Multi-scale deep intra-class transfer learning for bearing fault diagnosis

Xu Wang[a], Changqing Shen[a,*], Min Xia[b], Dong Wang[c], Jun Zhu[d], Zhongkui Zhu[a]

[a] *School of Rail Transportation, Soochow University, Suzhou 215131, China*
[b] *Department of Engineering, Lancaster University, Lancaster LA1 4YW, United Kingdom*
[c] *The State Key Laboratory of Mechanical Systems and Vibration, Shanghai Jiao Tong University, Shanghai 200240, PR China*
[d] *Department of Industrial Systems Engineering and Management, National University of Singapore, Singapore*

ABSTRACT

The tremendous success of deep learning in machine fault diagnosis is dependent on the hypothesis that training and test datasets are subordinated to the same distribution. This subordination is difficult to meet in practical scenarios of industrial applications. On the one hand, the working conditions of rotating machinery can change easily. On the other hand, vibration data and labels are difficult to obtain to train a specific model for each working condition. In this study, we solve these problems by constructing a novel deep transfer learning model called multi-scale deep intra-class adaptation network, which first uses the modified ResNet-50 to extract low-level features and then constructs a multiple scale feature learner to analyze these low-level features at multiple scales and obtain high-level features as input for the classifier. Pseudo labels are then computed to shorten the conditional distribution distance of vibration data collected under different working loads for intra-class adaptation. The proposed method is validated using two datasets to recognize the bearing normal state, the inner race, the ball and outer race faults, and their fault degrees under four different working loads. The high-precision diagnosis results of 24 transfer learning experiments reveal the reliability and generalizability of the constructed model.

Keywords:
Fault diagnosis
Deep transfer learning
Multi-scale feature learner
Intra-class adaptation

1. Introduction

Fault diagnosis has become an increasingly important part of enhancing the reliability and safety of mechanical systems and can prevent large economic losses and personal injury by monitoring the state of the running machinery equipment, determining when and where failures occurred, and recognizing which kind of fault took place [1]. Therefore, strengthening the monitoring of the working state of the machinery, rotating machinery in particular, and making timely and accurate identification of faults are important in preventing catastrophic accidents. Rotating machinery, including bearings, gearboxes, and motors, has wide applications in industrial society [2, 3]. The failure of rolling bearings, which accounts for 51% of all faults, is a major obstacle to the reliable and safe operation of mechanical systems [4].

Many studies show that equipment vibration signals are an important information source for condition monitoring and diagnosis of equipment [5]. Traditional methods usually rely on the manual extraction of features to analyze and process the vibration signals. Qin [6] constructed a new impulsive wavelet based on the vibration signals of faulty rolling bearing. He et al. [7] proposed a spectrogram of multiple scale stochastic resonance to diagnose the bearing failure. However, these manual feature-based methods, such as short-term Fourier transform [8], sparse representation methods [9], and empirical model decomposition [10], have several disadvantages, including the requirement for certain prior knowledge in the choice of extracted features; however, prior knowledge is difficult to access. Moreover, computing time and difficulty will increase considerably when dealing with massive vibration signals.

To deal with such problems, machine learning is developed for fault diagnosis to extract features automatically and accelerate the speed of computing. Numerous studies have demonstrated that fault diagnosis methods based on artificial intelligence are capable of processing massive vibration data and recognizing the health status of machinery [11]. The framework of fault diagnosis based on data mining generally consists of four main steps, namely, data collection, model building, model training, and model testing. Kuncan et al. [12] applied one-dimensional ternary patterns to extract fault features for bearing fault diagnosis and then achieved high diagnostic accuracy. Wang et al. [13] constructed a four-layer batch-normalized stack autoencoder (SAE) and used the frequency spectra of raw data as input to achieve the fast and intelligent health identity of machines. An autoencoder for enhancing

[*] Corresponding author.

noise reduction was constructed by Meng et al. [14] for the implementation of fault diagnosis and replacement of L2 regularization with elastic net regularization to strengthen the sparsity of parameters involved in training. Li et al. [15] integrated sparsity and neighborhood theories with deep extreme learning machines to deal with rotor fault diagnosis experiments. Liu et al. [16] presented a new model for the fault diagnosis of electric machines by inserting a dislocated layer before the convolution operation to enable the convolutional neural network (CNN) to deal with mechanical periodic signals. Zhu et al. [17] combined the outputs of the last convolution layer with the last pooling layer as the input of the fully connected layer to ensure that the model can analyze the features of different levels and calculate the remaining service life of bearings. An end-to-end fault diagnosis method embedded with an adaptive deep belief network (DBN) using the Nesterov moment optimization was designed by Xie et al. [18] to distinguish the states of signals collected from their self-made test platform effectively. Chen et al. [19] applied SAE for feature fusion and then trained a DBN for fault recognition. Wang et al. [20] modified a CNN model by constructing a structure with five different kinds of layers to deal with the fused images converted from vibration signals collected from different directions.

Traditional deep learning methods work effectively in the situation where the training and test datasets obey the same distribution [21]. These cases are rare in practical application scenarios, especially in fault diagnosis. The complex and changeable working conditions of rotating machinery make traditional deep learning-based fault diag-nosis methods unsuitable for handling the vibration data being sub-jected to different distributions. Moreover, meeting the required quantity of vibration data under different health states and corre-sponding labels to have enough samples for training a specific mode is typically difficult to obtain. Transfer learning has attracted considerable attention from researchers to address these problems and aim to seek invariability between different domains by reducing the distance be-tween them. Therefore, transfer learning tries to enhance the general-ization ability and robustness of the model by utilizing samples from the source (marked with labels) and target (few or no samples are marked with labels) domains. Dai et al. [22] developed boosting-based learning algorithms and combined them with semi-supervised transfer tasks by iteratively assigning weights to the samples. Shen et al. [23] applied singular value decomposition to learn features and adopted the TrAdaBoost algorithm to realize the diagnosis of bearing failure. By using a pre-trained neural network highlighted in the ImageNet Large Scale Visual Recognition Challenge 2012 as a feature extractor, Cao et al. [24] fed these extracted features to a new classifier trained with the experimental data of gear fault. Given that samples with no labels exist in the target domain of unsupervised transfer learning, Guo et al.[25] added a domain classifier on the basis of the health state classifier, which renders the model incapable of identifying which of the two domains (i.e., the source and target domains) the samples belong to and applied maximum mean discrepancy (MMD) to draw the domains closer to one another. Wen et al. [26] constructed a sparse autoencoder embedded with maximum mean discrepancy to find the similarities between the source and target domains for fault diagnosis. Yang et al. [27] computed the distance of the output of every convolution layer and integrated it into the objective function for optimization to further expand the role of MMD in closing the distance between the different domains. Except for MMD, which is used to minimize the distributional difference in the different domains, CORAL loss [28] and Wasserstein distance [29] are frequently applied in many transfer tasks.

However, most unsupervised transfer learning algorithms focus on marginal distributions while ignoring the conditional distributions of different domains, that is, studies have focused on the means for minimizing the distance of the entire domain rather than the distance between the same category from different domains or also called intra-class transfer. We develop a deep transfer model called the multi-scale deep intra-class adaptive network (MDIAN) by considering conditional distributions and improving the model's ability to adapt to various working conditions. Motivated by [30] and [31], this work presents MDIAN to address the problem of domain shift for fault diagnosis. First, the modified ResNet-50 is applied to extract low-level features automatically. Second, the multi-scale feature learner learns the high-level features from different scales. Finally, a classifier is optimized via the standard mini-batch algorithm of stochastic gradient descent. Meanwhile, the conditional distribution distance between the high-level features of samples from the different domains is reduced during the training process, and the results of stable high-precision diagnosis are obtained in 24 transfer tasks. The main contributions of this document are summarized as follows:

1) A pre-trained deep model originating from ImageNet 2015 is used to learn the low-level features automatically. This model can be easily extended to other fault diagnostic tasks through our transformation.
2) To deal with the domain shift, the distance between different domains is measured by the conditional maximum mean discrepancy, which forms part of the objective loss function for optimization.
3) A multiple scale feature extractor is constructed to decrease information loss and obtain high-level features by further analyzing low-level features from multiple scales. At the same time, the extractor can be embedded in most networks to improve classification accuracy.

The remainder of this document is organized as follows. The background knowledge of basic methods is presented in Section II. A novel fault diagnosis model called MDIAN is proposed in Section III. The outcomes of bearing experiments among the different transfer tasks and their corresponding analysis are discussed in Section IV. Section V discusses the contribution of this paper and the future work.

2. Theoretical background

2.1. *Unsupervised transfer learning*

Transfer learning refers to a learning process that uses the similarity between data, tasks, or models to apply the models learned in the old domain (source domain) to the new domain (target domain). According to whether the samples in the target domain are labeled or not, transfer learning can be divided into the following categories: 1) supervised transfer learning, 2) semi-supervised transfer learning, and 3) unsupervised transfer learning [32]. We focus on addressing the fault diagnosis problem wherein the target domain has no labeled samples and different data distribution from the source domain because of the variable load of the rotating machinery. The main task of our proposed model is to acquire useful knowledge learned from both the source domain consisting of examples marked with labels and the target domain consisting of examples without any labels and classify these unlabeled samples from the target domain correctly. For convenience, we use $D_s = \{(x_i^s, y_i^s)\}_{i=0}^{n_s-1}$, where $y_i^s \in \{0, 1, 2, ..., C-1\}$ represents the source domain having examples with C kinds of different labels, and $D_t = \{x_i^t\}_{j=0}^{n_t-1}$denotes the target domain having examples without labels. The source domain data are collected under the probability distribution P_s, the target domain data are collected under the probability distribution P_t, and $P_s \neq P_t$. This study aims to train a model $y = F(x)$ using D_s and D_t, which can forecast the true labels of the examples without any labels in the target domain as accurately as possible by seeking the common features belonging to both the source and target domains.

2.2. *Convolutional neural network*

A deep CNN can automatically extract senior features from the original images. A representative CNN contains three kinds of layers, namely, convolution, pooling, and fully connected layers.

Convolution operation can generate N-channel feature maps through N convolution kernels, and the dot product between the kernels and the input can be represented as follows:

$$y_{k,i,j} = \sum_{r=0}^{C-1}\sum_{p=0}^{L-1}\sum_{q=0}^{W-1} w_{k,r,p,q}x_{r,p+i,q+j} + b \tag{1}$$

where $y_{k,i,j}$ is the value at the position (i, j) in the k-th $(0 \leq k \leq N-1)$ channel of the output feature maps; $\mathbf{x}$ is the input image with C channels, $x_{r,p,q}$ is the node in the r-th channel of the input image with location (p, q); $w_{k,r,p,q}$ is the weight of the k-th convolution kernel at the location (p,q) in the r-th channel,b is the kernel bias, $C \times L \times W$ is the size of the convolution kernel, and C represents the amount of the channels of the convolution kernel, which is equal to the number of channels for the input image.

A pooling layer is executed to prevent overfitting after the operation of the convolution. Mathematically, a pooling operation can be defined as

$$y_{k,i,j} = pooling(x_{k,L,W}) \tag{2}$$

where $y_{k,i,j}$ represents the value at the location (i, j) in k-th channel of the output feature maps after the operation of down-sampling;$x_{k,L,W}$ denotes a rectangular region with L and W as its length and width, respectively, in the k-th channel of the input features maps before pooling; and pooling(•) refers to the pooling rule. Maximum and average pooling are commonly used pooling approaches.

