Measurement 187 (2022) 110332

Contents lists available at ScienceDirect

Measurement

journal homepage: www.elsevier.com/locate/measurement

Joint distribution adaptation with diverse feature aggregation: A new transfer learning framework for bearing diagnosis across different machines

Shiyao Jia [a], Yafei Deng [a], Jun Lv [b], Shichang Du [a,*], Zhiyuan Xie [a]

[a] School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai 200241, China
[b] Faculty of Economics and Management, Shanghai 200241, China

ARTICLE INFO

Keywords:
Fault diagnosis
Rolling element bearing
Deep transfer learning
Joint distribution adaptation

ABSTRACT

On account of lacking labeled samples for the bearing fault diagnosis in real engineering applications, transfer learning is widely investigated for transferring diagnosis information. A more challenging but realistic scenario called transfer across different machines (TDM) is investigated in this paper where previous approaches may degenerate greatly with more drastic domain shifts. A joint distribution adaptation-based transfer network with diverse feature aggregation (JDFA) is proposed, where the diverse feature aggregation module is added to enhance feature extraction capability across large domain gaps. Then the joint maximum mean discrepancy between source and target domain samples is adopted to reduce the distribution discrepancy automatically. Extensive TDM transfer learning experiments are conducted. The average accuracy reaches 99.178% that is much higher than state-of-the-art methods, demonstrating the proposed JDFA framework can effectively achieve superior diagnostic performance, and significantly promote fault diagnosis research under TDM scenario in view of applicability and practicability of algorithms.

1. Introduction

As an essential component of rotating machinery, the reliability of rolling element bearing has attracted extensive attention in both academic and industrial fields. Recently, deep learning has been well investigated and applied to the intelligent bearing fault diagnosis due to its superiority on learning hierarchical representations of complicated data [1]. The success of implementing reliable deep learning models is based on a large amount of labeled data, however, it is unpractical to obtain sufficient labeled data covering different working scenarios [2]. Firstly, machines usually operate in healthy states and rarely malfunction, which results in unbalanced datasets with abundant health data and insufficient failure data. Secondly, it is hard to obtain labeled data from machines on different working regimes considering the enormous cost of manual data tagging, thus results in most of the data collected in the engineering scenarios are unlabeled. Limited by the two aforementioned reasons, the deep learning-based diagnosis models may degenerate greatly in engineering scenarios. To overcome this limitation, transfer learning becomes a promising solution by reusing the acquired diagnostic information to other relevant diagnostic tasks [3]. For bearing diagnostic issues, the distribution inconsistencies across training and unlabeled testing data could be well addressed through reducing distribution discrepancies with common transferable feature mappings, which relaxes the label assumption and facilitates the diagnosis performance on different working scenarios.

For the research of rolling element bearings' fault diagnosis, the initial study is based on expert experience. The experience-based diagnosis method relies on subjective judgments, which are hard to achieve flexible fault diagnosis when the mechanical systems are complicated. In recent years, fault diagnosis has gradually developed into quantitative research. During this period, statistics-based research and data-driven research have attracted more attention [4–6]. By acquiring and analyzing vibration signal data of bearings, statistical or machine learning methods are used to establish bearing fault classification models. To a certain extent, this kind of research weakens the contribution of human labor in machine fault diagnosis [7]. However, the feature selection and model design are primarily dependent on manual operation and experience, which may be inappropriate when dealing with complex data. Furthermore, signal processing methods have poor generalization ability because it is difficult to select comprehensive and widely applicable features manually, and traditional machine learning methods are mainly based on shallow layers of feature space, making it

* Corresponding author.
E-mail addresses: jiashiyao@sjtu.edu.cn (S. Jia), phoenixdyf@sjtu.edu.cn (Y. Deng), jlv@dbm.ecnu.edu.cn (J. Lv), lovbin@sjtu.edu.cn (S. Du), zhiyuanxie@sjtu.edu.cn (Z. Xie).

https://doi.org/10.1016/j.measurement.2021.110332
Received 31 December 2020; Received in revised form 4 October 2021; Accepted 10 October 2021
Available online 27 October 2021

difficult to extract non-linear and non-stationary feature forms of vibration signals. As a large amount of data can be obtained from monitoring the state of machines, deep learning is gradually coming into the pictures. Deep learning is a new topic in the field of machine learning [8] and has also produced many achievements in the field of fault diagnosis [9–12]. In contrast to the above methods, deep learning can automatically extract fault features from the collected data instead of extracting features manually. Moreover, because of its multiple network structure, deep learning can learn multiple layers representations from input data through deep architectures with multi-layer data processing unit [13], to deeply recognize the semantic features of data, and then effectively overcome the limitations of traditional machine learning methods.

Compared with deep learning methods, transfer learning applies the knowledge learned from one or more tasks to other related but different domains [14]. It can overcome the weakness of lacking labels by reusing existing information to relevant domains based on already acquired datasets [1]. Related algorithms about transfer learning can trace back to 1995[15]. Since the 2010 s, some achievements have been yielded in the field of computer vision [16] and speech recognition [17], such as TrAdaboost [18], transfer component analysis (TCA) [19], joint distribution adaptation (JDA) [20], deep adaptation networks (DAN) [21], adversarial domain adaptation (ADA) [22], et al. In the field of fault diagnosis, some researchers have begun to develop some researches [23–25]. These approaches are expected to provide diagnostic models that can transfer the diagnostic knowledge learned from one or some diagnostic tasks to other related but different tasks. Current studies show that transfer learning can accelerate the convergence, reduce the training time, and improve the accuracy of the classifier in the study of fault diagnosis. Thus, transfer learning theories are expected to overcome the problem of insufficient labeled samples, through which diagnostic knowledge can be transferred from existing datasets to unlabeled bearing samples.

Transfer learning task includes two datasets, which are respectively from the source domain and the target domain. The data in the target domain have relevant knowledge but may belong to different distribution compared with those in the source domain. Transfer learning can realize the transfer of knowledge contained in the source domain to the target domain, and reduce the distribution discrepancy between the source and the target domain samples, thereby improving the performance of the predictive model for the target domain [3]. According to the different transfer scenarios, the current research can be roughly divided into two categories [7]: transfer in the same machine and transfer across different machines. Condition I: Transferring in the same machine means that the data is collected from the same machine, but data of the source and the target domains come from different operation conditions [26,27], such as different loads, different operating speeds, or different working environments, etc. Due to the different working conditions, the data distributions between the two domains tend to be different, which leads to the diagnosis models trained by source-domain data are unable to be used in the target domain directly. Condition II: Transferring across different machines (TDM) means that the data is collected from different but related machines. In this situation, the data from different machines may suffer diverse machine specifications, structures, measurement environments, working environments, etc.[7]. The distribution discrepancy will be more serious or even inconsistent, which is due to the difference in machine structures or processing methods. Therefore, for this type of distant domain transfer learning, models need to possess stronger generalization ability.

