaches 100% test accuracy 18 and 10 times during 20 tests, respectively. The lowest accuracy of CNN + SVM and CNN + Random forest reaches up to 99.96%, but is still higher than the average accuracy of the general CNN. Conversely, the decision tree has no positive effect on the accuracy of CNN.

Figure 13 indicates the outstanding performance of CNN + SVM due to its small 25–75% range and high median line. A summary of comparison results for the four indicators is presented in Table 9, which also verifies the superiority of CNN + SVM over other algorithms in three aspects (accuracy: 99.9969%, robustness: 99.02%, stability: 0.0097) except training speed. Therefore, it can be concluded that replacing the fully connected layer by SVM and random forest can improve the CNN’s classification

Table 8 Result summary of three input data types

Evaluation metric	Original signal	Envelope spectrum	Feature vector
Test set average accuracy	99.25%	99.94%	99.92%
Stability	0.65	0.11	0.04
Robustness	98.90%	99.69%	99.59%
Training speed (s)	143	141	20

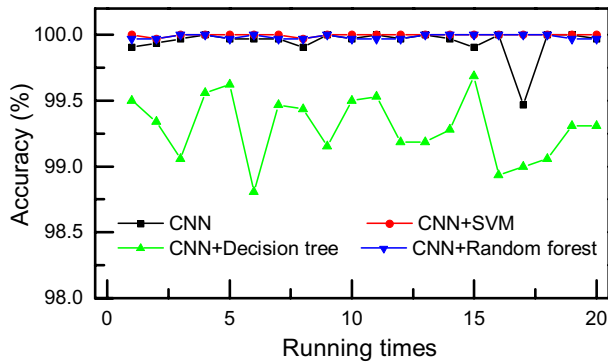


Fig. 12 Accuracy of modification for fully connected layer with envelope spectrum signal

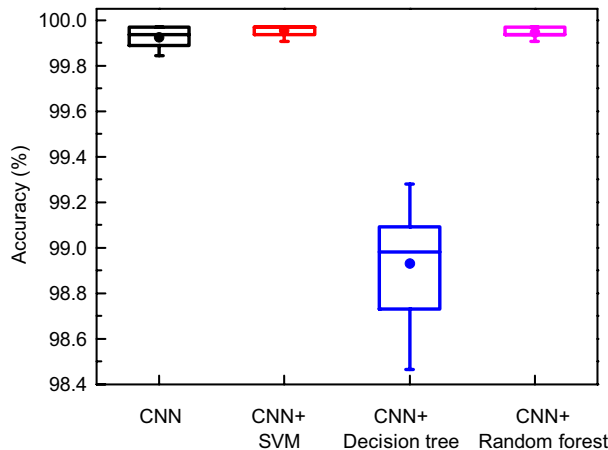


Fig. 13 Boxplot of four different classifiers

Table 9 Result summary of fully connected layer modification

Evaluation metric	CNN	CNN + SVM	CNN + decision tree	CNN + random forest
Test set average accuracy	99.9436%	99.9969%	99.2965%	99.9843%
Stability	0.11	0.0097	0.24	0.02
Robustness	99.69%	99.92%	99.79%	99.73%
Training speed (s)	143	176	162	164

performance. SVM is slightly better than random forest in all aspects except training speed, which makes SVM the best replacement of the softmax classifier.

Overall Validation of the Proposed Model

Based on the above results, a conclusion can be drawn that the performance of CNN on the bearing default classification can be enhanced after three modifications proposed in this work. Therefore, PSO, envelope spectrum signal and SVM are integrated into the general CNN. This optimized CNN is expressed as CNN + PSO + Spectrum + SVM, with the terms behind CNN standing for the combination of improvements. To further verify its performance, Gaussian noise is added to the signal manually. The SNRs of the new data are 5 dB, 10 dB, 15 dB, and 20 dB. As shown in Fig. 14, the average accuracy under 20 tests remains nearly the same. The distribution is centralized when SNR changes from 20 to 10 dB, and it is still higher than 99% even when the SNR is 5 dB. This indicates the significant robustness, stability and accuracy of our method proposed.

Analysis of the Three Modifications for CNN

To further investigate the hidden reasons behind the facts, the quantitative comparisons and causal analysis are conducted. Figure 15 demonstrates the average classification accuracy of CNN in 20 tests. Over the whole data under four different working conditions (0 HP, 1 HP, 2 HP, 3 HP), the general CNN has achieved a high accuracy of 98.8027%

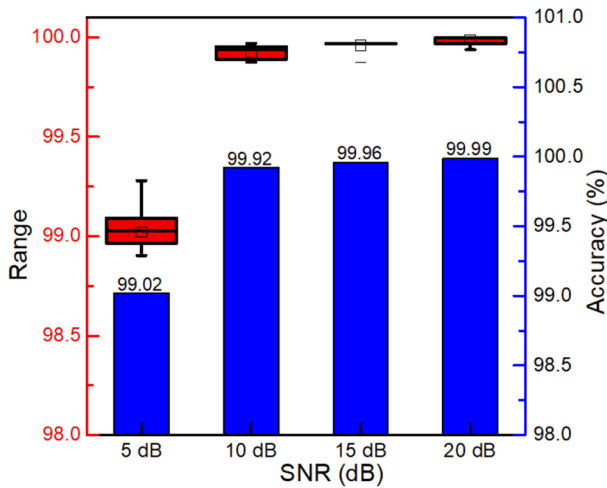


Fig. 14 Results of robustness validation test

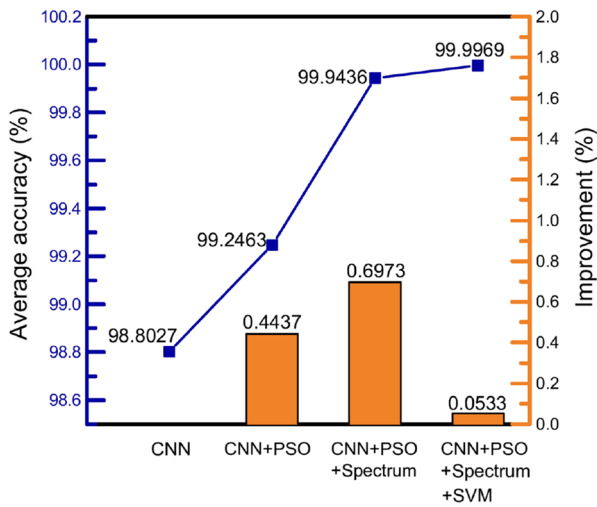
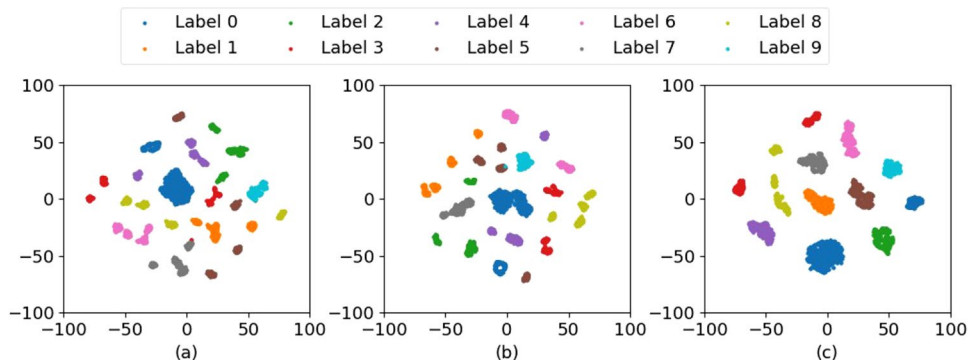


Fig. 15 Contributions to CNN's performance improvement from different modifications

Fig. 16 Visualization of fully connected layer: **a** general CNN, **b** CNN optimized by PSO (CNN + PSO) and **c** CNN + PSO with envelope spectrum input (CNN + PSO + spectrum)



in fault classification. The accuracy increases from 98.8027 to 99.2463% after the hyperparameter optimization, then further from 99.2463 to 99.9436% after the use of envelope spectrum as input data. The further use of SVM results in only 0.0533% improvement, indicating that the most potential way to optimize CNN is first the use of appropriate manually extracted input features and subsequently the implementation of hyperparameter optimization.