A fully connected layer is constructed in the traditional CNN structure after a series of convolution and pooling operations, which not only reduces the training speed but also makes it easy for overfitting to occur. Hinton et al. [33] established the dropout method, which can effectively prevent overfitting by setting a portion of the activations to zero in the training process.

In this article, the global average pooling [34] is used to take the place of the traditional fully connected layer after the last convolution layer in the network. The operation of global average pooling can be defined as

$$\mathrm{y}_k = pooling_{avg}(\mathbf{x}_k) \tag{3}$$

where $\mathbf{x}_k$ refers to the k-th channel of the input $\mathbf{x}$, $pooling_{avg}(\cdot)$ denotes the operation of calculating the average value of all feature points in $\mathbf{x}_k$, and y_k represents the k-th value of the output.

The feature map of each channel is connected to every neuron in the next layer in the traditional structure of CNN. No parameter requires optimization in the operation of global average pooling, which can be realized in any input feature map without considering their size.

Finally, a softmax function is utilized to obtain a classification result of the input. A softmax function can be expressed as:

$$q_i = \frac{e^{vi}}{\sum_{i=0}^{C-1} e^{vi}} \tag{4}$$

where q_{i} indicates the probability that the sample may be marked with label i, v_{i} denotes the input of classifier, and C represents the total quantity of categories.

The loss of a typical CNN is expressed as

$$loss_{classifier}(\mathbf{y}, \mathbf{X}) = \frac{1}{n}\sum_{i=0}^{n-1} J(y_i, f(\mathbf{x}_i)) \tag{5}$$

where n is total amount of samples involved in the training progress, $\mathbf{x}_i$ is the i th sample, y_i is the true label, $f(\mathbf{x}_i)$ is the output of the network, that is, the computed result of the network according to $\mathbf{x}_i$, and J (•,•) is the classification loss measured by the following cross-entropy function:

$$J(p, q) = -\sum_{i=0}^{C-1} pi\log(qi) \tag{6}$$

where p_i is equal to 1 when i is its real label and 0 if otherwise, q_i is the output probability after the softmax activation function, and C is the total number of all kinds of labels.

2.3. *Maximum mean discrepancy*

To achieve an appropriate function, that is, $\mathcal{F}(\bullet)$, many transfer learning methods focus on reducing the distribution difference $d(\mathbf{X}_s, \mathbf{X}_t)$ between the source and target domains. A nonparametric distance measure called MMD [30] has been frequently used in many transfer tasks and can estimate the difference in marginal distributions as follows:

$$d_{\mathcal{H}}(\mathbf{X}_s, \mathbf{X}_t) = \|\frac{1}{n_{\mathrm{s}}}\sum_{i=0}^{n_{\mathrm{s}}-1}\Phi(\mathbf{x}_i^{\mathrm{s}}) - \frac{1}{n_{\mathrm{t}}}\sum_{j=0}^{n_{\mathrm{t}}-1}\Phi(\mathbf{x}_j^{\mathrm{t}})\|_{\mathcal{H}}^2 \tag{7}$$

where $\mathcal{H}$ denotes the reproducing kernel Hilbert space, and Φ(•) refers to the function of the feature space map. Minimizing Eq. (7) can draw the source and target domain closer so that the model trained on D_s and D_t can forecast the labels of samples from target domain more accurately.

3. Proposed new multi-scale deep intra-class transfer learning network for bearing fault diagnosis

Given that the existing deep learning models are not applicable to the cross-domain problems with complex working conditions of rotating machinery, this study designs a novel network called MDIAN from the perspective of extracting as many invariant features as possible from the source and target domains. The detailed structure of MDIAN and the general procedure of mechanical fault diagnosis based on the proposed model will be introduced in this section.

3.1. *Structure of the proposed model*

The structure of the deep transfer neural network constructed in this study is illustrated in Fig. 1. Fast Fourier transform is abbreviated as FFT.

MDIAN is mainly composed of three parts, namely, modified ResNet-50, multiple scale feature extractor, and classifier. First, the modified ResNet-50 extracts low-level features from preprocessed data. Second, the multiple scale feature extractor draws high-level features on the basis of low-level features. Finally, a classifier is applied for fault diagnosis on the basis of high-level features. Furthermore, during the training progress, the model also extracts high-level features of samples from the target domain and gives out pseudo labels, which are utilized to reduce the conditional distribution distance of the two different domains (intra-class adaptation).

3.1.1. *Modified resnet-50*

Fig. 1 shows that ResNet-50 [35] is applied to extract the low-level features before the multiple scale feature extractor draws the high-level features, which are more discriminating. The pre-trained model in our study, ResNet-50, was successfully applied in ImageNet in 2015. This model has strong generalizability and specific information of ResNet-50, as shown in . Fig. 2 shows the specific internal structure of RESBLOCK in ResNet-50. Batch normalization and rectified linear units are abbreviated as BN and ReLU, respectively.

The original ResNet-50, which has been trained in ImageNet in 2015 and achieved remarkable success, can be used to address the vibration signal of the rotating machinery and identify the state of the health as a complete network. In our study, we only use ResNet-50 to extract low-level features and ignore the function of the classification. Hence, we modify ResNet-50 through the cancellation of its last two layers, namely, the average pooling and the fully connected layer and replacing them with the multiple scale feature extractor as shown in Fig. 1. This modification can prevent the fully connected layer from

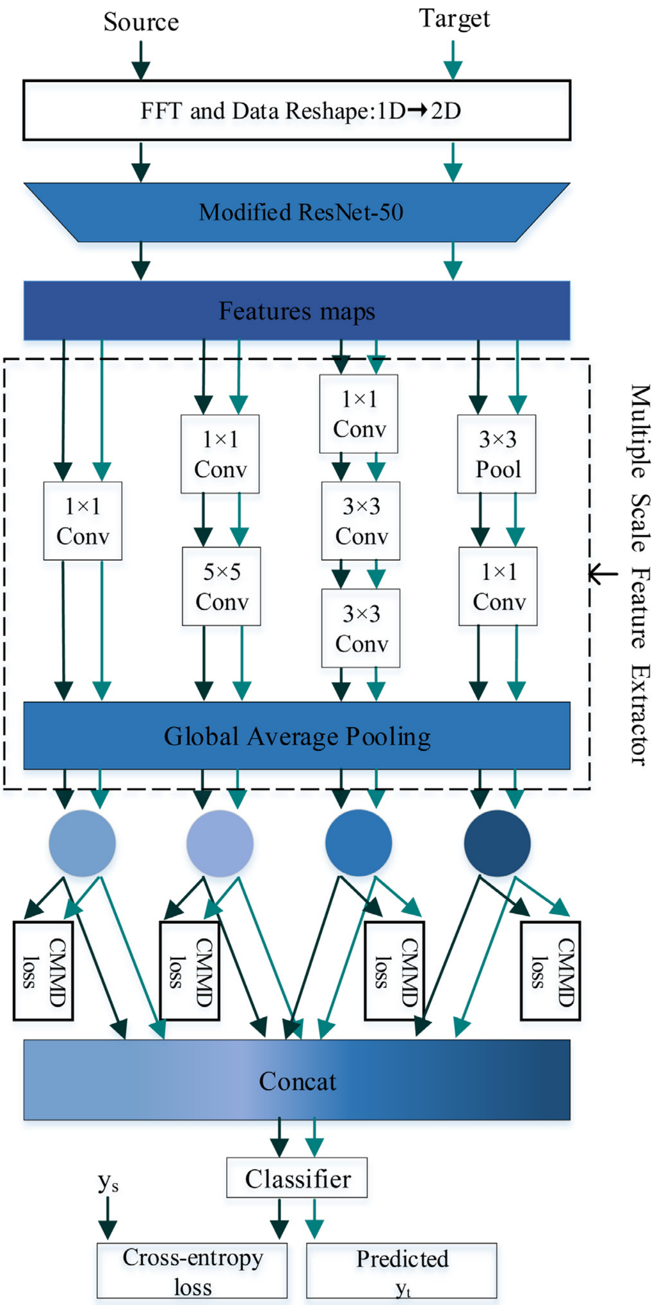


Fig. 1. Structure of the MDIAN network.

destroying the data structure obtained by the convolution layers as well as extract the high-level features from multiple angles to avoid information loss. The four feature maps extracted by the four substructures are then concatenated together as a vector and input into the new classifier whose number of output neurons is exactly the same as the number of possible working states of the machine. Training such a deep neural network takes an extended amount of time and requires a large quantity of labeled data. Fortunately, we can start with ResNet-50, which has already been trained on ImageNet in 2015 and train it to meet the classification needs of fault diagnosis.

3.1.2. *Multiple scale feature extractor*

Convolution kernels are crucial to any deep neural network composed of CNNs. The receptive field [36] used to represent the size of the area in the original input mapped by each feature point on the feature map of each layer is influenced by the size of convolution kernels. The large kernel size widens the coverage of the receptive field. The receptive field can detect local information of the entire input, which we call piece. Certain pieces are parts of the object we want to recognize, other pieces are completely mismatched with the object we want to recognize, or even worse, some pieces may mislead the neural networks. Then, the fully connected layers are constructed to put all the pieces together. Therefore, the type of input can be identified easily. On the one hand, the large receptive field enlarges the piece and may cause the piece to contain many useless features while additional detailed

Table 1
Complicated framework of the ResNet-50 structure.

Layer name	Output size	Channels × kernel size
Input	3 × 224×224	–
Conv1	64×112×112	64×7 × 7, stride=2
BN, Relu	64×112×112	–
Max pool	64×56×56	64×3 × 3, stride=2
RESBLOCK 1: Conv2_x	256×56×56	$\begin{bmatrix} 64\times1\times1 \\ 64\times3\times3 \\ 256\times1\times1 \end{bmatrix}\times3$
RESBLOCK 2 : Conv3_x	512×28×28	$\begin{bmatrix} 128\times1\times1 \\ 128\times3\times3 \\ 512\times1\times1 \end{bmatrix}\times4$
RESBLOCK 3 : Conv4_x	1024×14×14	$\begin{bmatrix} 256\times1\times1 \\ 256\times3\times3 \\ 1024\times1\times1 \end{bmatrix}\times6$
RESBLOCK 4 : Conv5_x	2048×7 × 7	$\begin{bmatrix} 512\times1\times1 \\ 512\times3\times3 \\ 2048\times1\times1 \end{bmatrix}\times3$
Relu	2048×7 × 7	–
Average pool	2048×1	2048×7 × 7
1000-d fc, softmax	1000	–

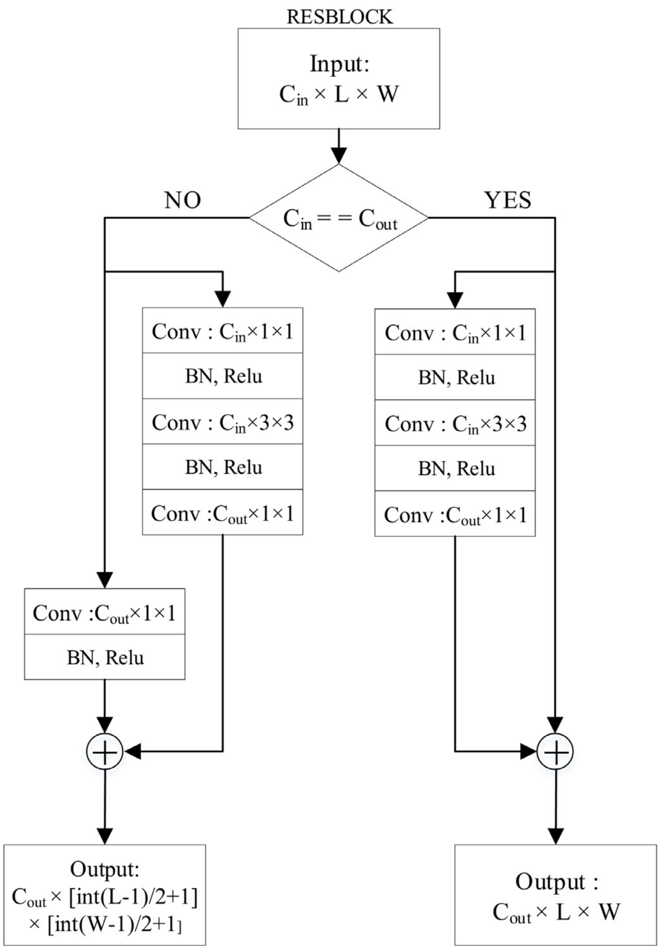


Fig. 2. Internal structure of RESBLOCK.