According to the survey[15], transfer learning methods can be categorized into four categories: instance-based, feature-based, parameter-based, and relation-based methods. In view of the above two types of scenarios, it has made some progress in feature-based transfer learning currently. Feature-based transfer learning is dedicated to finding common latent feature mapping through feature transformation i.e., mapping two domains into a sharing feature subspace and use them as a bridge to transfer knowledge [28]. Through converting original features into new feature representations, the cross-domain discrepancy will be reduced and knowledge transfer can be realized. Development of deep learning promotes the combination of transfer learning and deep learning, and the deep layer model of deep learning is used to automatically learn transferable features from cross-domain data, thereby further improving the approach. In the area of fault identification and diagnosis of bearings, some researches have appeared by using deep transfer learning. Wen et al. [29] combined deep learning and transfer learning to conduct fault diagnosis for bearings. The proposed model is based on a three-layer sparse auto-encoder (SAE), and MMD between source and target features is set as the error term which needs to be minimized. The method is validated on the dataset of Case Western Reserve University. Li et al. [27] applied a deep distance learning algorithm to solve bearing fault diagnosis in ambient noise and operating change states, where the MMD is adopted as the distribution discrepancy. Yang et al. [23] proposed a feature-based transfer neural network for bearing fault diagnosis, where multiple-layer MMD is minimized as a regularization term and the pseudo label term is attached at the same time, to realize the transfer learning from laboratory data set to a real-case dataset. Qian et al. [30] proposed a novel distribution discrepancy measuring algorithm called high-order Kullback-Leibler (*HKL*) divergence to construct a three-stage intelligent fault diagnosis model and verified it by experiments on a rolling bearing dataset and a gearbox dataset. Han et al. [31] established a fault diagnosis model by adopting joint distributed adaptation combined with a deep transfer network, which was verified on three fault datasets including bearings. Wang et al. [32] extracted the low-level features with the modified ResNet-50, analyzed the features with the multi-scale feature extractor, and took the conditional distribution distance between the source and target domains as the constraint condition to realize the intra-class adaptation. Finally, the model was verified in the bearing data sets of different working loads. Lei et al. [33] proposed a transferable method with adaptive manifold probability distribution to deal with bearing fault diagnosis under different working conditions, in which the geodesic flow kernel is utilized to align the cross-domain data distribution. Zhang et al. [34] designed an enhanced transfer joint matching (TJM) approach for cross-domain bearing defect diagnosis, combining the maximum variance discrepancy with the maximum mean discrepancy for the feature matching.

While a large number of approaches have been proposed for transferring diagnosis knowledge for bearing components, they still suffer the following problems which limit the extensions from the academic research to industrial applications:

1) Most of the studies only focus on the transfer in the identical machine (TIM) scenario, where the diagnosis knowledge is transferred across different working conditions. Actually, in real-world industrial applications, labeled data are difficult to be obtained from bearings running in complex mechanical systems compared with the laboratory bearings. Therefore, the transfer across different machines (TDM) is more essential and crucial. Only if the model under TDM scenario works well, the diagnosis knowledge could be transferred and extended to engineering applications in which labelled data is not available.
2) For the study of fault classification of bearing faults across different machines (TDM), the challenge is the larger, more intense, and grossly inconsistent distribution discrepancy, where previous approaches may degenerate greatly with more drastic domain shifts. Such a challenge places greater demands on the generalization ability and robustness of the model.
3) The previous studies most consider marginal distribution but neglect the effect of joint distribution when calculating distribution discrepancies, leading to the danger of distribution misalignment when encountering TDM scenario. Some studies have made exploratory work on investigating the joint distribution through merging the

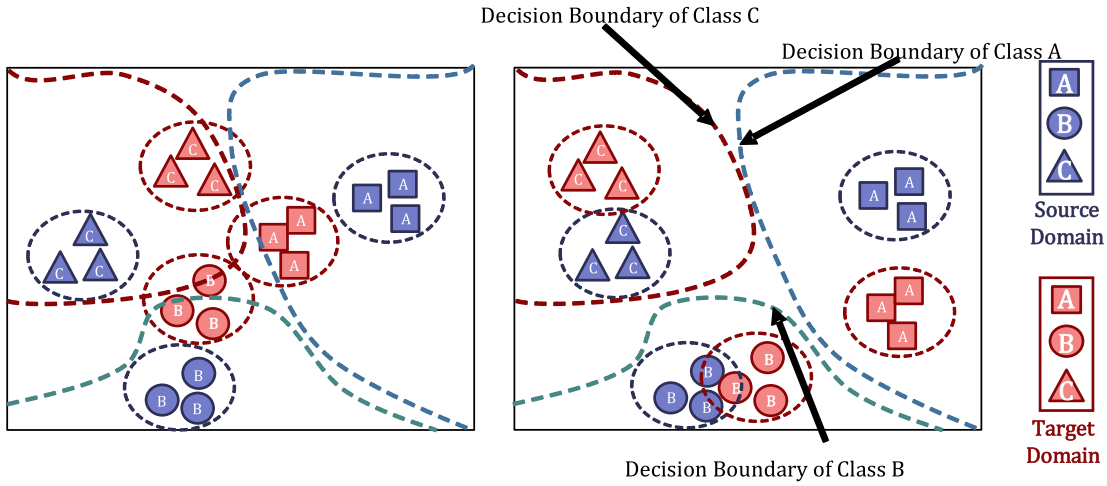

Fig. 1. Fault diagnosis by transfer learning: (a) without transfer learning, (b) with transfer learning.

marginal and conditional distributions together, but they may still lack generalization under different degrees of domain shift.

In order to solve the aforementioned pending problems and promoting the bearing diagnosis model under cross-machines transfer scenario, this paper proposes JDFA architecture: a joint distribution adaptation-based transfer network with diverse feature aggregation module for fault diagnosis of bearing across machines. In the proposed architecture, diverse feature aggregation (DFA) modules are added to the CNN backbone to extract richer hidden layer features, and joint distribution adaptation is adopted to reduce the distribution discrepancy between the source and target domains to effectively diagnose bearing faults across machines.

The contributions of the proposed method are summarized as follows.

1) A more challenging but more realistic scenario called as transfer across different machines (TDM) is set for bearing fault diagnosis, in which the training labeled data and testing unlabeled data are obtained from different machines. A novel transfer framework JDFA is proposed to tackle the cross-machine scenario bearing fault diagnosis issue and is expected to promote the practical applications on the intelligent bearing diagnosis for real industrial scenarios.
2) The diverse feature aggregation (DFA) module is designed and cooperated with the CNN structure, which effectively improves the capability to extract comprehensive features under different degrees of domain shifts.
3) The joint distribution joint maximum mean discrepancy (JMMD) is added into the constraints of model training to diminish the distribution discrepancy between source and target domains automatically.
4) A comprehensive case study of transferring diagnosis knowledge across different bearings is designed to evaluate the proposed method, and the comparative experimental results prove the superiority of the proposed method on promoting the diagnosis performance under TDM scenarios (significantly improving diagnosis accuracy to 99.178% on average which is much higher than compared state-of-the-art algorithms). Furthermore, the proposed work is beneficial to promote the applicability and practicability of research about bearing fault diagnosis.