These results can be well explained by visualization of the fully connected layer output as shown in Fig. 16, in which each colour represents one bearing fault label. From Fig. 16a, b, there is no significant improvement in the boundaries between labels. However, the boundaries between every two labels in Fig. 16c appear much clearer, with better compactness in one label and separateness between labels.

For simplicity, the optimized CNN expressed as CNN+PSO+Spectrum+SVM above is hereafter referred to as CNN*.

Case 2: Training Dataset from One Working Load and Test Dataset from Other Working Loads

Problem Formulation

According to the above results, the CNN* performs quite well in the bearing fault diagnosis. However, the training and test datasets are individually collected from one load condition of four working loads. This means that the training and test datasets have the same data distribution. In practice, nevertheless, it is often the case that the test dataset is collected from a working condition that is different from that for the training dataset. This leads to a crucial problem since the training and testing datasets present different distributions. To simulate this practical scenario, the optimized CNN* in case 1 is trained with dataset from one working load and then tested with datasets from the other three working loads that have never been used in the training phase. As shown in Table 10, there are totally four groups, each group takes data from only one working load as training set and data from the other three working loads as test sets.

Table 10 Data description for working load adaption

Sub-case	Training set	Test sets
Group 1	Load 0	Load 1, Load 2, Load 3
Group 2	Load 1	Load 0, Load 2, Load 3
Group 3	Load 2	Load 0, Load 1, Load 3
Group 4	Load 3	Load 0, Load 1, Load 2

As shown in Table 11, when the CNN* is trained with data from one working condition and then transferred directly to the other working conditions, the CNN*'s performance reduces sharply. Take group 1 as an example, the CNN* achieves an accuracy of 100% under load 0. However, when it is applied to load 1, load 2 and load 3, the accuracy reduces to 80.13%, 86.67% and 82.41% respectively. In group 2, when a well-trained CNN* is transferred directly from load 1 to load 0, the accuracy decreases even to 58.63%. The test results indicate that the previously optimized CNN* remain the highest accuracy in the training phase but is not able to detect the fault correctly in the test phase. Considering that the extracted and learned features during the training phase tend to overfit the data in one working condition, the CNN*'s performance reduction in the working condition adaptation can be explained. Therefore, the trained CNN* fails to identify more common features shared by the data from other various working conditions.

The above results naturally raise a question: how do we maintain the performance of CNN* when transferring it from one working condition to other working conditions? In this paper, this phenomenon is termed as working condition adaptation, which has been regarded as an urgent issue to be solved before deploying CNN in practical industrial applications. In the following, three effective solutions are introduced to deal with this problem.

Methodology for Condition Adaptation

Multiple Convolutional Layers (MCL)

In general, the CNN's ability to extract complex features improves as the number of convolutional layers increases.

Table 11 Results of CNN* directly transferred between different working conditions

Training set	Load 0			Load 1		
Training set accuracy	100.00%			100.00%		
Test set	Load 1	Load 2	Load 3	Load 0	Load 2	Load 3
Test set accuracy	80.13%	86.67%	82.41%	58.63%	83.06%	77.16%
Training set	Load 2			Load 3		
Training set accuracy	100.00%			100.00%		
Test set	Load 0	Load 1	Load 3	Load 0	Load 1	Load 2
Test set accuracy	86.33%	96.20%	94.48%	74.27%	75.87%	83.35%

Table 12 CNN modified with MCL

Layer type	Kernel size	Stride	Filters number	Padding
Convolutional layer 1	1 × 7	1 × 1	16	Same
Pooling layer 1	1 × 2	1 × 2	16	Same
Convolutional layer 2	1 × 7	1 × 1	32	Same
Pooling layer 2	1 × 2	1 × 2	32	Same
Convolutional layer 3	1 × 5	1 × 1	64	Same
Pooling layer 3	1 × 2	1 × 2	64	Same
Convolutional layer 4	1 × 5	1 × 1	64	Same
Pooling layer 4	1 × 2	1 × 2	64	Same
Convolutional layer 5	1 × 3	1 × 1	64	Same
Pooling layer 5	1 × 2	1 × 2	64	Same
Convolutional layer 6	1 × 3	1 × 1	64	Same
Pooling layer 6	1 × 2	1 × 2	64	Same
Fully connected layer	512	–	1	–
Softmax layer	10	–	1	–

Therefore, the CNN optimized in case 1 is initially modified by increasing the number of convolutional layers from 2 to 6. The improved CNN's structure is presented in Table 12. The optimized CNN* is applied in present case 2 study, the same PSO, input data type and softmax classifier as in case 1 are used for consistent performance comparison.

Data Augmentation (DA)

Data augmentation consists in generating more data based on the given limited one, which has been often used for image classification to overcome the overfitting problem [40]. The most common way is shifting, rotation, rescaling or noise addition to the existing data. In this way, not only the number, but also the variety of data is increased. As a result, the increasing diversity of training set makes it possible for CNN to have a stronger ability to yield important and general features from the provided data. Consequently, even though the training and test datasets are collected from different working conditions, the CNN will still be able to figure out the common features shared by both and make the correct predictions. The implementation of data augmentation is detailly explained below. Four data augmentation

operators will be defined. After data augmentation, the number of samples in the new training set will be 20 times larger than that in previous one.

- **Shifting** The measurement signals are shifted upward or downward, to the left or to the right with some distance. Supposed $p(x, y)$ is a point in the two-dimensional space, m and n stand for the steps, with which p is shifted along the x and y axes, respectively, $p'(x', y')$ is then the new point after shifting:

$$p'(x', y') = p(x + m, y + n), \tag{15}$$

with $m \in \{-250, 0, 250\}, n \in \{-0.2, 0, 0.2\}$ in this paper. Figure 17 demonstrates the results of shifting.

- **Rotation** The measurement signals are rotated clockwise or counterclockwise around the origin by a certain degree. Given a point $p(x, y)$ in the 2-dimensional space and θ as the rotation angle, the coordinates of the new point $p'(x', y')$ after rotation can be calculated as:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}, \tag{16}$$

with $\theta \in \{-0.005, -0.01, -0.015, 0.005, 0.01, 0.015\}$. Figure 18 demonstrates the results of rotation.