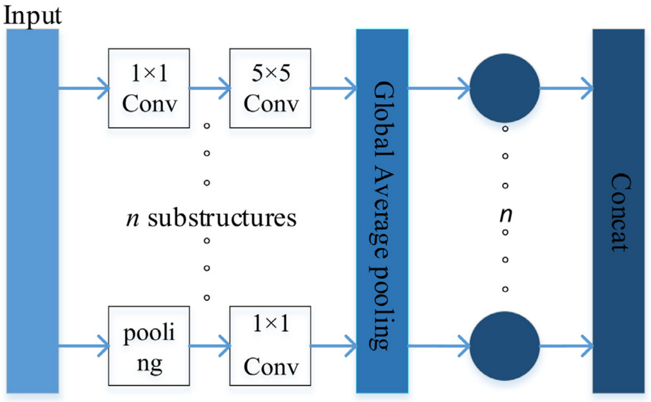


Fig. 3. Structure of the multiple scale feature extractor.

information could be ignored. On the other hand, the narrow receptive field reduces the size of the piece. Then, the fully connected layer may be unable to put together the object that needs to be recognized with such small pieces and lose many important features. Hence, the size of the convolution kernel, which affects the size of the pieces, is crucial to the accuracy of the results calculated by the entire network. Although different convolution kernels with different sizes are used during the forward propagation process, the fixed size of the receptive field indicates that the entire network deals with input from one single scale. We propose a multiple scale feature extractor, which can simultaneously obtain different pieces with different sizes, to enable the model to view the input from multiple scales. The structure of the multiple scale feature extractor is shown in Fig. 3.

Once the feature extractor receives the input signal, n kinds of different operations, including convolution and pooling with different kernel sizes, will be performed on the input at the same time. After the global average pooling layer, n kinds of different features or n types of receptive fields are obtained through the n substructures. These n features are then concatenated into a vector. This process can be expressed as follows:

$$g(\mathbf{x}) = [g_0(\mathbf{x}), g_1(\mathbf{x}), \cdots, g_{n-1}(\mathbf{x})] \tag{8}$$

where$g_i(\mathbf{x})$refers to the result after global average pooling in the i th substructure and n is equal to the number of substructures and$g(\mathbf{x})$denotes the input of the classifier. Similarly, the distance measured between the source and target domains of all the substructures can be expressed as follows:

$$d(\mathbf{X}_s, \mathbf{X}_t) = \sum_{i=0}^{n-1} d(\mathrm{g}_i(\mathbf{X}_s), \mathrm{g}_i(\mathbf{X}_t)) \tag{9}$$

Therefore, the concatenated vector contains information on the different receptive fields with different sizes. Thus, the fully connected layers can choose the most suitable pieces with the appropriate size by assigning different weights to put together the object that we want to recognize. Compared with one single-scale feature extraction method, the multiple scale feature extractor can avoid looking at a leopard through a tube or seeing too much scope that causes it to ignore the details. With the replacement of the global average pooling with multiple scale feature extractors in a trained model, we can modify the model into a multi-scale network and then the output of the multiple scale feature extractor can be fed directly into a classifier. Meanwhile, the distance between the source and target domains can be decreased from different scales. Through this modification, the performance of the model can be improved considerably.

3.1.3. Conditional maximum mean discrepancy

As mentioned above, many transfer learning documents have focused on narrowing the MMD between the target and source domains. However, transfer component analysis [30] assumes that as long as the marginal distributions of two different domains are close, that is, $P_s(\mathbf{x_s}) \approx P_t(\mathbf{x_t})$, the conditional distributions of the two different domains are alike, that is, $P_s(y_s|\mathbf{x}_s) \approx P_t(y_t|\mathbf{x}_t)$, which also means the conditional distributions of D_s and D_t are often ignored. In other words, most published studies have focused on learning the global domain shift without considering the intra-class similarity (intra-class transfer) [37]. Therefore, we use the conditional distributions instead of marginal distributions of the source and target domains to narrow the differences between them. Minimizing the discrepancy between the conditional distributions of Ds and Dt can contribute considerably to the generalizability and robustness of the network we trained [38] and additional useful information can be delivered to the target domain.

However, matching the conditional distributions between $P_s(y_s|\mathbf{x}_s)$ and $P_t(y_t|\mathbf{x_t})$ seems impossible given that all examples from the target domain have no labels, that is, $\mathbf{y}_t$ cannot available. Interestingly, the unlabeled samples from the target domain can use the outputs of the transfer network$\hat{\mathbf{y}}_t = \mathcal{F}(\mathbf{X}_t)$as pseudo labels. Hence, MMD can be modified as follows:

$$\hat{d}_{\mathcal{H}}(\mathbf{X}_s, \mathbf{X}_t) = \frac{1}{C}\sum_{c=0}^{C-1} \| \frac{1}{n_s^{(c)}} \sum_{i=0}^{n_s^{(c)}-1} \Phi(\mathbf{x}_i^{s(c)}) - \frac{1}{n_t^{(c)}} \sum_{j=0}^{n_t^{(c)}-1} \Phi(\mathbf{x}_j^{t(c)}) \|_{\mathcal{H}}^2 \tag{10}$$

where$\mathbf{x}_i^{s(c)}$represents the i th sample of samples with label c from the source domain, $n_s^{(c)}$is equal to the amount of all samples with label c from the source domain, $\mathbf{x}_j^{t(c)}$is the j-th sample of the samples with pseudo label c from the target domain, and$n_t^{(c)}$is equal to the quantity of all samples with pseudo label c from the target domain. The modified MMD or Eq. (10), also called CMMD, is applied to estimate the difference between the intra-class conditional distributions $P_s(\mathbf{x}_s|\ y_s = c)$ and $P_t(\mathbf{x}_t|\ y_t = c)$. By minimizing Eq. (10), the conditional distributions of the source and target domains are pulled closer. Although we used pseudo labels for the target domain in our training process, the correctness of the pseudo labels will improve gradually during the optimization to ensure that the difference between the intra-class conditional distributions $P_s(\mathbf{x}_s|\ y_s = c)$ and $P_t(\mathbf{x}_t|\ y_t = c)$ can be smaller.

3.2. Loss function of multi-scale high-level feature alignment

To enhance the generalizability and robustness of the network in tasks of unsupervised transfer learning, we developed a multiple scale intra-class transfer adaptive network, which can learn high-level features at multiple scales and simultaneously align these high-level features of the source and target domains. The standard mini-batch algorithm of stochastic gradient descent (SGD) is used when training to achieve an appropriate function$\mathcal{F}(\bullet)$. We use$\mathrm{g}(\bullet)$ to denote the output of one substructure in the multiple scale feature extractor to ensure that the objective loss function is formulated as follows:

$$\begin{aligned} loss_{total} &= \min loss_{classifier}(\mathbf{y}_s, \mathbf{X}_s) + \lambda \sum_{i=0}^{n_{sub}-1} d(\mathrm{g}_i(\mathbf{X}_s), \mathrm{g}_i(\mathbf{X}_t)) \\ &= \min_{\mathcal{F}} \frac{1}{n_s} \sum_{i=0}^{n_s-1} J(y_i^s, \mathcal{F}(\mathbf{x}_i^s)) + \lambda \sum_{i=0}^{n_{sub}-1} \hat{d}_{\mathcal{H}}(\mathrm{g}_i(\mathbf{X}_s), \mathrm{g}_i(\mathbf{X}_t)) \end{aligned} \tag{11}$$

where $J(\bullet,\bullet)$ is the cross-entropy loss; λ is a hyper parameter, $\lambda > 0$; n_{sub} is equal to the number of substructures in the multiple scale feature extractor; $\hat{d}_{\mathcal{H}}(\cdot,\cdot)$ is the conditional distributions, that is, the CMMD mentioned above. By minimizing Eq. (11), the proposed network$\mathcal{F}(\bullet)$ can be trained to be capable of predicting accurately the labels of samples from the target domain.

3.3. General procedure of the proposed system

The complex working conditions of rotating machinery make the traditional deep learning method unsuitable for the correct identification of the machine status. Hence, we develop a deep transfer neural network called MDIAN, which can analyze the original vibration signals from multiple scales and be adapted to different application scenarios.

Generally, the different fault diagnosis methods embedded with deep neural networks contain the three steps of data collection, model training, and model testing. As to our method, first, the vibration data of rotating machinery under different working conditions are collected and then the collected data subordinate to different distributions are cut into segments and marked with different labels according to the health states, that is, samples with labels. Second, the samples with labels from the source domain together with the samples without labels from the target domain are fed into the proposed network to complete the training process. Finally, to estimate the properties of the proposed model, the samples from the target domain need to be classified as input, and the trained model will give out the predicted labels for the input samples, which can be compared with the marked labels for input samples. Notably, the samples from the target domain should be marked with true labels in the first step. These labels are not part of the training process and are only used to calculate the accuracy of the trained model in the last step. The overall process of fault diagnosis is illustrated in Fig. 4.

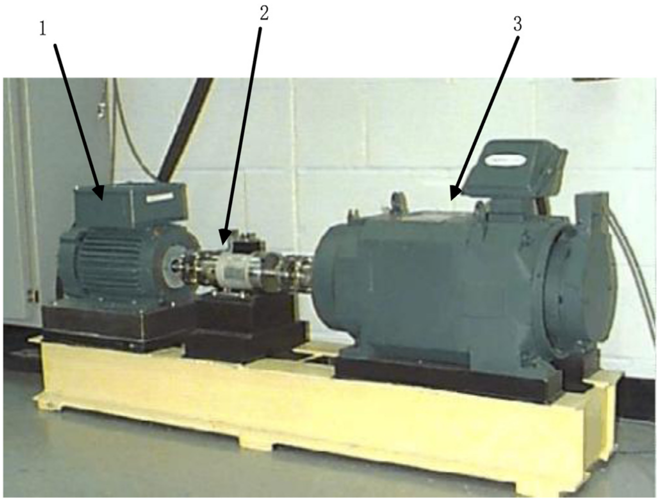


Fig. 5. Test rig: (1) motor, (2) torque transducer/encoder, and (3) dynamometer.