The rest of this paper is organized as follows. In Section 2, the theoretical background is described. In Section 3, the proposed method, JDFA is presented in detail. Section 4 conducts the experimental diagnosis cases. The discussion and conclusion are drawn in Section 5 and Section 6, respectively.

2. Theoretical background

2.1. One-dimensional convolutional neural network

One-dimensional convolutional neural network (1-DCNN) is a CNN network, where the input data is one dimensional. 1-DCNN is consists of three parts, convolution layers for feature selection, pooling layers for down-sampling, and fully connected layers for classification. 1-DCNN achieves decent performance in learning high-level feature representations from dynamic signals and also achieves great results in reducing the computational complexity.

The convolutional layer is essentially a feature extractor with local connectivity. Multiple fixed-step kernel filters are used to convolve with the one-dimensional inputs, and feature maps are generated after the convolutional operation. The output of the j th neuron at the layer l is obtained as follow.

$$x_j^l = f\left(\sum_{i=1}^{N_{l-1}} conv\left(w_{ij}^{l-1}, x_i^{l-1}\right) + b_j^l\right) \tag{1}$$

where $conv(\cdot)$ is the 1D-convolution operation, w_{ij}^{l-1} is the weight of the i th neuron at the layer $l-1$, i.e., the parameter of corresponding kernel filter, b_j^l is the bias of the j th neuron at the layer l , and $f(\cdot)$ is an activation function, usually using rectified linear unit (ReLU).

The pooling layer is used to conduct down-sampling, which occurs after a convolutional layer. Max pooling and average pooling can reduce feature redundancy and prevent overfitting. The difference between max pooling and average pooling is that the max-pooling selects the maximum value in the region, while the average pooling chooses the mean value. Taking max-pooling as an example, the reduced-resolution feature map is expressed as follow.

$$y_j^l = max\left(x_i^l\right), i \in R_j^l \tag{2}$$

where R_j^l denotes the feature index of the j th pooling region in the l th layer and x_i^l represents the i th element of the pooling region.

The fully connected layer acts as a classifier in the convolutional neural network, which is similar to the hidden and output layers of a standard multi-layer perceptron (MLP). The convolutional layer, pooling layer, and activation function layer map the original data to the hidden layer feature space, while the fully connected layer plays the role of mapping the learned “distributed feature representation” to the sample label space.

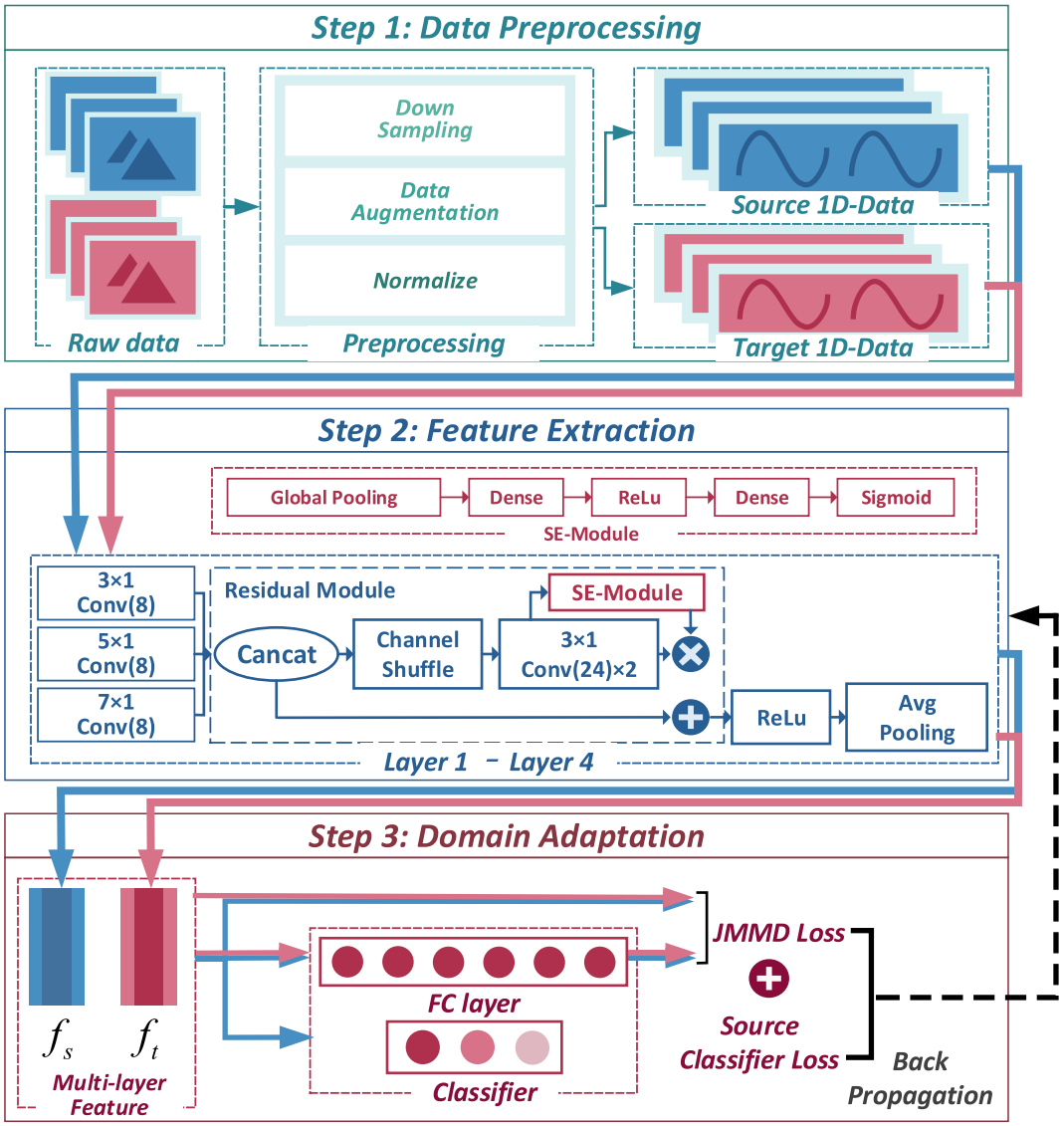

Fig. 2. The overall framework of JDFA.