- **Noise addition** The measurement signals from different working conditions contain different noises. Therefore, adding noise to the training data obtained from one working condition is a reasonable way to increase its information and generality. In this study, noise is characterized by Signal to Noise Ratio (SNR), the value set is defined with 3 elements as $\text{SNR} \in \{20\text{dB}, 15\text{dB}, 10\text{dB}\}$. Figure 19

demonstrates the new signals with different SNR (left) and the corresponding noise (right) which is added to the original signal.

- **Rescaling** Rescaling is employed to raise the richness of measurement data from the perspective of scale. The original signal is rescaled by a factor k . It is zoomed out when $0 < k < 1$, zoomed in when $k > 1$, and unchanged when $k = 1$. Take $p(x, y)$ as a sequence in the 2-dimensional space, then $p'(x', y')$ identifies the new point after being rescaled. If $0 < k < 1$, then

$$x'_i = x_{\frac{i}{k}}, \tag{17}$$

$$y'_i = ky_{\frac{i}{k}}. \tag{18}$$

If $k > 1$, then

$$x'_{ki} = x'_{ki+1} = \dots = x'_{ki+k-1} = x_i, \tag{19}$$

$$y'_{ki} = y'_{ki+1} = \dots = y'_{ki+k-1} = ky_i. \tag{20}$$

In this research, $k \in \{\frac{1}{3}, \frac{1}{2}, 1, 2, 3\}$. Figure 20 shows the results of rescaled signals.

To sum up, shifting, rotation, noise addition and rescaling are four basic operators defined in this study to raise the information richness of signals from different aspects. For example, shifting changes the signal position and amplitude, rotation and noise addition can modify the signal shape, while rescaling regenerates signals with different