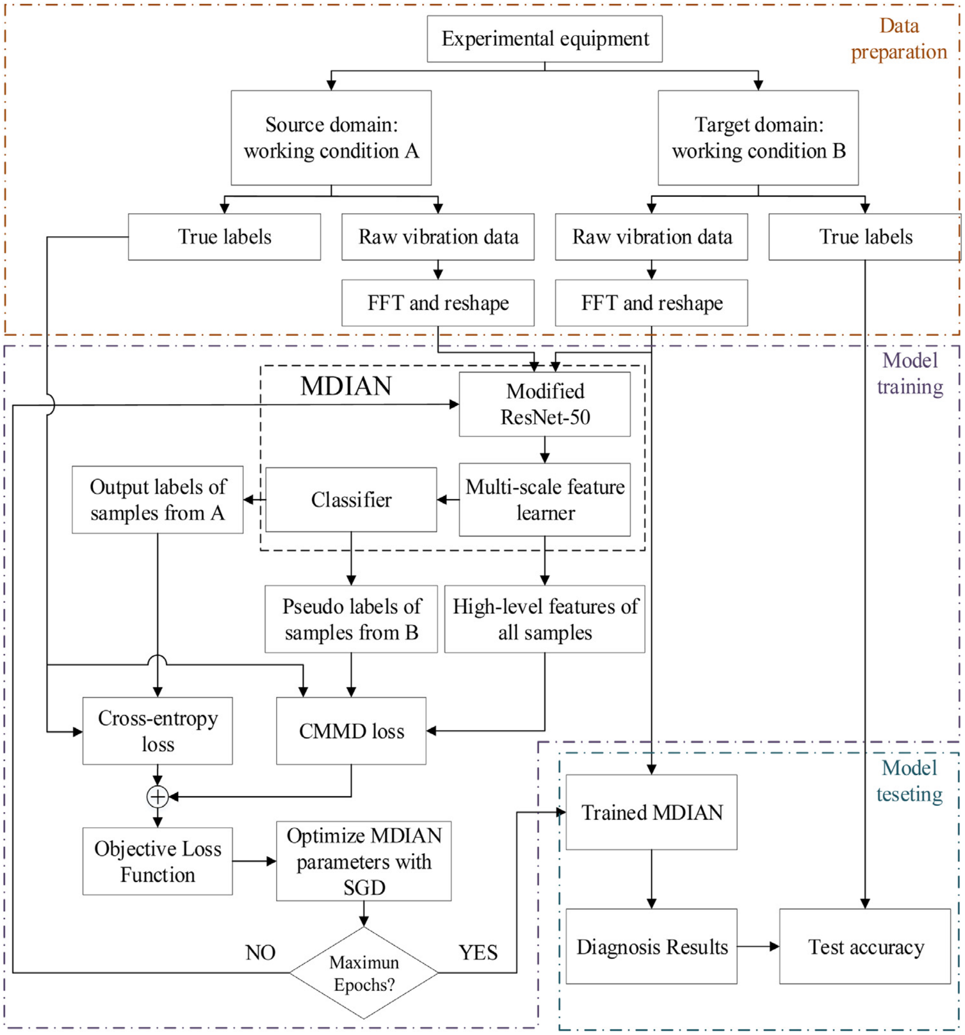


Fig. 4. General procedure of mechanical fault diagnosis embedded with MDIAN.

Table 2
Description of the 10 health states in each domain.

Fault diameter/in.	Heath state	Class label	Number of samples	Description
–	Normal	0	200	NO
0.007	Inner race	1	200	IF07
0.007	Ball	2	200	BF07
0.007	Outer race	3	200	OF07
0.014	Inner race	4	200	IF14
0.014	Ball	5	200	BF14
0.014	Outer race	6	200	OF14
0.021	Inner race	7	200	IF21
0.021	Ball	8	200	BF21
0.021	Outer race	9	200	OF21

4. Experimental validations

4.1. Case I

In this section, several experiments are conducted to test the performance of the proposed MDIAN and verify its generalizability and robustness. The bearing data used in this case were extracted from CWRU [39]. Fig. 5 shows the test rig. The data acquisition system consisted of three main parts from left to right, namely, a 2-hp motor, a torque transducer, and a dynamometer. Vibration data were collected with an accelerometer. The vibration data used in this case were measured from the drive-end bearings at a sampling frequency of 12 kHz under 0 to 3-hp loads. Faults were introduced by electro-discharge machining (EDM) at the inner race, the ball element, and the outer race of the bearing, and the fault degrees were 0.007, 0.014, and 0.021 inches, respectively. Therefore, each load had nine fault states and one normal state.

We built four kinds of datasets (0, 1, 2, and 3 hp) under different working conditions, that is, variable loads, to simulate the task of transfer learning. These datasets are named after their working load. For example, dataset 0 hp means the samples come from vibration signals collected under the working load of 0 hp. Hence, the four datasets of variable loads represent the four domains whose distributions differ from one another. Each dataset has 2000 samples and contains 10 different kinds of health states, that is, 2000 samples with 10 different labels exist in each domain, and the number of samples with the same label is 200. The complicated information on the 10 health states is listed in Table 2. The spectrogram figures of 10 samples with different labels in dataset 0 hp are shown in Fig. 6.

Given that the four datasets obey different distribution, 12 experiments of transfer learning tasks have been conducted. Each transfer learning task is denoted by A hp → B hp, which represents the bearing working conditions with loads A and B hp, respectively. Dataset A hp is considered the source domain, which consists of 2000 samples marked with ten kinds of labels. Dataset B hp is regarded as the target domain, which is composed of 2000 unlabeled samples. Each experiment aims to forecast the health states of samples in the target domain, that is, dataset B hp by learning useful knowledge from both datasets A and B hp.

To reveal the effectiveness of the proposed MDIAN network, several simulations using other models are also implemented for comparison. All the methods are listed as follows:

1) Support vector machine, that is, SVM [4]

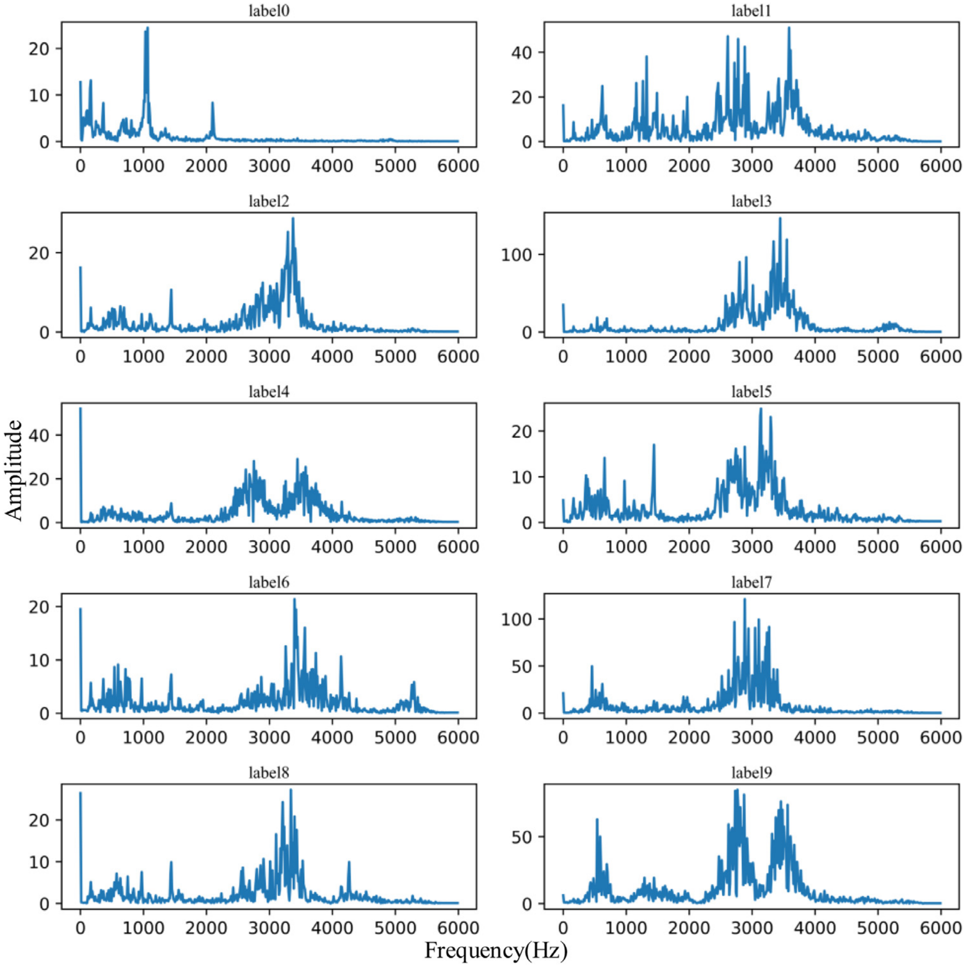


Fig. 6. Spectrogram figures of ten samples with different labels in dataset 0 hp.

Table 3
Experimental results of the seven methods.

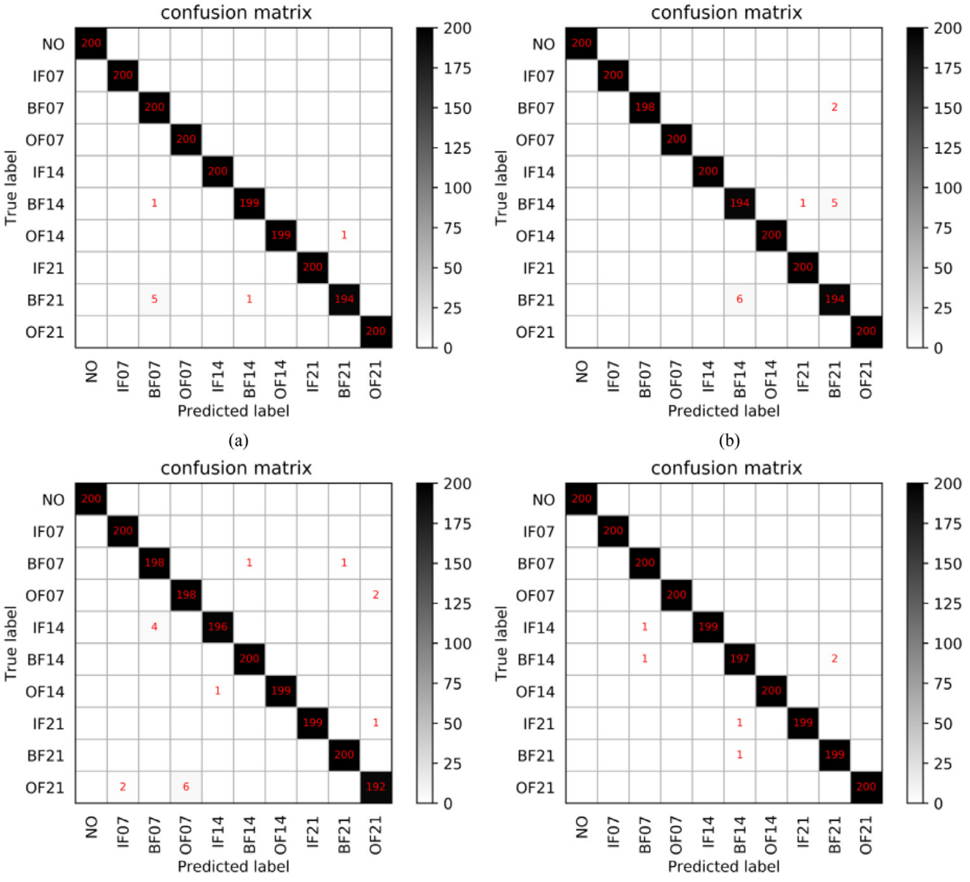
Tasks	SVM	CNN	CNN + MMD	CNN + CMMD	MDDAN	DIAN	MDIAN
0 hp→1 hp	70.70%	72.25%	81.00%	79.60%	87.15%	88.35%	99.60%
0 hp→2 hp	66.45%	70.55%	79.90%	86.70%	90.60%	87.60%	99.30%
0 hp→3 hp	63.40%	62.45%	55.85%	69.25%	91.65%	90.30%	99.10%
1 hp→0 hp	71.30%	87.30%	88.95%	86.05%	84.00%	86.95%	99.70%
1 hp→2 hp	70.00%	89.80%	88.70%	87.45%	92.40%	91.15%	99.65%
1 hp→3 hp	74.00%	74.70%	80.50%	82.25%	94.20%	94.85%	99.80%
2 hp→0 hp	62.85%	60.35%	64.65%	61.90%	87.40%	87.60%	97.60%
2 hp→1 hp	61.60%	75.50%	79.80%	78.15%	91.95%	91.60%	99.45%
2 hp→3 hp	67.65%	84.30%	79.95%	76.55%	91.50%	94.30%	99.45%
3 hp→0 hp	65.30%	66.90%	75.25%	81.90%	84.25%	88.45%	97.45%
3 hp→1 hp	65.70%	81.15%	71.15%	73.25%	87.35%	91.65%	98.60%
3 hp→2 hp	63.25%	74.95%	74.85%	74.85%	92.15%	89.70%	99.50%
mean value	66.85%	75.02%	76.71%	78.16%	89.55%	90.21%	99.10%
standard deviation	0.0392	0.0934	0.0941	0.0760	0.0338	0.0260	0.0080

Table 4
F-score, training accuracy, and testing accuracy of MDIAN.