2.2. A brief introduction to domain adaptation

In the domain adaptation process, a source domain $\mathscr{D}^s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$ with n_s labeled samples is given, which is devoted as the one that can provide some diagnosis knowledge to other task domain, and a target domain $\mathscr{D}^t = \{(x_i^t)\}_{i=1}^{n_t}$ with n_t unlabeled samples is provided, which is served as a task domain that needs to be transferred due to lack of diagnostic information and is expected to be correctly classified by using the diagnosis knowledge drawn from the source domain. The data samples from the source domain and the target domain are subject to the marginal probability distribution p and q, respectively. Besides, in order to provide enough diagnosis knowledge for the target domain, the label space of the source domain is expected to cover that of the target domain, i.e., $Y^t \subseteqq Y^s \subseteqq Y$, where Y^s and Y^t are label spaces in the source and target domains, respectively.

The samples in the source and the target domain are collected from different types of vibration data, leading to the serious distribution discrepancy of these data. Thus the learned features will also be subject to the distribution discrepancy if trained under supervised mode. As shown in Fig. 1(a), when the classifier $h(\cdot)$ is only trained by samples from the source domain, then using the classifier $h(\cdot)$ to classify the target domain sample often leads to misclassification because of the serious distribution discrepancy between the features learned from the source and the target domain. Therefore, the key point of the domain-adaptive method is to reduce the cross-domain discrepancy by extracting transferable features. As shown in Fig. 1(b), when the classifier $h(\cdot)$ can minimize the risk of $\mathrm{E}\left[h(x_i^t) \neq y_i^t\right]$ by learning transferable features with similar distribution, the domain sharing classifier can correctly identify the target domain samples by the knowledge provided by the source domain.

2.3. Maximum mean discrepancy

The main challenge of transfer learning is that there is no or only limited labeled sample information in the target domain. To resolve the problem, some methods are aiming to combine the source error with the discrepancy metric between the source domain and target domain to bound distribution discrepancy and to reduce the classification error of target domain. The maximum mean discrepancy (MMD) is a commonly used distance metric to measure the discrepancy between two domains.

Define $\mathscr{H}_k$ as the reproducing kernel Hilbert space (RKHS) with a characteristic kernel k and suppose the samples from the source and the target domain are subject to the marginal probability distribution p and q. The kernel mean embedding of p in $\mathscr{H}_k$ can be defined as $\mathrm{E}_p[\phi(x)]$,

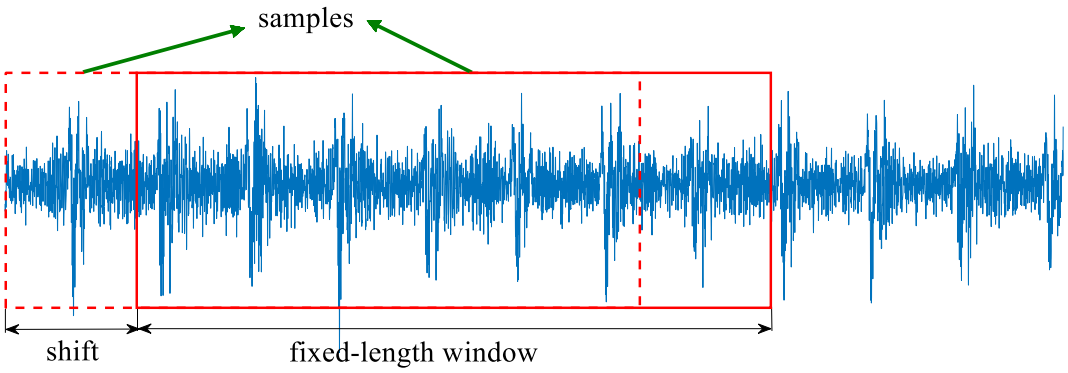

Fig. 3. Data augmentation.

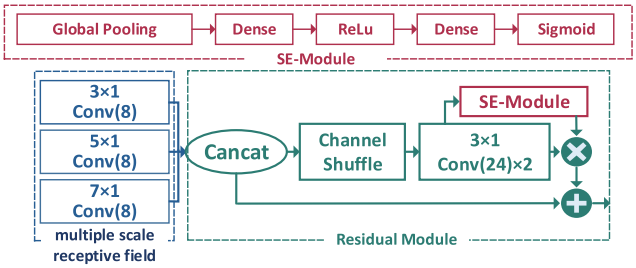

Fig. 4. The framework of the DFA module.

such that $\langle \mathrm{E}_p[\phi(\boldsymbol{x})], f\rangle_{\mathscr{H}_k} \triangleq \mathbb{E}_{x\sim p}(f(\boldsymbol{x})), \forall f \in \mathscr{H}_k$, in which ϕ is feature mapping. The MMD $d_k(p,q)$ between probability distribution p and q is defined as the RKHS distance between the kernel embedding of p and q , such that [21]

$$d_k^2(p,q) \triangleq \|\mathrm{E}_p[\phi(x^s)] - \mathrm{E}_q[\phi(x^t)]\|_{\mathscr{H}_k}^2 \tag{3}$$

And based on the kernel two-sample test theoretical result [35], $p = q$ if and only if $d_k^2(p,q) = 0$, the MMD can be unbiasedly estimated using a small batch of samples to overcome the difficulty of acquiring the true distribution, so that the unbiased estimator $\widehat{d}_k^2(p,q)$ is defined as

$$\begin{aligned}\widehat{d}_k^2(p,q) &= \frac{1}{n_s^2}\sum_{i=1}^{n_s}\sum_{j=1}^{n_s} k\left(\mathrm{x}_i^s, \mathrm{x}_j^s\right) \\ &+\frac{1}{n_t^2}\sum_{i=1}^{n_t}\sum_{j=1}^{n_t} k\left(\mathrm{x}_i^t, \mathrm{x}_j^t\right) \\ &-\frac{2}{n_s n_t}\sum_{i=1}^{n_s}\sum_{j=1}^{n_t} k\left(\mathrm{x}_i^s, \mathrm{x}_j^t\right)\end{aligned} \tag{4}$$

3. Proposed method

3.1. Overview of the proposed method

The framework of our proposed JDFA method is shown in Fig. 2, which consists of three parts: **data preprocessing, feature extraction, domain adaptation**. JDFA framework receives raw data from two domains: the source domain contains 1-D data set with labeled samples which can provide the diagnosis knowledge, and the target domain of which samples are unlabeled and need to be identified the health states. In the first step, down-sampling, data augmentation, and normalization are conducted to preprocess the original 1-D data from the source and the target domain. Then the feature extraction, i.e., diverse feature aggregation (DFA) extractor is designed in the second step to extract transferable features from samples of source and target domains. Noted that the samples from the source and the target domains are simultaneously handled by the same feature extractor. The diverse feature aggregation extractor is based on the 1D-CNN backbone with four hidden layers and consists of four optimization modules, including multiple-scale receptive field, Squeeze-and-Excitation Module (SE-Module) [36], residual module [37], and channel shuffle[38] operation. As for domain adaptation, joint maximum mean discrepancy (JMMD) is used to measure the distribution discrepancy of the learned transferable features. Then JMMD is added as part of the optimization objective to promote backward propagation in the training phase. By minimizing the distribution discrepancy of transferable features, features with small cross-domain discrepancy are obtained to classify the unlabeled samples in the target domain. Since the distribution of learned features from target domain can be similar to that in source domain samples by domain adaptation, the unlabeled samples in the target domain can be correctly classified by the domain-shared classifier.