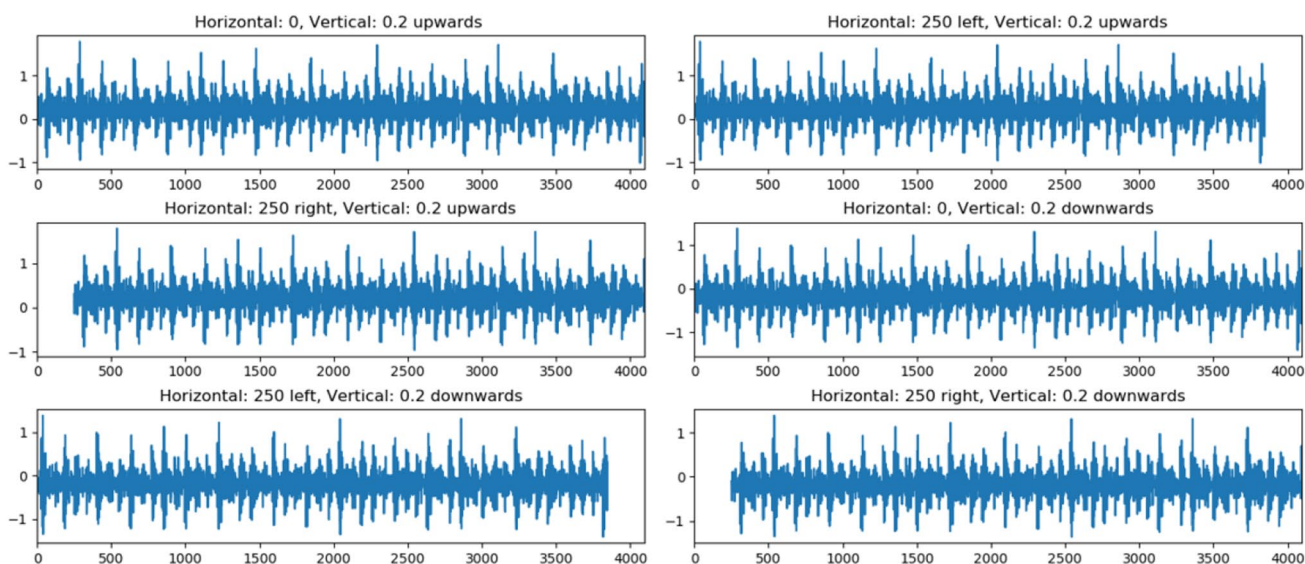


Fig. 17 Signal after shifting for inner race fault sample

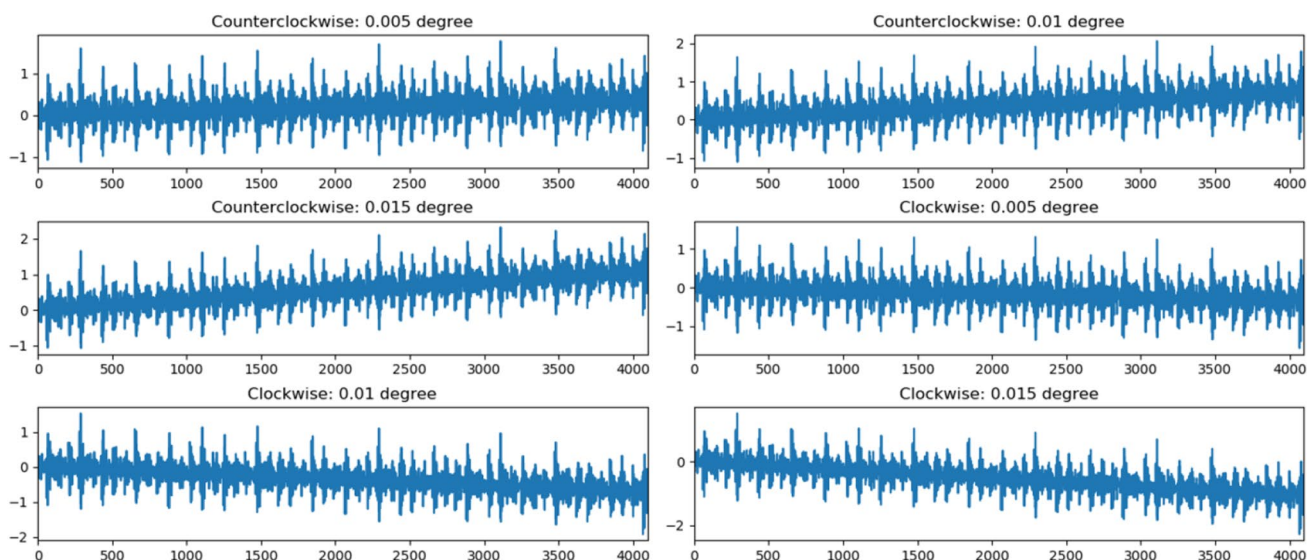


Fig. 18 Signal after rotation for inner race fault sample

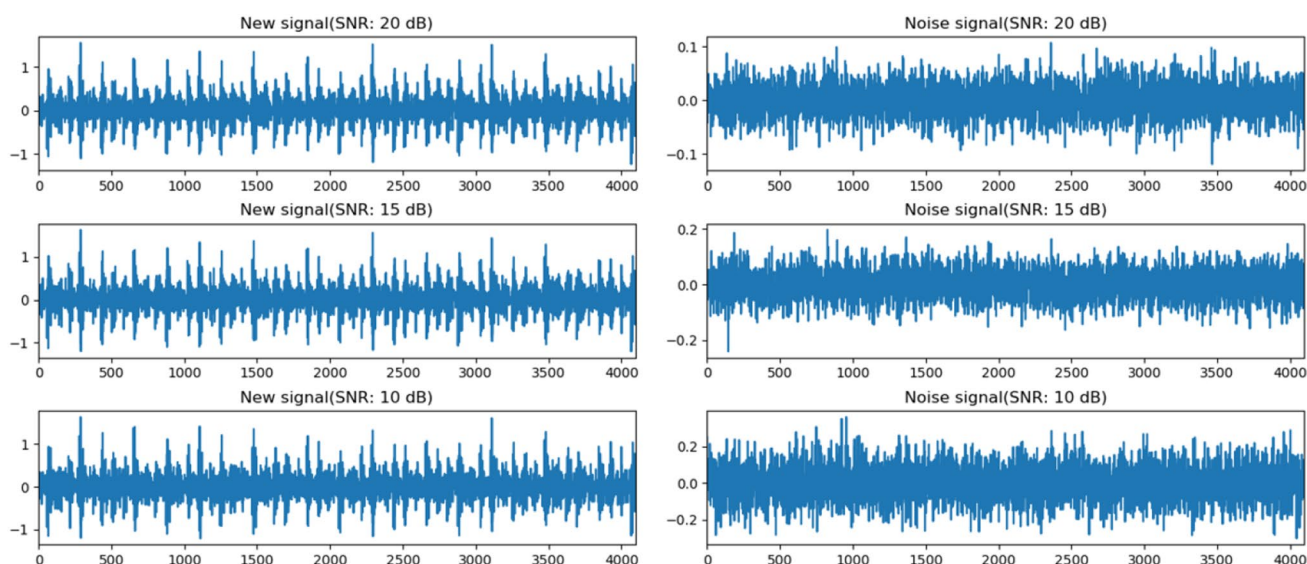


Fig. 19 Noise signal of different SNR for inner race fault sample

scales. Figure 21 presents the process of sample generation with data augmentation, which mainly includes three steps: selecting an operator Q_i , defining the parameters for Q_i , and generating new samples with Q_i .

Signal Concatenation (SC)

To further increase the variety of training dataset, signal concatenation is proposed. The main idea is to divide the

original signal into small parts, then all the four data augmentation operators introduced above are applied to each small part instead of the whole signal. Finally, the augmented parts are concatenated to form new signal that has the same length as the original one. Since each new signal contains multiple subparts that are modified with different data augmentation operators, its complexity and diversity are certainly increased. To well illustrate this method, a

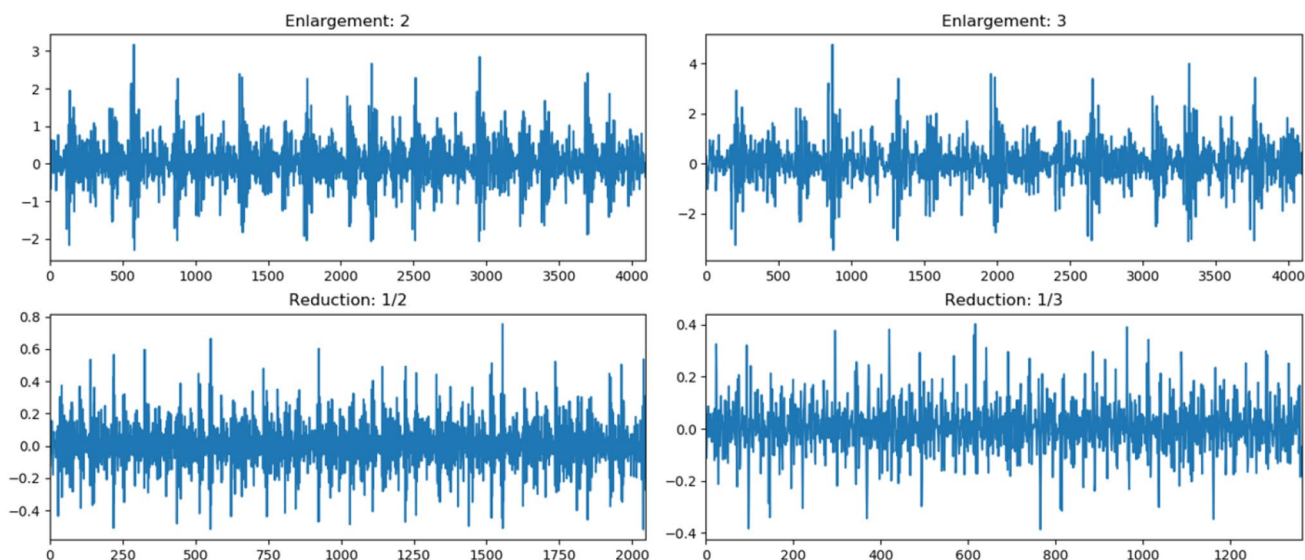


Fig. 20 Signals after rescaling for inner race fault sample

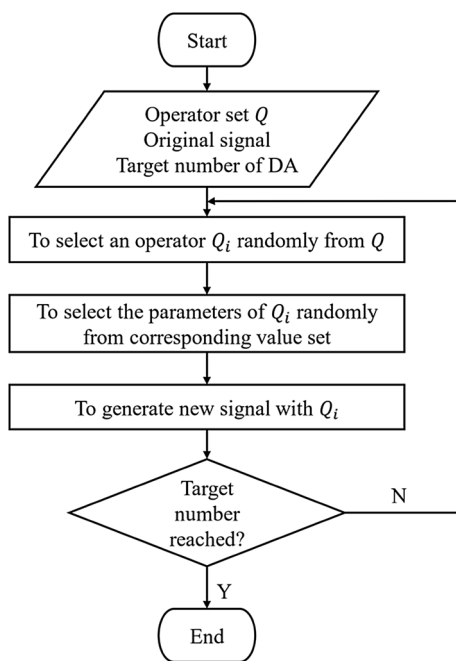


Fig. 21 The flowchart of data augmentation

Table 13 Operator and value set

Operator set Q	Value set
Shifting	$m \in \{-250, 0, 250\}, n \in \{-0.2, 0, 0.2\}$
Rotation	$\theta \in \{-0.005, -0.01, -0.015, 0.005, 0.01, 0.015\}$
Noise addition	$SNR \in \{20\text{dB}, 15\text{dB}, 10\text{dB}\}$
Rescaling	$k \in \{\frac{1}{3}, \frac{1}{2}, 1, 2, 3\}$

As given in Table 13, each operator has the same value set as defined in previous section. Algorithm 1 gives the algorithm implementation process. Firstly, the size of new training set needs to be defined. For example, the number of samples in the new training set is j times larger than that of the original one. Secondly, the original sample is divided into l subparts. For each subpart, an operator is randomly chosen from Q and the corresponding parameters of the chosen operator are also randomly determined. Thirdly, the chosen operators with the corresponding parameters are implemented in the l subparts. Finally, the augmented subparts are concatenated to form a new signal that has the same length as the original one.

set Q is defined as a collection of four data augmentation operators:

$$Q \in \{shifting, rotation, noise\ addition, rescaling\}. \quad (21)$$

Algorithm 1 Algorithm for signal concatenation**Require:**

Number of signals to be generated, j ;
 Number of subparts to divide each signal, l ;
 Operator set Q and corresponding value set;
 Original signal \vec{x}_0 ;

Ensure:

Generated signals: $\vec{x}_1, \vec{x}_2, \dots, \vec{x}_j$;
 $i \leftarrow 1$
while $i \leq j$ **do**
 Divide the original signal \vec{x}_0 into l subparts;
for Each subpart k ($1 \leq k \leq l$) **do**
 Select a random operator from Q ;
 Select a random parameter for the selected operator from the corresponding value set;
 Generate the new subparts l_k by the operator and value randomly selected above;
end for
 Concatenate all the l subparts to form a new signal \vec{x}_i ;
 $i \leftarrow i + 1$
end while

In this work, j is defined as 20 and l as 4. An example is given below to illustrate the concatenation process and results. The original signal is shown in Fig. 22, and the 5 new signals generated by means of data augmentation and signal concatenation are presented in Fig. 23. The involved operators and corresponding parameters are summarized in Table 14.