Tasks	F-score	training accuracy	testing accuracy
0 hp→1 hp	0.9959	100.00%	99.60%
0 hp→2 hp	0.9930	100.00%	99.30%
0 hp→3 hp	0.9909	100.00%	99.10%
1 hp→0 hp	0.9930	100.00%	99.70%
1 hp→2 hp	0.9965	100.00%	99.65%
1 hp→3 hp	0.9929	100.00%	99.80%
2 hp→0 hp	0.9722	100.00%	97.60%
2 hp→1 hp	0.9822	100.00%	99.45%
2 hp→3 hp	0.9944	100.00%	99.45%
3 hp→0 hp	0.9741	100.00%	97.45%
3 hp→1 hp	0.9843	100.00%	98.60%
3 hp→2 hp	0.9959	100.00%	99.60%

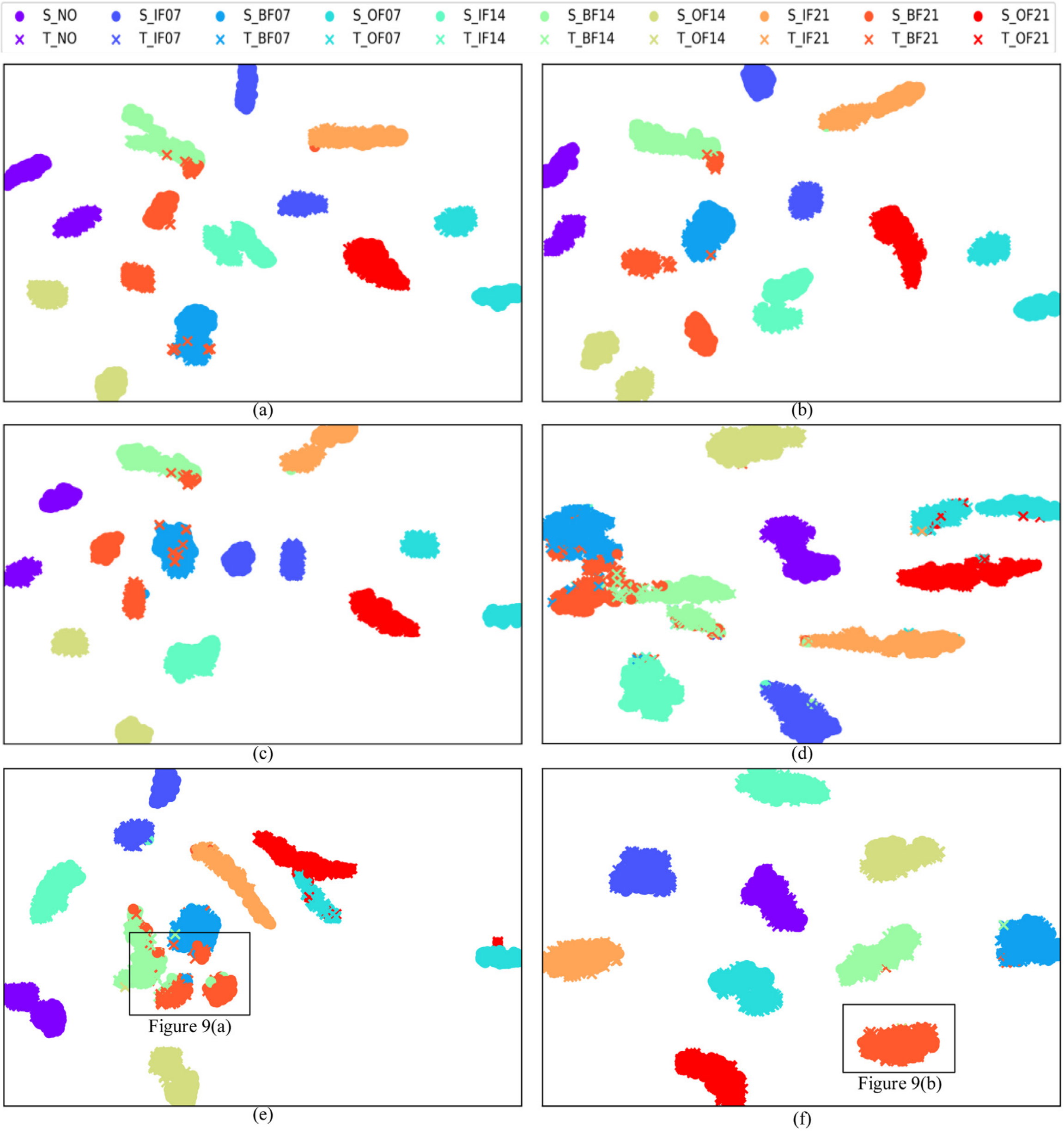
2) Deep CNN, that is, CNN
3) Deep CNN using MMD [30], that is CNN + MMD
4) Deep CNN using CMMD, that is, CNN + CMMD
5) MDIAN using MMD [31] instead of CMMD (multi-scale deep domain adaptive network), that is, MDDAN
6) MDIAN without the multi-scale feature extractor, that is, DIAN
7) MDIAN

Compared with the modified ResNet-50 used in 5, 6, and 7, the same three-layer CNN is constructed in 2, 3, and 4. At the same time, we modify MDIAN into 6 and 7 for further comparison. The performance measure of the seven models is the accuracy rate, which is expressed mathematically as follows:



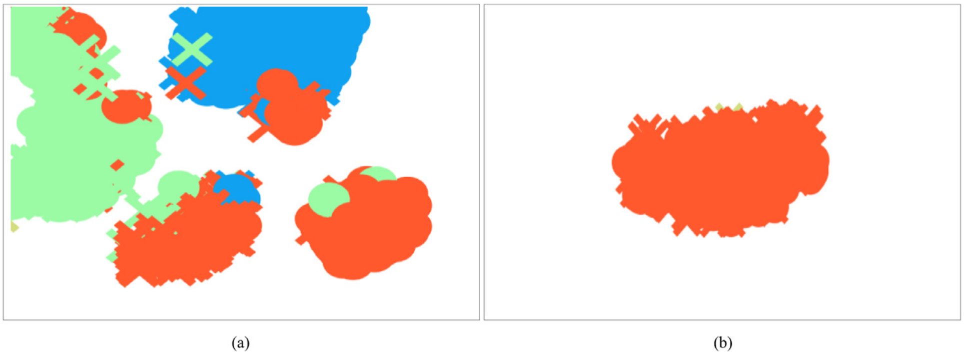
(a)
(b)
(c)
(d)

Fig. 7. Confusion Matrix: (a) 0 hp → 1 hp, (b) 0 hp → 2 hp, (c) 0 hp → 3 hp, and (d) 1 hp → 0 hp.



(a) (b) (c) (d) (e) (f)

Fig. 8. Feature visualization through t-SNE from task 0 hp → 1 hp. (a) CNN, (b) CNN + MMD, (c) CNN + CMMD, (d) MDDAN, (e) DIAN, and (f) MDIAN.

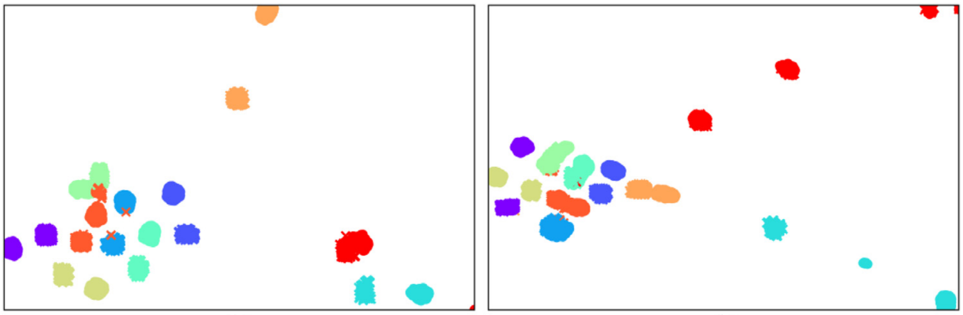


(a) (b)

Fig. 9. Partial enlarged drawing of Fig. 8(e) and (f).

$$\mathrm{acc} = \frac{1}{n_{\mathrm{t}}} \sum_{i=0}^{n_{\mathrm{t}}-1} sign\left(\mathcal{F}(\mathbf{x}_i^{\mathrm{t}}) = y_i^{\mathrm{t}}\right) \tag{12}$$

where$\mathcal{F}(\mathbf{x}_i^{\mathrm{t}})$denotes the predicted label of the i th sample from the target domain, which is given by the trained model; y_{i} represents the true label of the i th sample from the target domain, which is collected in advance but is not involved in the training progress; and$sign(\bullet)$ is the indicator function. The hyper parameterλmentioned in Eq. (11) is set as follows:



(a) (b)

Fig. 10. Feature visualization through t-SNE from task 0 hp → 1 hp. (a) original signal (b) low-level features.

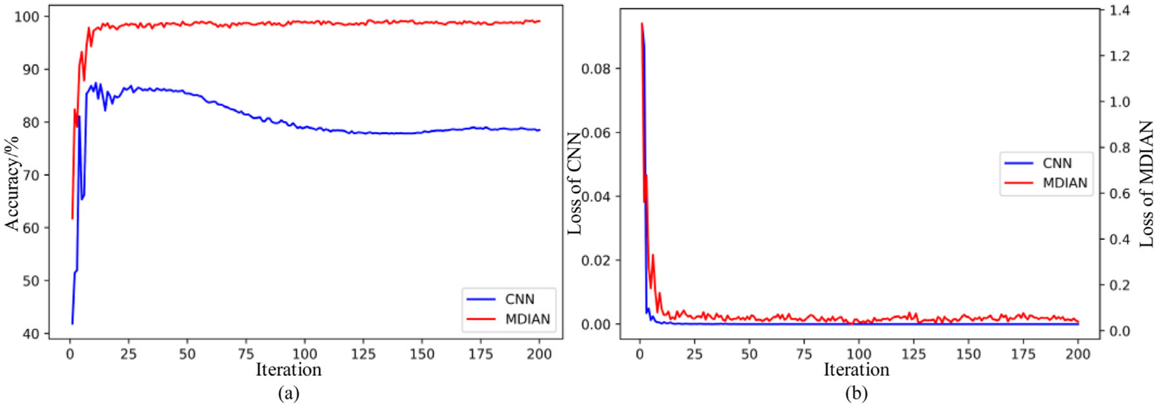


Fig. 11. Trends of (a) testing accuracy and (b) loss in the fine-tuning iterations of the transfer task 1 hp → 0 hp.