3.2. JDFA architecture

In this section, the components of JDFA architecture including data preprocessing, DFA extractor, and domain adaptation with JMMD are demonstrated in detail. What's more, various optimization modules in DFA extractor are also presented in this part to explain the effect on final fault diagnosis.

3.2.1. Data augmentation

When addressing issues such as fault diagnosis and health management, it is often difficult to obtain large-scale data due to difficulties in data collection. If dealing with fault diagnosis problems using limited samples, CNNs tend to suffer overfitting which is the disadvantage for fault diagnosis in the TDM scenario. In the field of computer vision, data augmentation has been commonly used to increase the number of training samples to improve the generalization ability of models, including horizontal flipping, random crops/scales, and color jittering [39,40]. To overcome this situation in the fault diagnosis field, the data augmentation technique on the smaller datasets is proposed to avoid overfitting and improve the generalization performance when data is in a cyclical stabilization process.

A random generator is designed to perform fixed-length window slicing in a given one-dimensional vibration signal data, which is similar to window sliding on the time axis, as shown in Fig. 3. Assuming that the length of the given sample is n , and the length of the slice is s , the random generator can generate up to $n-s+1$ samples after slicing when $n > s$.

3.2.2. Framework of diverse feature aggregation (DFA) extractor

The backbone of the DFA extractor is one-dimensional domain-shared CNN with four hidden layers, which consists of convolutional layers, pooling layers, and full connection layers. The network parameters for source and target domains are shared. The input of the overall

network is one-dimensional vibration data with fixed length while the output of the overall network is a probability distribution over the classes in the dataset. The overall network includes four hidden layers with the Rectified Linear Unit (ReLU) as activation function, which is a commonly used activation function to solve the problem of gradient disappearance.

The diverse feature aggregation (DFA) modules are added to the backbone to improve the stability of the overall framework and the accuracy of final classification, including multiple-scale receptive field, SE-Module, residual module, and channel shuffle operation. The framework of the DFA module is shown in Fig. 4. The specific parameter setting of the overall network is shown in Appendix (i.e., Table10).

3.2.3. Multiple-scale receptive field

The term receptive field comes from the field of biological vision and refers to the area in which neurons respond to stimuli. In CNN's, the receptive field is defined as the size of pixels on the feature map corresponding to the area of the original image, that is, the area of the affected input space in the feature.

The receptive field in CNN architecture is related to the size of the convolution kernel. A larger convolution kernel means a larger receptive field and neural network is more inclined to obtain the global information of input signal. On the contrary, a smaller convolution kernel, i.e., a smaller receptive field represents that more local details tend to be obtained by the neural network. Generally speaking, one-dimensional vibration data usually contains information with diverse feature scales. Due to the uncertainty of feature scales of input vibration signal and feature maps generated from deeper layers, using convolution kernel with single scale cannot guarantee either global feature coverage or effective extraction for local details, especially when dealing with complex and variable signal data. Considering inappropriate feature extraction will lead to degradation in final fault classification, multi-scale convolutional kernels representing multiple receptive fields are used for feature extraction when constructing the model, which allows the network to obtain features at multiple scales. The process can be formalized as follows:

$$conv_i = concat\left(conv_{i_1(3\times1)}(input), conv_{i_2(5\times1)}(input), conv_{i_3(7\times1)}(input)\right)$$

$$(3\times1),\ (5\times1)\ and\ (7\times1)$$

3.2.4. Squeeze-and-Excitation module (SE-Module)

The number of feature channels will increase with hidden layers becoming deeper especially when multiple-scale receptive field mechanism is attached. More channels will contain richer features combination. However, features in different channels will have diverse importance, i.e., diverse degrees of contribution to final fault diagnosis. Therefore, different weights should be assigned to each feature channel instead of adopting uniform channel weights.

The SE-Module [36] possesses the ability to learn the weights of individual channels during the training phase. SE-Module contains two branches including one branch for SE operation and another branch for transmitting the original signal. SE operation contains a series of computations such as 1D-global pooling (transforming the features into the form of $1\times C$, C is the number of feature channels), fully connection, and non-linear activation functions to output weights of channels. After SE operation, input features are transformed into the weight vector of $1\times C$ and the weighted features can be represented as follows:

$$\mathrm{F}_{scale(\mathbf{u}_c,s_c)} = s_c\cdot\mathbf{u}_c \tag{6}$$

where s_c represents the original feature map and $\mathbf{u}_c$ represents the learned channel weight vector.

3.2.5. Residual learning

Outputs obtained from different hidden layers represent features of different levels. For example, outputs from shallow layers usually represent local features or detailed features while deep hidden layers tend to extract semantic features or abstract features. Shallow features and semantic features will both have an influence on final fault recognition from the local perspective and global perspective respectively. However, features obtained from shallow layers may be submerged among all features, and the impact of shallow features will disappear gradually. On the contrary, semantic features from deep layers will occupy more importance in the following classification process. This phenomenon may have a negative effect on following fault classification. Moreover, the overall network has the probability to experience network degeneration and gradient vanishment or gradient explosion with the number of network layers increasing.

To overcome the problems mentioned above, the residual module [37] is used in the DFA extractor. By adding the residual network structure, features from shallow layers will flow to deeper layers directly and be fused with semantic features simply. Thereby, hierarchical features can be preserved and corresponding problems can be avoided to some degree. The output of the residual module can be represented as follow,

$$y = conv + \mathrm{F}_{scale(\mathbf{u}_c,s_c)} \tag{7}$$

where *conv* represents the features from shallow layers after multiple-scale receptive field, $\mathrm{F}_{scale(\mathbf{u}_c,s_c)}$ represents the output undergoing the SE-Module.

3.2.6. Channel shuffle

In the TDM scenario, the diagnosis model needs to possess enough generalization ability to tackle challenges caused by the greatly varying distribution of data. Except for optimization operations presented above, channel shuffle operation is adopted in the JDFA framework to make the overall network stable and have sufficient generalization ability. In detail, feature channels obtained from multiple-scale receptive field module are divided into three groups equally. Then the feature matrix is reshaped, transposed, and reshaped again to conduct channel shuffle operation. With the order of origin feature channel disturbed in training phase, the JDFA framework can achieve more stable performance in the TDM scenario.