Results of Working Condition Adaption

In this section, three methods proposed above are applied individually as well as collaboratively to improve the CNN*'s performance in working condition adaptation. There are totally four modified CNNs: CNN* + MCL, CNN* + DA, CNN* + MCL + DA and CNN* + MCL +

DA + SC. The terms behind CNN* stand for the improvement approach. All of the four improved CNN*s are trained with the dataset from one load and then tested with the datasets from the other three different loads. The fault classification accuracies under the CNN*s are compared with each other, especially with the CNN*. As detailed in Table 16, when the CNN* + MCL is trained with the dataset from load 1 and transferred to load 0, load 2 and load 3 directly, the accuracy increases from 58.63 to 95.11%, from 83.06 to 100% and from 77.16 to 98.41%, respectively. However, the improvement percentages under load 3 and load 1 as the training datasets are not so significant. When CNN* + MCL is transferred from load 2 to load 0, the accuracy even drops from 86.33 to 78.39%. When DA is additionally

Table 14 Operator series of five generated samples based on signal concatenation

New sample	First part	Second part	Third part	Fourth part
1	Original signal	Noise SNR 20 dB	Rotation 0.015	0.2 downward
2	Rotation 0.005	Rescale $\times 3$	Noise SNR 15 dB	0.2 up 250 right
3	Rotation -0.01	0.2 down 250 left	Rescale $\times 1/2$	Noise SNR 10 dB
4	Rescale $\times 2$	Rotation 0.01	0.2 upwards	Rotation -0.005
5	0.2 up 250 left	Rotation -0.015	0.2 down 250 right	Rescale $\times 1/3$

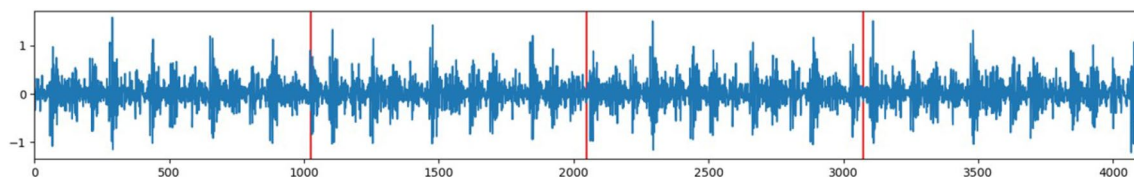


Fig. 22 Original signal divided into four parts for inner race fault sample

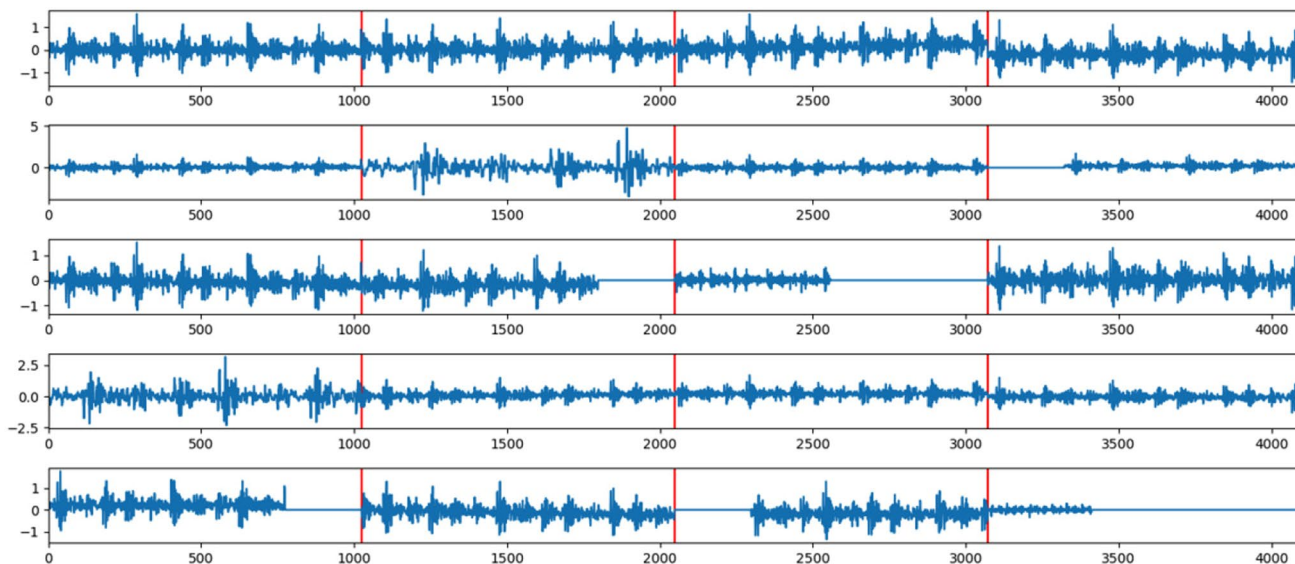


Fig. 23 Five generated samples based on signal concatenation for inner race fault sample

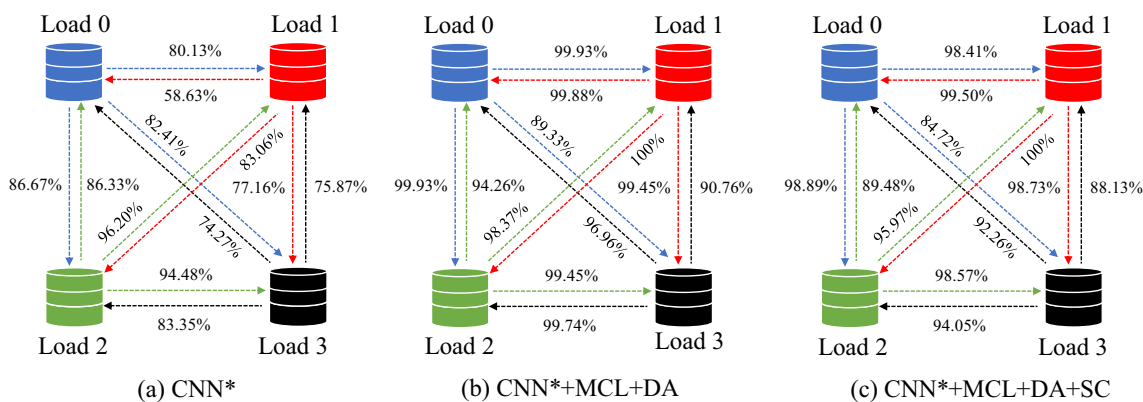


Fig. 24 Comparison of condition adaptation results

Table 15 Average classification accuracy comparison of different methods

Method	Load 0 (%)	Load 1 (%)	Load 2 (%)	Load 3 (%)	Overall (%)
CNN*	83.07	72.95	92.33	77.83	81.55
CNN*+MCL	89.29	97.84	91.44	83.88	90.61
CNN*+DA	89.78	97.59	96.02	96.76	95.03
CNN*+MCL+DA	96.40	99.78	97.36	95.82	97.34
CNN*+MCL+DA+SC	94.01	99.41	94.67	91.48	94.89

applied, the performance of transfer learning under all the four working loads is improved. Accuracies under load 1, load 2 and load 3 are higher than 90%, while the accuracy under load 0 is still a little bit low. CNN* + MCL + DA performs better than CNN*, CNN* + MCL and CNN* + DA under all 12 working condition adaptation tasks, which means that MCL + DA is a better solution than individual