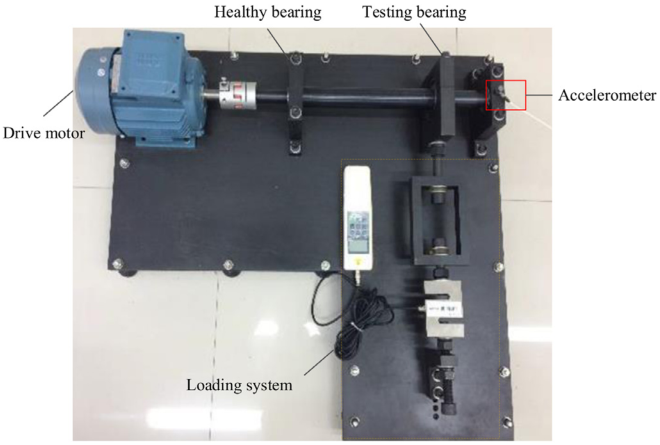


Fig. 12. Self-made test bench.

$$\lambda = \frac{2}{1 + e^{\frac{-10*epoch}{epochs}}} - 1 \tag{13}$$

where *epochs* are equal to the total steps of iterations and *epoch* represents the current iteration step.

The experimental results of these seven methods are listed in Table 3.

F-score, training accuracy, and testing accuracy of MDIAN are listed in Table 4.

Table 3 presents that MDIAN outperforms the other methods. We analyze these results further to confirm the superiority of our model as follows:

- MDIAN achieves the best results on all transfer tasks and dominates the list with an average accuracy rate of 99.06%, indicating the

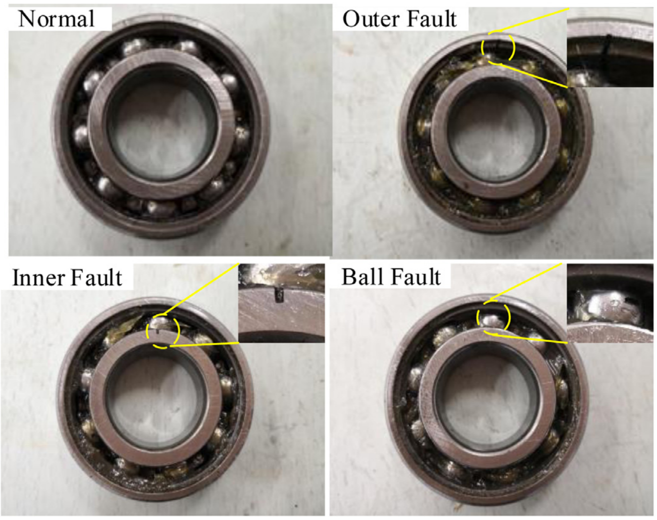


Fig. 13. Pictures of the bearings with four different health states.

Table 5
Description of 7 health states in each domain.

Fault diameter/mm	Heath state	Class label	Number of samples	Description
–	Normal	0	200	NO
0.2	Inner race	1	200	IF2
0.2	Ball	2	200	BF2
0.2	Outer race	3	200	OF2
0.3	Inner race	4	200	IF3
0.3	Ball	5	200	BF3
0.3	Outer race	6	200	OF3

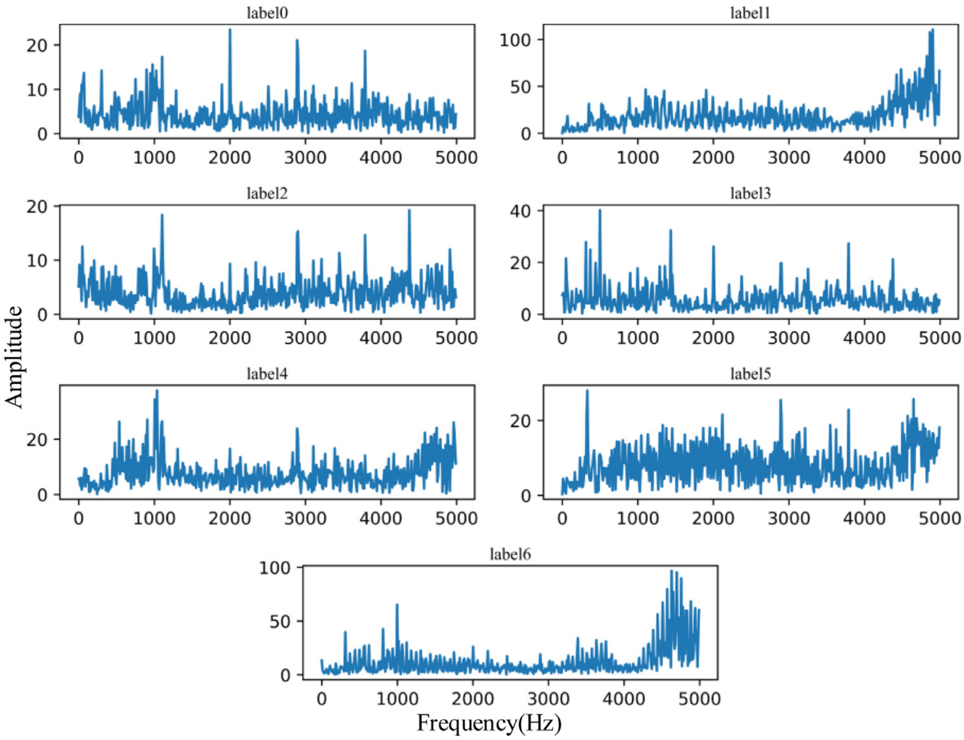


Fig. 14. Spectrum of seven samples with different labels.

Table 6
Experimental results of the seven methods.

Tasks	SVM	CNN	CNN + MMD	CNN + CMMD	MDDAN	DIAN	MDIAN
0 kN→1 kN	19.79%	88.43%	87.86%	87.64%	85.43%	89.29%	92.86%
0 kN→2 kN	15.64%	65.00%	64.36%	62.57%	75.14%	73.00%	90.07%
0 kN→3 kN	14.29%	74.93%	76.57%	77.64%	74.00%	72.57%	90.79%
1 kN→0 kN	47.29%	78.00%	84.43%	81.43%	94.36%	89.78%	97.57%
1 kN→2 kN	25.07%	70.71%	70.86%	70.93%	83.86%	83.07%	99.29%
1 kN→3 kN	18.36%	84.64%	85.00%	85.64%	87.79%	86.46%	99.14%
2 kN→0 kN	36.07%	57.43%	59.36%	53.43%	82.21%	72.79%	93.79%
2 kN→1 kN	40.07%	68.64%	79.79%	73.14%	81.71%	87.21%	99.07%
2 kN→3 kN	29.00%	88.29%	84.00%	84.14%	86.00%	82.64%	98.93%
3 kN→0 kN	39.00%	70.71%	68.36%	69.50%	76.86%	75.50%	93.79%
3 kN→1 kN	38.07%	84.64%	84.79%	84.93%	84.21%	85.07%	99.93%
3 kN→2 kN	33.64%	91.21%	89.00%	92.14%	90.50%	87.79%	98.93%
mean value	29.69%	76.89%	77.87%	76.93%	83.51%	82.10%	96.18%
standard deviation	0.1093	0.1069	0.0988	0.1137	0.0607	0.0676	0.0365

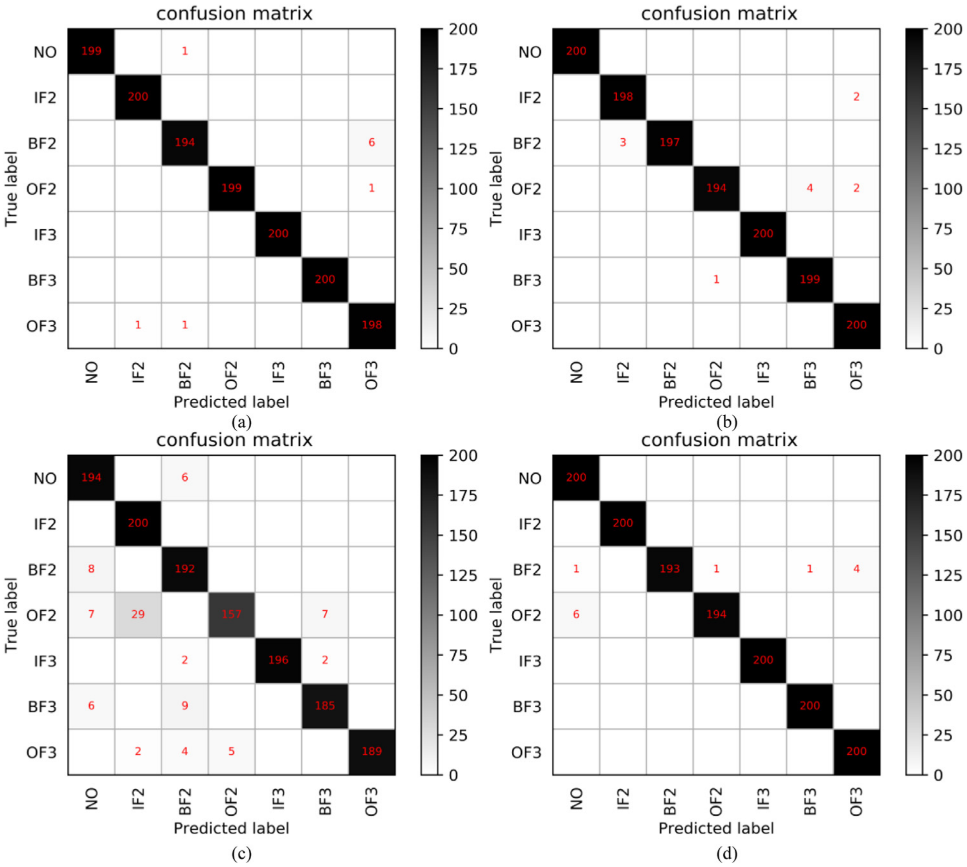
Table 7
F-score, training accuracy, and testing accuracy of MDIAN.

Tasks	F-score	training accuracy	testing accuracy
0 kN→1 kN	0.9272	100.00%	92.86%
0 kN→2 kN	0.8989	100.00%	90.07%
0 kN→3 kN	0.9055	100.00%	90.79%
1 kN→0 kN	0.9757	100.00%	97.57%
1 kN→2 kN	0.9928	100.00%	99.29%
1 kN→3 kN	0.9914	100.00%	99.14%
2 kN→0 kN	0.9373	100.00%	93.79%
2 kN→1 kN	0.9906	100.00%	99.07%
2 kN→3 kN	0.9892	100.00%	98.93%
3 kN→0 kN	0.9360	100.00%	93.79%
3 kN→1 kN	0.9992	100.00%	99.93%
3 kN→2 kN	0.9892	100.00%	98.93%

superior generalizability and robustness of the proposed model. Our model can deal with different working conditions and give the most accurate diagnostic results.