3.2.7. Joint distribution adaptation

Recent deep transfer learning mines more transferable features of deep networks through domain adaptation to matches the marginal distributions across domains. The MMD between the source and target domains has always been focused on many transfer learning tasks. However, the limitation of the MMD is that it only focuses on marginal distributions across domains, assuming the conditional distribution of the two domains are approximately equal, that is $P_s(y_s|\mathbf{x}_s)\approx P_t(y_t|\mathbf{x}_t)$. In practice, as the depth of the network increases, the joint distribution between the input features and the output labels causes more changes, subsequently leading to the greater cross-domain discrepancy and less transferability of features. Therefore, the joint distributions instead of marginal distributions of the source and target domains are used to narrow the discrepancy between them, so as to achieve greatly improved generalization performance and robustness of models. Instead of considering marginal distributions and conditional distributions separately or fusing simply, JMMD is adopted in JDFA architecture for domain adaptation.

The core of JMMD is the joint embedding of two or more variables, which is extended by kernel embeddings[41]. The joint embeddings can be viewed as the cross-covariance operator $C_{\mathbf{X}^{1:m}}$ by the standard equivalence between tensor and linear map [41]. $C_{\mathbf{X}^{1:m}}$ is computed as

$$C_{\mathbf{X}^{1:m}}(P)\triangleq\mathbb{E}_{\mathbf{X}^{1:m}}\left[\otimes_{\ell=1}^{m}\phi^{\ell}\left(\mathbf{X}^{\ell}\right)\right]=\int_{\times_{\ell=1}^{m}\Omega^{\ell}}\left(\otimes_{\ell=1}^{m}\phi^{\ell}\left(\mathrm{x}^{\ell}\right)\right)\mathrm{d}P\left(\mathrm{x}^{1},\cdots,\mathrm{x}^{m}\right) \tag{8}$$

in which $\mathbf{X}^{1:m}$ is a set of variables $\{\mathbf{X}^1,\cdots,\mathbf{X}^m\}$ on the domain $\times_{\ell=1}^{m}\Omega^{\ell}=\Omega^1\times\cdots\times\Omega^m$, ϕ^{ℓ} is the feature map endowed with kernel k^{ℓ}

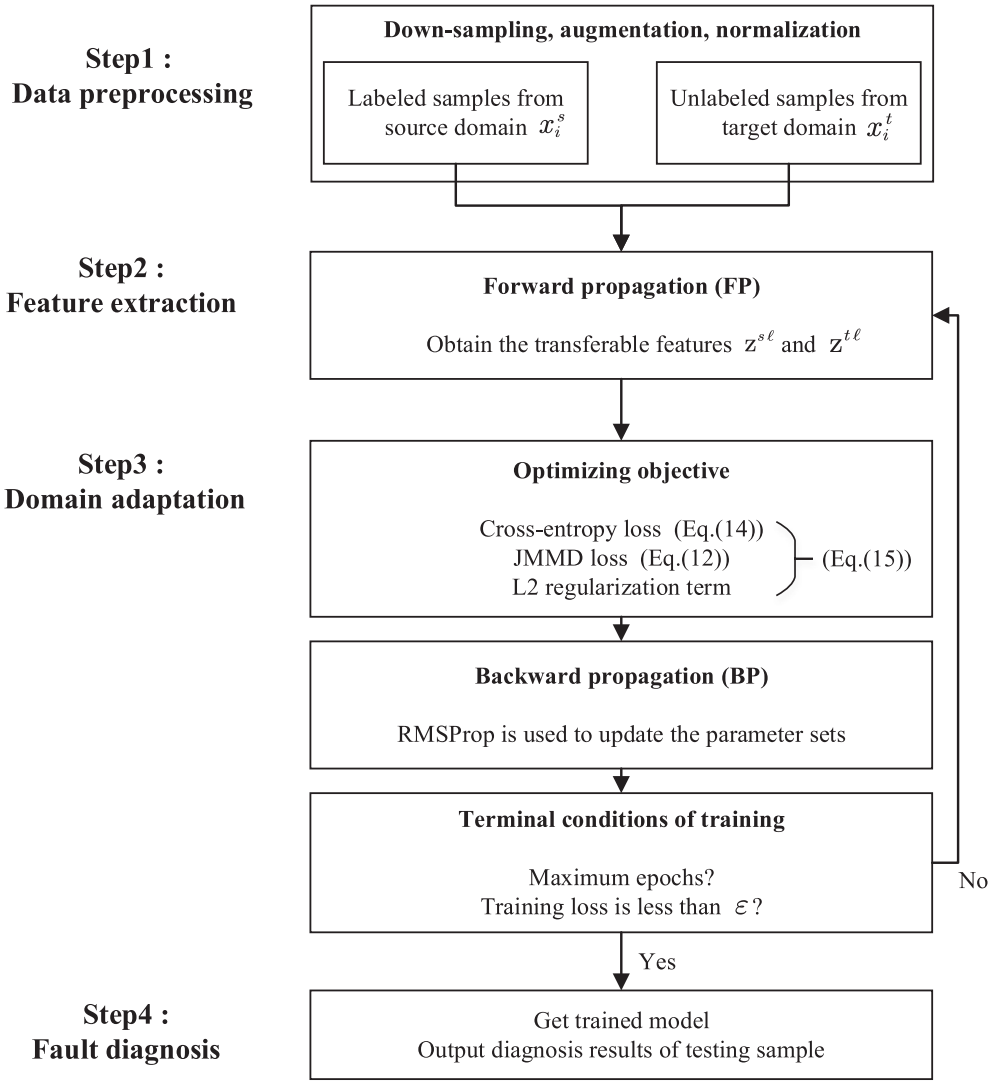

Fig. 5. Flowchart of the training process.

in reproducing kernel Hilbert space (RKHS) $\mathscr{H}^{\ell}$ for variable $\mathbf{X}^{\ell}$, $\otimes_{\ell=1}^{m}\phi^{\ell}(\mathbf{x}^{\ell})=\phi^{1}(\mathbf{x}^{1})\otimes\cdots\otimes\phi^{m}(\mathbf{x}^{m})$, where the inner product satisfies $\left\langle\otimes_{\ell=1}^{m}\phi^{\ell}(\mathbf{x}^{\ell}),\otimes_{\ell=1}^{m}\phi^{\ell}(\mathbf{x}'^{\ell})\right\rangle=\prod_{\ell=1}^{m}k^{\ell}(\mathbf{x}^{\ell},\mathbf{x}'^{\ell})$. When given a set of functions $f^{1},\cdots,f^{m}$, the joint embedding can be computed as

$$\mathbb{E}_{\mathbf{X}^{1:m}}\left[\prod_{\ell=1}^{m}\mathrm{f}^{\ell}(\mathbf{X}^{\ell})\right]=\left\langle\otimes_{\ell=1}^{m}\mathrm{f}^{\ell},\mathrm{C}_{\mathbf{X}^{1:m}}\right\rangle \tag{9}$$

As with the unbiased estimation of kernel embedding, the finite sample can be used to estimate joint embedding when the true distribution is unknown [42]. The empirical joint embedding can be estimated as

$$\widehat{C}_{\mathbf{X}^{1:m}}=\frac{1}{n}\sum_{i=1}^{n}\otimes_{\ell=1}^{m}\phi^{\ell}\left(\mathrm{x}_{i}^{\ell}\right) \tag{10}$$

Due to the shifts in the joint distribution present in the activation of higher network levels [1], joint distributions of the activations in the fully connected layers $\mathscr{L}=\{\mathrm{fc1},\mathrm{fc2}\}$, i.e. $P(\mathbf{Z}^{s1},\ldots,\mathbf{Z}^{s|\mathscr{L}|})$ and $Q(\mathbf{Z}^{t1},\ldots,\mathbf{Z}^{t|\mathscr{L}|})$, can be used to surrogate the original joint distribution $P(\mathbf{X}^{s},\mathbf{Y}^{s})$ and $Q(\mathbf{X}^{t},\mathbf{Y}^{t})$, respectively.