MCL or DA to the CNN*'s working condition adaptation problem. When compared with CNN* + MCL + DA, the last variant CNN* + MCL + DA + SC performs a little bit worse but still much better than CNN* + MCL and CNN* + DA. This confirms that all three proposed methods (MCL, DA and SC) have positive effect to improve the CNN*'s performance in the working condition adaptation, while the

Table 16 Transfer results under different working conditions

Training set	Load 0			Load 1		
Test set	Load 1 (%)	Load 2 (%)	Load 3 (%)	Load 0 (%)	Load 2 (%)	Load 3 (%)
CNN*	80.13	86.67	82.41	58.63	83.06	77.16
CNN* + MCL	90.89	92.36	84.62	95.11	100.00	98.41
CNN* + DA	95.35	90.80	83.19	99.15	98.08	95.52
CNN* + MCL+DA	99.93	99.93	89.33	99.88	100.00	99.45
CNN* + MCL + DA + SC	98.41	98.89	84.72	99.50	100.00	98.73
Training set	Load 2			Load 3		
Test set	Load 0 (%)	Load 1 (%)	Load 3 (%)	Load 0 (%)	Load 1 (%)	Load 2 (%)
CNN*	86.33	96.20	94.48	74.27	75.87	83.35
CNN* + MCL	78.39	99.19	96.76	81.66	84.72	85.27
CNN* + DA	97.11	98.31	92.63	97.03	95.22	98.02
CNN* + MCL+DA	94.26	98.37	99.45	96.96	90.76	99.74
CNN* + MCL + DA+SC	89.48	95.97	98.57	92.26	88.13	94.05

Table 17 Comparison between CNN* + MCL + DA + SC and CNN + MCL + DA + SC

Training set	Load 0			Load 1		
Test set	Load 1 (%)	Load 2 (%)	Load 3 (%)	Load 0 (%)	Load 2 (%)	Load 3 (%)
CNN* + MCL + DA + SC	98.41	98.89	84.72	99.50	100.00	98.73
Average	94.01			99.41		
CNN*-SVM + MCL + DA + SC	99.97	99.90	90.17	99.50	100.00	99.09
Average	96.68			98.94		
Training set	Load 2			Load 3		
Test set	Load 0 (%)	Load 1 (%)	Load 3 (%)	Load 0 (%)	Load 1 (%)	Load 2 (%)
CNN* + MCL + DA + SC	89.48	95.97	98.57	92.26	88.13	94.05
Average	94.67			91.48		
CNN*-SVM + MCL + DA + SC	94.34	97.79	99.77	96.11	92.94	97.30
Average	97.30			95.45		

combination of different methods should be carefully tackled with. Figure 24 gives a more intuitive comparison among CNN*, CNN* + MCL + DA and CNN* + MCL + DA + SC on the condition adaptation performance within every two different working loads.

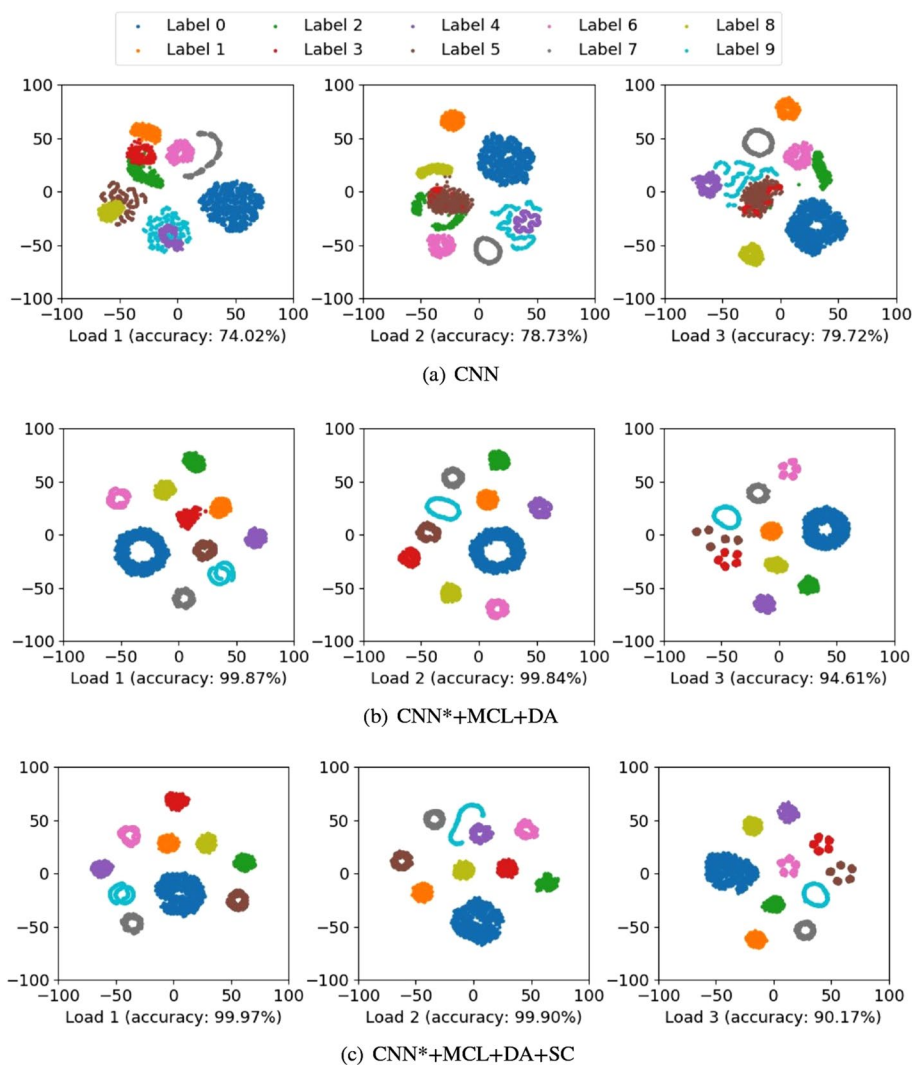
Analysis of Working Condition Adaption

Table 15 presents the overall average accuracy. It is only 81.55% when CNN* is directly transferred from one working condition to the other three different working conditions. When MCL, DA, MCL + DA and MCL + DA + SC are integrated to CNN* on training data, they lead to significant improvements. Actually, regarding overall classification accuracy under four working loads, an individual combination of MCL and DA with CNN* already yields an accuracy

improvement percentage of 9.06% and 13.48%, respectively. The combination of two or three improvement methods with CNN* performs even better. The results confirm that the combination of MCL + DA achieves the highest accuracy (97.34%), which means the best way to improve the CNN*'s working condition adaptation performance is to add more convolutional layers and augment the training data based on the data augmentation operators (shifting, rotation, noise and rescaling).