- The traditional machine learning method SVM ranks last among these models because it ignores the distribution difference and treats the source and target domains as domains subjected to the same distribution. Although CNN achieves better results than SVM, the CNN results are unsatisfactory because CNN also fails to narrow the distribution difference between the source and target domains.
- We combine CNN with the distance metric MMD and CMMD to illustrate the necessity of reducing the distribution difference between the source and target domains. The results of CNN + MMD and CNN + CMMD demonstrate that the effect of narrowing the distance metric is unclear. On the one hand, compared with CNN, the accuracy rates of CNN + MMD and CNN + CMMD improve slightly and CNN + MMD and CNN + CMMD even perform worse than CNN in very few transfer tasks. On the other hand, the results of CNN + MMD and CNN + CMMD are unstable, which means these two methods cannot deal appropriately with the complex and changeable working conditions of fault diagnosis because a three-layer CNN cannot extract useful and common features that belong to both the source and target domains.

MDDAN is constructed to demonstrate ResNet-50′s powerful feature extraction ability and the advantages of CMMD compared with MMD.



(a) (b) (c) (d)

Fig. 15. Confusion Matrix: (a) 1 kN → 2 kN, (b) 1 kN → 3 kN, (c) 2 kN → 0 kN, and (d) 2 kN → 1 kN.

MMDAN performs better than the first four methods because of ResNet-50′s significant extraction ability. At the same time, the discrepancy of marginal distributions is measured using MDDAN, and the source and target domains are drawn closer by minimizing MMD. Thus, MDDAN achieves better results. Unlike MDDAN, MDIAN measures the discrepancy of conditional distributions by minimizing CMMD instead of MMD. CMMD aims to bring closer together the samples from the source domain and samples with the same label from the target domain. Hence, MDIAN obtains the best results among these methods.

· The multiple scale feature extractor is crucial in MDIAN to confirm that the multiple scale feature extractor is removed from MDIAN, that is, DIAN. Table 3 shows that compared with MDIAN, the accuracy of DIAN decreases by approximately 10%. This decrease implies the extraction of high-level features at multiple scales can enhance the transferability of the model.

Based on the above comparative experiments, MDIAN combines the advantages of modified ResNet-50, CMMD, and multi-scale feature learner to ensure that it achieves the most satisfactory diagnosis results. MDIAN performs better than the traditional SVM and simple CNN models because of ResNet-50′s powerful feature learning ability. MDIAN using CMMD as the distance metric outperforms CNN + MMD and MDDAN, which use MMD as the distance metric. The multi-scale feature learner that can analyze low-level features further from different scales also contributes to the high-precision diagnosis results of MDIAN, whereas the diagnostic performance of DIAN without the multi-scale feature learner declines.

Fig. 7 illustrates that the normal state of bearing can be easily recognized but the roller failures are easily misjudged because of the irregular impulse caused by cracks on the rollers.

To illustrate the ability of the proposed model in aligning the same class of samples in the different domains, Fig. 8 shows that t-SNE [40] is utilized to show the distribution of differently distributed data in the transfer task 0 hp → 1 hp after model processing, that is high-level features. Take samples marked with label 8 for example, Fig. 9(a) shows a partial enlarged drawing of Figs. 8(e) and 9(b) enlarges part of Fig. 8(f).

Fig. 8(f) clearly demonstrates that the features divided by t-SNE have the smallest number of wrong clustering. This finding indicates that MDIAN discriminates the samples in the target domain more clearly than the other methods. CNN (Fig. 8(a)) shows the worst clustering results among all the methods. MDDAN (Fig. 8(d)) and MDIN (Fig. 8(e)) performs better than CNN + MMD (Fig. 8(b)) and CNN + CMMD (Fig. 8(c)) due to ResNet-50′s strong learning ability. By combining the advantages of ResNet-50, CMMD, and the multi-scale feature extractor, the samples with the same label from both the source and target domains could be brought closer together. Fig. 9 clearly shows that MDIAN is capable of clustering all samples with the same label from different domains. However, other methods may mix them with different labels, which will lead to misclassification.

Fig. 10 shows the visualization results of the original signal features (Fig. 10(a)) and the low-level features (Fig. 10(b)) extracted by MDIAN in task 0 hp → 1 hp. We can conclude that compared with the original signal features, the low-level features obtained by ResNet-50 begin to alleviate the feature overlapping of the samples with different labels. Compared with the high-level features (Fig. 8(f)), the proposed model obtains better clustering features as the level gets higher. Finally, the high-level features obtained by the multi-scale feature learner not only separate the samples of different labels but also gather the samples with the same label from different domains.

To further highlight the advantages of MDIAN, Fig. 11 shows the test results and errors during the tuning iterations in the transfer learning task 1 hp → 0 hp. Compared with the machine learning method CNN without any adaption, MDIAN achieves high accuracy and exhibits good convergence. Meanwhile, the error of MDIAN at almost 0

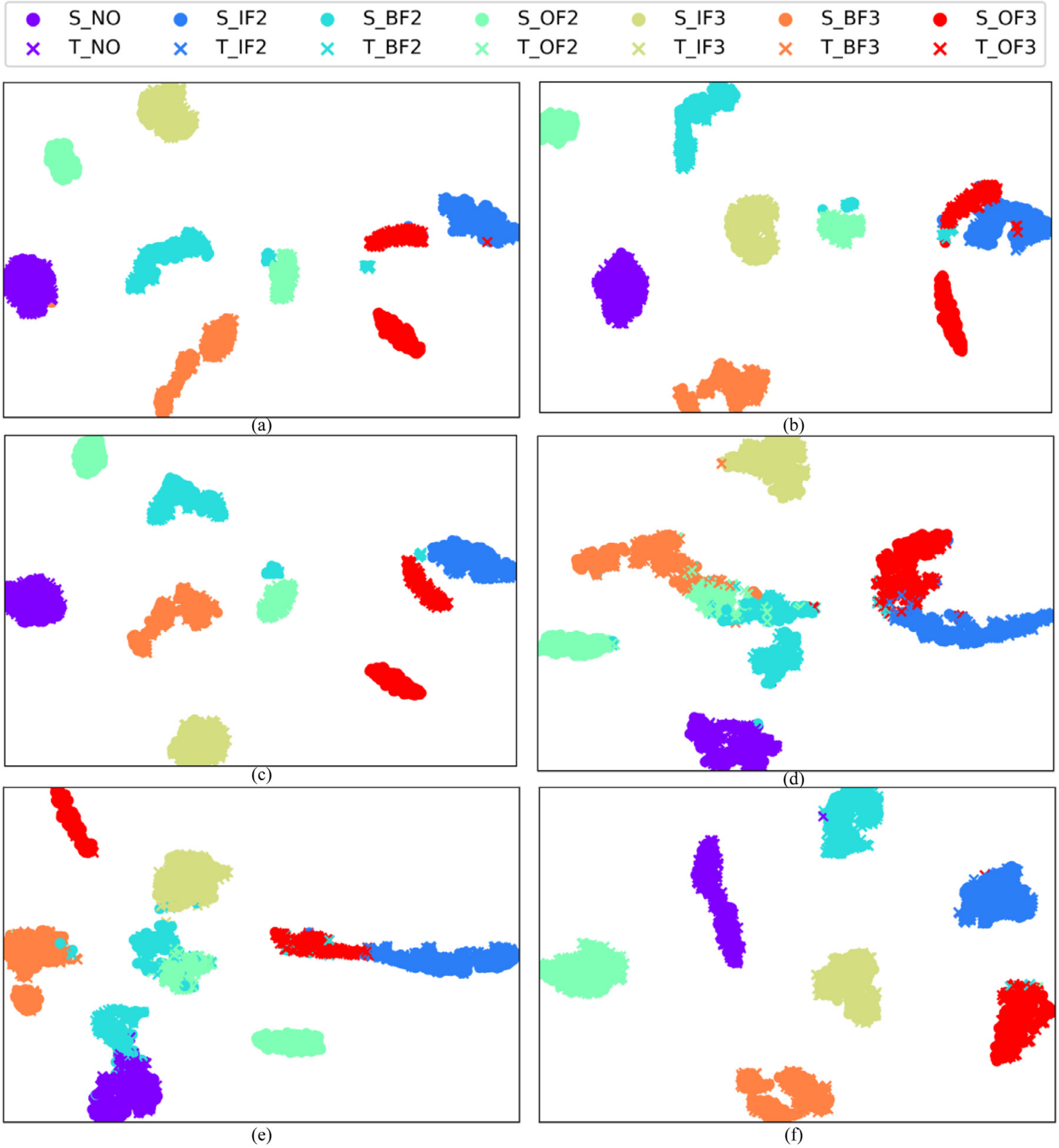


Fig. 16. Feature visualization through t-SNE from task 1 kN → 2 kN. (a) CNN, (b) CNN + MMD, (c) CNN + CMMD, (d) MDDAN, (e) DIAN, and (f) MDIAN.

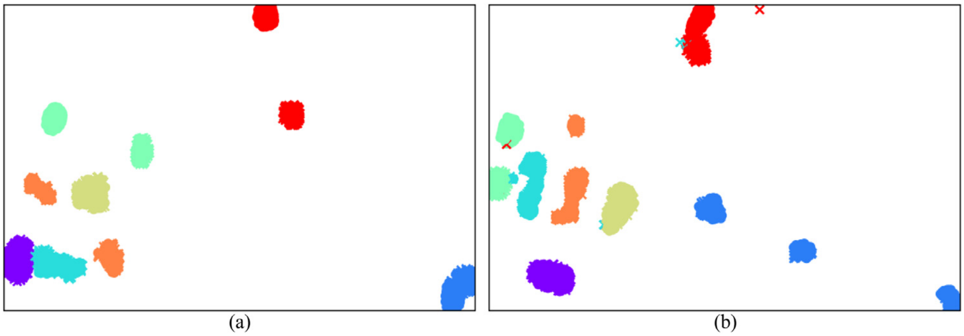


Fig. 17. Feature visualization through t-SNE from task1 kN → 2 kN. (a) original signal (b) low-level features.

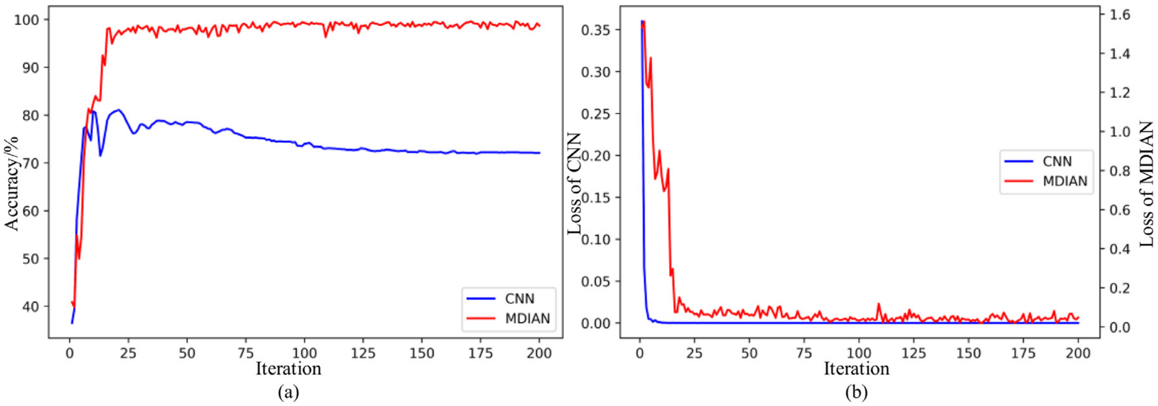


Fig. 18. Trends of (a) testing accuracy and (b) loss during the fine-tuning iterations in the transfer task 2 kN → 1 kN.

indicates that useful knowledge can be conveyed well from the source domain to the target domain by reducing the distance between both domains.