By using the advantage of MMD, the Hilbert space kernel embedding of the joint distribution can be used to measure the discrepancy of two joint distributions of the activations in layers $\mathscr{L}$. As a result, JMMD can be obtained, which is defined as

$$D_{\mathscr{L}}(P,Q)\triangleq\left\|C_{\mathbf{Z}^{s,1:|\mathscr{L}|}}(P)-C_{\mathbf{Z}^{t,1:|\mathscr{L}|}}(Q)\right\|^{2}_{\otimes_{\ell=1}^{|\mathscr{L}|}\mathscr{H}^{\ell}} \tag{11}$$

With the linear unbiased estimation of JMMD, JMMD can be estimated with a small batch of samples. Based on the virtue of the kernel two-sample test theory [35], the $P(\mathbf{Z}^{s1},\ldots,\mathbf{Z}^{s|\mathscr{L}|})=Q(\mathbf{Z}^{t1},\ldots,\mathbf{Z}^{t|\mathscr{L}|})$ if an only if $D_{\mathscr{L}}(P,Q)=0$. The empirical estimate of $D_{\mathscr{L}}(P,Q)$ is computed as the squared distance between the empirical kernel mean embeddings as

$$\begin{aligned}\widehat{D}_{\mathscr{L}}(P,Q)&=\frac{1}{n_s^2}\sum_{i=1}^{n_s}\sum_{j=1}^{n_s}\prod_{\ell\in\mathscr{L}}k^{\ell}\left(\mathrm{z}_i^{s\ell},\mathrm{z}_j^{s\ell}\right)\\&+\frac{1}{n_t^2}\sum_{i=1}^{n_t}\sum_{j=1}^{n_t}\prod_{\ell\in\mathscr{L}}k^{\ell}\left(\mathrm{z}_i^{t\ell},\mathrm{z}_j^{t\ell}\right)\\&-\frac{2}{n_s n_t}\sum_{i=1}^{n_s}\sum_{j=1}^{n_t}\prod_{\ell\in\mathscr{L}}k^{\ell}\left(\mathrm{z}_i^{s\ell},\mathrm{z}_j^{t\ell}\right)\end{aligned} \tag{12}$$

in which n_s is the number of the labeled points from the source domain, n_t is the number of unlabeled points from the target domain, $\mathrm{z}^{s\ell}$ and $\mathrm{z}^{t\ell}$ are activations in the layer ℓ from source and target domains, respectively.

It is noted that JMMD uses the product of kernels in each layer to

Table 1
Training and testing process of the proposed method.

Input: Source domain samples $\{x_i^s, y_i^s\}_{i=1}^{n_s}$ and target domain samples $\{x_i^t\}_{i=1}^{n_t}$ after data preprocessing, the batch size M, the tradeoff parameter λ, α, the learning rate η, the iteration number N	
Output: Diagnostic results $\{y_i^t\}_{i=1}^{n_t}$	
1.	Randomly initialize trainable parameters θ in the network.
2.	While the parameters θ do not converge or the training epoch n does not reach N do
3.	Select a batch of samples from the source domain and target domain respectively
4.	Extract hierarchical features z^s and z^t through forward-propagation
5.	Classify inputting samples $\{x_i^s\}_{i=1}^{M}$ based on extracted features
6.	Calculate loss function $L\leftarrow$ Cross-entropy + JMMD loss + L2 regularization term
7.	Calculate gradient g_θ base on loss value L
8.	Update parameters $\theta\leftarrow\theta+\eta\cdot g_\theta$ using RMSProp optimizer in backward-propagation
9.	Add training epoch $n\leftarrow n+1$
10.	If n reaches N and parameters θ do not converge
11.	Reset training epoch n and go to step 2
12.	Else if n reaches N
13.	Return network parameters θ
14.	Conduct testing process with target domain samples $\{x_i^t\}_{i=1}^{n_t}$
15.	Return the diagnostic results $\{y_i^t\}_{i=1}^{n_t}$

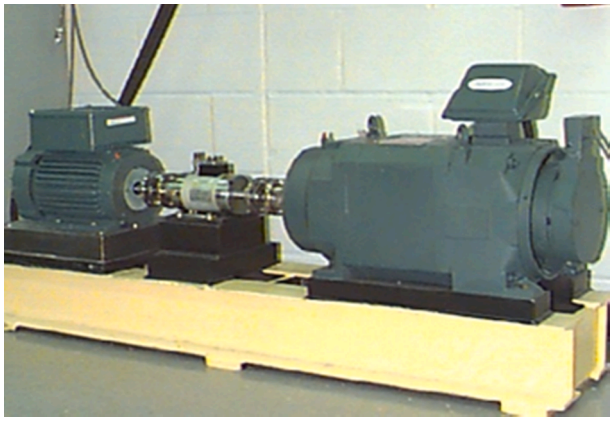

Fig. 6. The test rig of the CWRU bearing dataset, consisting of a 2 HP motor, a torque sensor, a power meter, and a motor control system [45].

Table 2
Introduction to datasets.

Datasets	Data sources	Operation Conditions	Damage Method
CWRU-DE-1	CWRU(DE)	0HP(1797 r/min)	EDM
CWRU-DE-2	CWRU(DE)	3HP(1730 r/min)	EDM
CWRU-FE-1	CWRU(FE)	0HP(1797 r/min)	EDM
CWRU-FE-2	CWRU(FE)	3HP(1730 r/min)	EDM
PDB-EDM	Paderborn	1500 rpm	EDM
PDB-EG	Paderborn	1500 rpm	manual electric engraver
PDB-ALT	Paderborn	1500 rpm	accelerated lifetime test

express the interaction of different variables in the joint distributions. And each kernel $k(z^s, z^t) = \langle \phi(z^s), \phi(z^t) \rangle$ can be defined as the combination of m kernels $\{k_u\}$ [21],

$$\mathscr{K} \triangleq \left\{ k = \sum_{u=1}^{m} \beta_u k_u : \sum_{u=1}^{m} \beta_u = 1, \beta_u \hat{a}\copyright ¾ 0, \forall u \right\} \tag{13}$$

where the β_u is the constraints on coefficients, which can be used to guarantee the generated multiple kernels k is characteristic. The multi-kernel k can use different kernels to enhance the MK-MMD test, which gives a principled method for optimal kernel selection [21].