As to the reason why CNN* + MCL + DA + SC does not perform better than CNN* + MCL + DA as expected, it can be explained from two aspects (Table 16). Firstly, in the signal concatenation, a complete and continuous acceleration sample is divided into many short subparts, and the augmentation operators for each part are randomly selected. This may bring discontinuity, long zero-padding and even abrupt peaks in the new generated signals, which can be seen

Fig. 25 Visualization of fully connected layer



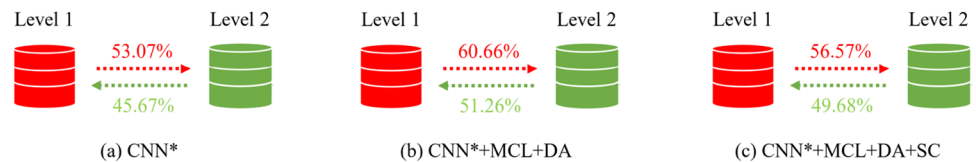
from the five generated signals in Fig. 23, especially the fifth signal. Secondly, in CNN* + MCL + DA + SC, the default fully connected layer in CNN has been replaced by SVM, which has much stronger mapping and classification ability. Consequently, some generated signals from SC, which have unnecessary and unimportant local characteristics, may also be classified as fault, resulting in the inevitable CNN*'s accuracy reduction. Take a look at Table 17, it can be found that once SC combined with MCL and DA for data augmentation, CNN* without SVM (expressed as CNN*-SVM) performs better than or at least the same as CNN* under all condition adaptation cases, which also proves the above analysis to some extent.

To further explain the obtained results, the t-Distributed Stochastic Neighbor Embedding (t-SNE) which is a non-linear technique for high-dimension reduction, is

applied to visualize the fully connected layer of CNN* (a), CNN*+MCL+DA (b) and CNN*+MCL+DA+SC (c) in Fig. 25. It is obvious that the margins between classes become bigger in Fig. 25b and c, which indicates that the proposed methods indeed contribute to the improvement of classification performance of CNN* in the working condition adaptation. Additionally, it can be found that CNN* + MCL + DA + SC performs better than CNN* + MCL + DA in fully connected layer in terms of classifiability, namely the in-class compactness and between-class separateness, but the final classification accuracy presents opposite results, which further confirms our above second explanation.

Last but not least, to validate the proposed method with data from other sources, the experimental data from Paderborn University (PU) bearing test bench was used. Generally, the PU dataset provides bearing measurement

Fig. 26 Comparison of condition adaptation results (PU dataset)



data in three categories: normal, inner race failure, and outer race failure. According to the damage size, the failure severity can be further divided into two levels, with damage size smaller than 2 mm as level 1 and damage size above 2 mm, but smaller than 4.5 mm as level 2 [41]. In addition, to keep the acceleration fed into CNN having nearly the same sampling frequency as the acceleration measurement from the CWRU dataset, a downsampling factor of 4 is applied to filter the raw acceleration that is collected with an initial sampling frequency of 64 kHz. With the PU dataset, transfer learning between two different fault levels was implemented. The structure and parameters of the CNN* model for this dataset can be found in [42], with the three proposed methods (MCL, DA, and SC) in “Methodology for condition adaptation” to improve the CNN*’s transfer learning performance. As shown in Fig. 26, we can find that under both the transfer scenarios from level 1 to level 2 and from level 2 to level 1, when the CNN* is combined with the proposed methods, its performance in condition transfer learning can be enhanced, which further confirms the effectiveness of the proposed methods. Regarding the low values of accuracy, it maybe caused by the mismatch between the operator parameters and the measurement data from PU dataset. Nevertheless, the change tendency is consistent with the result obtained from the CWRU dataset.

Conclusion

CNN has been widely used in the bearing fault diagnosis. To further enhance its performance, CNN has been improved by three aspects. Firstly, eight hyperparameters are optimized by the particle swarm optimization. Secondly, the normal original signal input is replaced by the envelope spectrum and feature vector (157 features extracted from time and frequency domain). Finally, the fully connected layer is substituted by the SVM, decision tree and random forest to improve its classification ability. The contributions to the CNN’s performance improvement from each modification are quantitatively discussed and compared. Furthermore, an overall evaluation method is introduced in aspects of classification accuracy, stability, robustness to noise and computing efficiency. Test results have shown that training the CNN model with the optimized hyperparameters and envelope spectrum as input data can improve its performance. Moreover,

the proposed method sustains high accuracy under the noise condition. However, when the training and test datasets are obtained from different working conditions, the CNN’s performance cannot be maintained. To solve the performance reduction problem, the multi-convolutional layer (MCL), data augmentation (DA) and signal concatenation (SC) have been proposed and investigated. Validation results based on experimental data have shown that DA has better performance than MCL and the combination of two or three approaches has better performance than the individual one. As for further research, the data augmentation-based signal change dynamics and physics across working conditions will be explored. Data augmentation on spectrum instead of time-domain signal will also be addressed in a future work.

Acknowledgements This research is supported by CSC doctoral scholarship (201806250024) and Zhejiang Lab’s International Talent Fund for Young Professionals (ZJ2020XT002). Acknowledgment is also made for the bearing dataset provided by Case Reserve Western University.

Data availability Data will be available on request.

References

- Vachtsevanos GJ, Vachtsevanos GJ (2006) Intelligent fault diagnosis and prognosis for engineering systems, vol 456. Wiley Online Library, New York
- Zhang S, Zhang S, Wang B, Habetler TG (2019) Machine learning and deep learning algorithms for bearing fault diagnostics—a comprehensive review. arXiv preprint [arXiv:1901.08247](https://arxiv.org/abs/1901.08247)
- Zhang S, Zhang S, Wang B, Habetler TG (2019) Deep learning algorithms for bearing fault diagnostics—a review. In: 2019 IEEE 12th international symposium on diagnostics for electrical machines, power electronics and drives (SDEMPED). IEEE, pp 257–263
- Eren L, Ince T, Kiranyaz S (2019) A generic intelligent bearing fault diagnosis system using compact adaptive 1d CNN classifier. *J Signal Process Syst* 91:179–189
- Xia M, Han G, Zhang Y, Wan J et al (2020) Intelligent fault diagnosis of rotor-bearing system under varying working conditions with modified transfer CNN and thermal images. *IEEE Trans Ind Inform* 17:3488–3496
- Wang D, Guo Q, Song Y, Gao S, Li Y (2019) Application of multiscale learning neural network based on CNN in bearing fault diagnosis. *J Signal Process Syst* 91:1205–1217
- Pan H, He X, Tang S, Meng F (2018) An improved bearing fault diagnosis method using one-dimensional CNN and LSTM. *Strojnicki Vestnik J Mech Eng* 64:443–452
- Ma P, Zhang H, Fan W, Wang C, Wen G, Zhang X (2019) A novel bearing fault diagnosis method based on 2d image representation