4.2. *Case II*

In this case, the bearing data came from a self-made test bench located in our laboratory (Fig. 12). Fig. 12 shows the test bench, which included the drive motor, the bolt and nut loading system, the standard bearing, the test bearing, the accelerometer, and the data acquisition system. The tested bearing model was 6205-2RS SKF and the defects with widths of 0.2 and 0.30 mm were set artificially on the inner ring, the outer ring, and the ball by spark-erosion wire cutting. During the sampling process, the accelerometer was placed on the test bearing pedestal at the 12 o'clock direction. The motor speed was 961 rpm, the sampling frequency was 10 kHz, and four health status signals, namely, the inner race fault, the outer race fault, the ball fault, and the normal condition, were collected under the working loads of 0, 1, 2, and 3 kN. Fig. 13 shows the pictures of the bearings with four different health states.

Similarly, four differently distributed data sets were named after their working loads (0, 1, 2, and 3 kN). Each data set contains 1400 samples marked with seven different labels and 200 samples with the same label are available. Table 5 summarizes the information on the seven health states. The spectrum of the seven samples with different labels is shown in Fig. 14.

Seven kinds of methods (SVM, CNN, CNN + MMD, CNN + CMMD, MDDAN, DIAN, and MDIAN) are also applied to 12 transfer tasks to verify the effectiveness of MDIAN further. Task A kN → B kN represents source domain A, which consists of 1400 samples labeled with 7 kinds of health states, and target domain B, which contains 1400 unlabeled samples. The results of these methods are presented in Table 6. The F-score, training accuracy, and testing accuracy of MDIAN are listed in Table 7. The confusion matrices of test accuracy based on MDIAN are shown in Fig. 15. Fig. 16 presents the cluster visualization using the t-SNE of certain methods in transfer task 1 kN → 2 kN. The visualization of the original signals and low-level features extracted by MDIAN in task 1 kN → 2 kN are presented in Fig. 17. The tendencies of testing accuracy and loss during the training steps in transfer task 2 kN → 1 kN are shown in Fig. 18.

In these experiments, the roller and outer ring faults are the most difficult to identify. The diagnosis performance of SVM is seriously degraded, whereas MDIAN exhibits superior diagnostic performance. The simplified versions of MDIAN, that is, CNN + CMMD and DIAN perform better than SVM but not as well as MDIAN. Both CNN + MMD and MDDAN use MMD as the distance metric, but the diagnostic accuracy of both is lower than that of MDIAN. These comparative experiments further confirm the superiority of MDIAN in this case.

5. Conclusions and future work

The safety and reliability of the rotating machinery rely strongly on some fundamental elements, such as bearings. In this study, a transfer learning model for bearing fault diagnosis called MDIAN is designed to overcome the problems of the source and target domain data obeying different distributions. This model aligns the conditional distributions of multiple scale high-level features extracted through a multiple scale feature extractor. Benefiting from the deep structure of MDIAN and the role of CMMD, MDIAN can address the domain shift problem and then extract discriminative and powerful features from different classes. Some new findings are summarized as follows: 1) ResNet-50 is modified to automatically learn low-level features without manually selecting features. 2) The distance between different domains is measured by the conditional maximum mean discrepancy, which can improve the prediction accuracy of fault diagnosis effectively under variable operating conditions. 3) The multi-scale feature extractor can be embedded easily in most fault diagnosis models to decrease information loss during feature extraction. The superiority and robustness of the designed network are confirmed through extensive experiments, including comparative experiments.

Our future work will focus on simplifying and optimizing the proposed model and its training algorithm to increase calculation efficiency. Multiple related source domains may also provide more beneficial knowledge to the target domain than only one source domain.

References

[1] Liu B, Liang Z, Parlikad AK, Xie M, Kuo W. Condition-based maintenance for systems with aging and cumulative damage based on proportional hazards model. Reliab Eng Syst Saf 2017;168:200–9.

[2] Feng Z, Chen X. Adaptive iterative generalized demodulation for nonstationary complex signal analysis: principle and application in rotating machinery fault diagnosis. Mech Syst Signal Process 2018;110:1–27.

[3] Kaplan K, Kaya Y, Kuncan M, Minaz MR, Ertunc HM. An improved feature extraction method using texture analysis with LBP for bearing fault diagnosis. Appl Soft Comput 2020;87:13.

[4] Islam MMM, Kim J-M. Reliable multiple combined fault diagnosis of bearings using heterogeneous feature models and multiclass support vector Machines. Reliab Eng Syst Saf 2019;184:55–66.

[5] Yan R, Gao RX, Chen X. Wavelets for fault diagnosis of rotary machines: a review with applications. Signal Process 2014;96:1–15.

[6] Qin Y. A new family of model-based impulsive wavelets and their sparse representation for rolling bearing fault diagnosis. IEEE Trans Ind Electron 2017;65:2716–26.

[7] He Q, Wu E, Pan Y. Vibration. Multi-scale stochastic resonance spectrogram for fault diagnosis of rolling element bearings. J Sound Vib 2018;420:174–84.

[8] Cocconcelli M, Zimroz R, Rubini R, Bartelmus W. STFT based approach for ball bearing fault detection in a varying speed motor. Condition monitoring of machinery in non-stationary operations. Springer; 2012. p. 41–50.

[9] Li N, Huang W, Guo W, Gao G, Zhu Z. Multiple enhanced sparse decomposition for gearbox compound fault diagnosis. IEEE Trans Instrum Meas 2019;69:770–81.

[10] Yu D, Cheng J, Yang Y. Application of EMD method and Hilbert spectrum to the fault diagnosis of roller bearings. Mech Syst Signal Process 2005;19:259–70.

[11] Jia F, Lei Y, Guo L, Lin J, Xing S. A neural network constructed by deep learning technique and its application to intelligent fault diagnosis of machines. Neurocomputing 2018;272:619–28.

[12] Kuncan M, Kaplan K, Mi Naz MR, Kaya Y, Ertunc HM. A novel feature extraction method for bearing fault classification with one dimensional ternary patterns. ISA Trans 2019;100:346–57.

[13] Wang J., Li S., An Z., Jiang X., Qian W., Ji S. Batch-normalized deep neural networks for achieving fast intelligent fault diagnosis of machines. arXiv preprint arXiv. 2019;329:53–65.

[14] Meng Z, Zhan X, Li J, Pan Z. An enhancement denoising autoencoder for rolling bearing fault diagnosis. Measurement 2018;130:448–54.

[15] Li K, Xiong M, Li F, Su L, Wu J. A novel fault diagnosis algorithm for rotating machinery based on a sparsity and neighborhood preserving deep extreme learning machine. Neurocomputing 2019;350:261–70.

[16] Liu R, Meng G, Yang B, Sun C, Chen X. Dislocated time series convolutional neural architecture: an intelligent fault diagnosis approach for electric machine. IEEE Trans Ind Inform 2016;13:1310–20.

[17] Zhu J, Chen N, Peng W. Estimation of bearing remaining useful life based on multiscale convolutional neural network. IEEE Trans Ind Electron 2018;66:3208–16.

[18] Xie J, Du G, Shen C, Chen N, Chen L, Zhu Z. An end-to-end model based on improved adaptive deep belief network and its application to bearing fault diagnosis. IEEE Access 2018;6:63584–96.

[19] Chen Z, Li W. Multisensor feature fusion for bearing fault diagnosis using sparse autoencoder and deep belief network. IEEE Trans Instrum Meas 2017;66:1693–702.

[20] Wang H, Li S, Song L, Cui L. A novel convolutional neural network based fault recognition method via image fusion of multi-vibration-signals. Comput Ind 2019;105:182–90.

[21] Pan SJ, Yang Q. A survey on transfer learning. IEEE Trans Knowl Data Eng 2009;22:1345–59.

[22] Dai W, Yang Q, Xue G-R, Yu Y. Boosting for transfer learning. Proceedings of the 24th international conference on machine learning. ACM; 2007. p. 193–200.

[23] Shen F, Chen C, Yan R, Gao RX. Bearing fault diagnosis based on SVD feature extraction and transfer learning classification. Proceedings of 2015 prognostics and system health management conference (PHM). IEEE; 2015. p. 1–6.

[24] Cao P, Zhang S, Tang J. Preprocessing-free gear fault diagnosis using small datasets with deep convolutional neural network-based transfer learning. IEEE Access 2018;6:26241–53.

[25] Guo L, Lei Y, Xing S, Yan T, Li N. Deep convolutional transfer learning network: a new method for intelligent fault diagnosis of machines with unlabeled data. IEEE Trans Ind Electron 2018;66:7316–25.

[26] Wen L, Gao L, Li X. A new deep transfer learning based on sparse auto-encoder for fault diagnosis. IEEE Trans Syst Man Cybern Syst 2017;49:136–44.

[27] Yang B, Lei Y, Jia F, Xing S. An intelligent fault diagnosis approach based on transfer learning from laboratory bearings to locomotive bearings. Mech Syst Signal Process 2019;122:692–706.

[28] Sun B, Saenko K. Deep coral: correlation alignment for deep domain adaptation. Proceedings of European conference on computer vision. Springer; 2016. p. 443–50.

[29] Tolstikhin I., Bousquet O., Gelly S., Schoelkopf B. Wasserstein auto-encoders. arXiv preprint arXiv. 2017.

[30] Pan SJ, Tsang IW, Kwok JT, Yang Q. Domain adaptation via transfer component analysis. IEEE Trans Neural Netw 2010;22:199–210.

[31] Zhu J, Chen N, Shen C. A new deep transfer learning method for bearing fault diagnosis under different working conditions. IEEE Sens J 2019. https://doi.org/10.1109/JSEN.2019.2936932.

[32] Udmale SS, Singh SK, Singh R, Sangaiah AK. Multi-fault bearing classification using sensors and ConvNet-based transfer learning approach. IEEE Sens J 2019:1433–44.

[33] Hinton G.E., Srivastava N., Krizhevsky A., Sutskever I., Salakhutdinov R.R. Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv. 2012.

[34] Lin M., Chen Q., Yan S. Network in network. arXiv preprint arXiv. 2013.

[35] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. p. 770–8.

[36] Dumoulin V., Visin F.A guide to convolution arithmetic for deep learning. arXiv preprint arXiv. 2016.

[37] Wang J, Chen Y, Hu L, Peng X, Philip SY. Stratified transfer learning for cross-domain activity recognitio. Proceedings of 2018 IEEE international conference on pervasive computing and communications (PerCom). IEEE; 2018. p. 1–10.

[38] Sun Q, Chattopadhyay R, Panchanathan S, Ye J. A two-stage weighting framework for multi-source domain adaptation. Adv Neural Inf Process Syst 2011:505–13.

[39] "Case Western Reserve University Bearing Data Center Website", https://csegroups.case.edu/bearingdatacenter/home. Accessed 52019.

[40] Donahue J, Jia Y, Vinyals O, Hoffman J, Zhang N, Tzeng E, et al. Decaf: a deep convolutional activation feature for generic visual recognition. Proceedings of International conference on machine learning. 2014. p. 647–55.