- and transfer learning-convolutional neural network. *Meas Sci Technol* 30:055402
9. Huang W, Cheng J, Yang Y, Guo G (2019) An improved deep convolutional neural network with multi-scale information for bearing fault diagnosis. *Neurocomputing* 359:77–92
 10. Qian W, Li S, Yi P, Zhang K (2019) A novel transfer learning method for robust fault diagnosis of rotating machines under variable working conditions. *Measurement* 138:514–525
 11. Hasan MJ, Sohaib M, Kim J-M (2018) 1d CNN-based transfer learning model for bearing fault diagnosis under variable working conditions. In: *International conference on computational intelligence in information system*. Springer, pp 13–23
 12. Zhang R, Tao H, Wu L, Guan Y (2017) Transfer learning with neural networks for bearing fault diagnosis in changing working conditions. *IEEE Access* 5:14347–14357
 13. Zhao B, Zhang X, Li H, Yang Z (2020) Intelligent fault diagnosis of rolling bearings based on normalized cnn considering data imbalance and variable working conditions, *Knowledge-Based Systems*, 105971
 14. Case western reserve university bearing data center website. <http://csegroups.case.edu/bearingdatacenter/pages/download-data-file>. Accessed 10 May 2021
 15. Yuan Z, Zhang L, Duan L, Li T (2018) Intelligent fault diagnosis of rolling element bearings based on HHT and CNN. In: *Prognostics and system health management conference (PHM-Chongqing)*. IEEE, pp 292–296
 16. LeCun Y et al (2015) *Lenet-5*, convolutional neural networks. <http://yann.lecun.com/exdb/lenet>. Accessed 20 May 2021
 17. Bouchain D (2006) Character recognition using convolutional neural networks. *Inst neural Inf Proc* 2007:1–6
 18. Salleh I, Belkourchia Y, Azrar L (2019) Optimization of the shape parameter of RBF based on the PSO algorithm to solve nonlinear stochastic differential equation. In: *2019 5th international conference on optimization and applications (ICOA)*. IEEE, pp 1–5
 19. Wang G, Guo J, Chen Y, Li Y, Xu Q (2019) A PSO and BFO-based learning strategy applied to faster R-CNN for object detection in autonomous driving. *IEEE Access* 7:18840–18859
 20. Han T, Tian Z, Yin Z, Tan AC (2020) Bearing fault identification based on convolutional neural network by different input modes. *J Braz Soc Mech Sci Eng* 42:1–10
 21. Aszemi NM, Dominic P (2019) Hyperparameter optimization in convolutional neural network using genetic algorithms. *Int J Adv Comput Sci Appl* 10:269–278
 22. Deutsch J, He D (2017) Using deep learning-based approach to predict remaining useful life of rotating components. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 48:11–20
 23. Ruan D, Zhang F, Gühmann C (2021) Exploration and effect analysis of improvement in convolution neural network for bearing fault diagnosis. In: *2021 IEEE international conference on prognostics and health management (ICPHM)*. IEEE, pp 1–8
 24. Yuan Z, Zhang L, Duan L, Li T (2018) Intelligent fault diagnosis of rolling element bearings based on HHT and CNN. In: *Prognostics and system health management conference (PHM-Chongqing)*. IEEE, pp 292–296
 25. Gao D, Zhu Y, Wang X, Yan K, Hong J (2018) A fault diagnosis method of rolling bearing based on complex morlet cwt and CNN. In: *Prognostics and system health management conference (PHM-Chongqing)*. IEEE, pp 1101–1105
 26. Randall RB, Antoni J (2011) Rolling element bearing diagnostics—a tutorial. *Mech Syst Signal Process* 25:485–520
 27. Luo C, Jia M, Wen Y (2017) The diagnosis approach for rolling bearing fault based on kurtosis criterion EMD and hilbert envelope spectrum. In: *IEEE 3rd information technology and mechatronics engineering conference (ITOEC)*. IEEE, pp 692–696
 28. Tsao W-C, Li Y-F, Pan M-C (2010) Resonant-frequency band choice for bearing fault diagnosis based on EMD and envelope analysis. In: *8th World congress on intelligent control and automation*. IEEE, pp 1289–1294
 29. Antoni J, Borghesani P (2019) A statistical methodology for the design of condition indicators. *Mech Syst Signal Process* 114:290–327
 30. Tahir MM, Khan AQ, Iqbal N, Hussain A, Badshah S (2016) Enhancing fault classification accuracy of ball bearing using central tendency based time domain features. *IEEE Access* 5:72–83
 31. Ye M, Huang J (2018) Bearing fault diagnosis under time-varying speed and load conditions via speed sensorless algorithm and angular resample. In: *2018 XIII international conference on electrical machines (ICEM)*. IEEE, pp 1775–1781
 32. Niu L, Cao H, He Z, Li Y (2015) A systematic study of ball passing frequencies based on dynamic modeling of rolling ball bearings with localized surface defects. *J Sound Vib* 357:207–232
 33. Saruhan H, Saridemir S, Qicek A, Uygur I (2014) Vibration analysis of rolling element bearings defects. *J Appl Res Technol* 12:384–395
 34. Kong F, Huang W, Jiang Y, Wang W, Zhao X (2018) A vibration model of ball bearings with a localized defect based on the hertzian contact stress distribution. *Shock Vib* 2018:1–14
 35. Ruan D, Song X, Gühmann C, Yan J (2021) Collaborative optimization of CNN and GAN for bearing fault diagnosis under unbalanced datasets. *Lubricants* 9:105
 36. Jin T, Yan C, Chen C, Yang Z, Tian H, Wang S (2021) Light neural network with fewer parameters based on CNN for fault diagnosis of rotating machinery. *Measurement* 181:109639
 37. Bhadra R, Dutta S, Kedia A, Gupta S, Panigrahy PS, Chattopadhyay P (2018) Applied machine learning for bearing fault prognostics. In: *IEEE applied signal processing conference (ASP-CON)*. IEEE, pp 158–162
 38. Gao D, Zhu Y, Wang X, Yan K, Hong J (2018) A fault diagnosis method of rolling bearing based on complex morlet cwt and CNN. In: *Prognostics and system health management conference (PHM-Chongqing)*. IEEE, pp 1101–1105
 39. Wen L, Li X, Gao L, Zhang Y (2017) A new convolutional neural network-based data-driven fault diagnosis method. *IEEE Trans Ind Electron* 65:5990–5998
 40. Shijie J, Ping W, Peiyi J, Siping H (2017) Research on data augmentation for image classification based on convolution neural networks. In: *Chinese automation congress (CAC)*. IEEE, pp 4165–4170
 41. Lessmeier C, Kimotho JK, Zimmer D, Sextro W (2016) Condition monitoring of bearing damage in electromechanical drive systems by using motor current signals of electric motors: a benchmark data set for data-driven classification. In: *PHM Society European conference (PHME-Bilbao)*, vol. 3
 42. Ruan D, Wang J, Yan J, Gühmann C (2023) Cnn parameter design based on fault signal analysis and its application in bearing fault diagnosis. *Adv Eng Inform* 55:101877

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

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.