3.3. Training process

After feature extraction from the source domain and target domain, the JMMD is supposed to be added to the loss function of CNN for backward adjusting. Suppose the empirical risk of CNN on the source domain is

$$L_E = \min_{\Theta} \frac{1}{n_s} \sum_{i=1}^{n_s} J\left(\theta\left(x_i^s\right), y_i^s\right) \tag{14}$$

where J is the cross-entropy loss function, $\theta(x_i^s)$ is CNN classifier that assigns x_i^s to label y_i^s, $\Theta = \left\{\mathbf{w}^{\checkmark}, \mathbf{b}^{\checkmark}\right\}_{\checkmark=1}^{l}$ is the set of all CNN parameters. Since the parameters cannot be directly transferred to the target domain, and the distribution of the source and the target domain are required to become similar under the hidden representations of fully connected layers, the JMMD based discrimination error L_D is added to the CNN risk to create a new optimization goal L. The optimizing objective L is given by

$$\begin{aligned} L &= L_E + L_D + L_R \\ &= \min_{\Theta} \frac{1}{n_s} \sum\nolimits_{i=1}^{n_s} J\left(\theta\left(x_i^s\right), y_i^s\right) + \lambda \widehat{D}_{\mathscr{L}}(P, Q) + \alpha \cdot \sum\nolimits_{j=1}^{N_w} w_j^2 \end{aligned} \tag{15}$$

where $\lambda > 0$ is the penalty parameter, and $\widehat{D}_{\mathscr{L}}(P, Q)$ is the JMMD between the source and the target domain on the special layers $\mathscr{L}$ as shown in Eq. (12). We set $\mathscr{L} = \{fc1, fc2\}$ for the proposed model. α is the trade-off parameter, $\sum_{j=1}^{N_w} w_j^2$ is the L2 regularization term [43] of CNN which is attached to enhance the generalization ability of the network, N_w is the total number of CNN weights, w is the weight of CNN.

The Eq. (15) is used as a loss function for the model and the loss is propagated back from the output layer to the hidden layer until it propagates to the input layer. The values of parameters are adjusted during backpropagation until the overall network achieves convergence. And the optimization algorithm RMSProp [44] is used to update the parameters.

The specific training process is presented in Fig. 5. In the step of data preprocessing, the datasets are down-sampled, augmented, and normalized in order from the source and target samples. In the step of feature extraction, the deep transferable features of source and target data are extracted through forward propagation, respectively. In the step of domain adaptation, the JMMD between source and target transferable features is calculated by Eq. (12). And then the optimizing objectives are calculated by Eq. (15) to enforce constraints on CNN parameters. RMSProp is used to update the parameter sets. When the training phase is over, the model can output the diagnosis results to classify the samples in the target domain.

3.4. Overall experimental implementation

In this subsection, in order to detail the overall implementation of the proposed method, the training and testing process of JDFA is illustrated in Table 1.

4. Case study: Transfer learning from CWRU bearings to Paderborn bearings

4.1. Introduction to datasets

In this section, the validity of the proposed model is demonstrated through the experimental validation of two datasets from different machines. In the case study, we try to use diagnostic information from

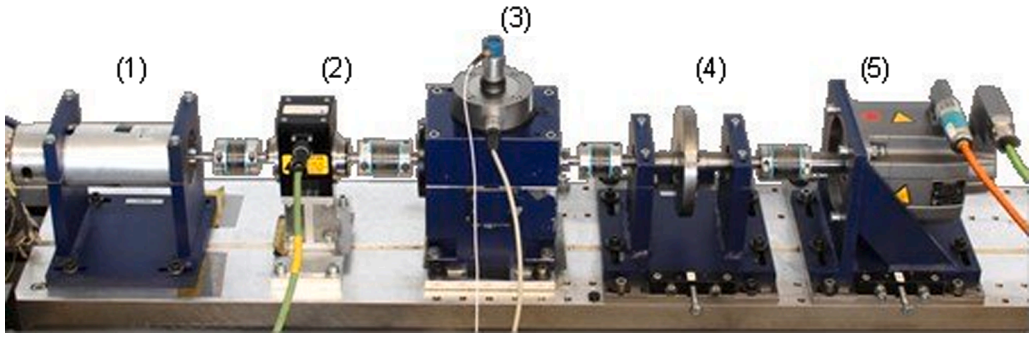

Fig. 7. The test rig of the Paderborn bearing dataset, consisting of (1) an electric motor, (2) a torque measuring shaft, (3) a rolling bearing test module, (4) a flywheel, (5) a load motor [46].

Table 3
Description of the three health states in each dataset.

Datasets	Health state	Class label	Number of samples	Number ofsample points
CWRU	Normal	0	1000	1200
	Inner race	1	1000	1200
	Outer race	2	1000	1200
PDB	Normal	0	1000 + 100(test)	6000
	Inner race	1	1000 + 100(test)	6000
	Outer race	2	1000 + 100(test)	6000

Table 4
Transfer Task.

Task Name	Source Data	Target Data
C_1	CWRU-DE-1	PDB-EDM
C_2	CWRU-DE-2	PDB-EDM
C_3	CWRU-FE-1	PDB-EDM
C_4	CWRU-FE-2	PDB-EDM
C_5	CWRU-DE-1	PDB-EG
C_6	CWRU-DE-2	PDB-EG
C_7	CWRU-FE-1	PDB-EG
C_8	CWRU-FE-2	PDB-EG
C_9	CWRU-DE-1	PDB-ALT
C_{10}	CWRU-DE-2	PDB-ALT
C_{11}	CWRU-FE-1	PDB-ALT
C_{12}	CWRU-FE-2	PDB-ALT

one motor bearing to diagnose the health information of another bearing, which was completely inconsistent with the working condition provided with the diagnostic information previously. These two datasets are detailed below.

The first data set is a motor bearing data set from Case Western Reserve University (CWRU) [45]. The data acquisition system consists of a 2 HP motor, a torque sensor, a power meter, and a motor control system, as shown in Fig. 6 [45]. The vibration data are collected by an accelerometer and the experimental bearings are SKF bearings. In the experiment, Electrical discharge machining (EDM) is used to simulate the single point failure of bearings, which include four health states: normal state (N), inner race fault (IF), outer race fault (OF), and roller fault (RF), and only N, IF, and OF are selected as case samples. The fault diameter of selected samples is 0.007 in.. As shown in Table 2, the sampling frequency is set to 12 kHz, and vibration data is collected with the motor load of 0HP (the motor speed of approximately 1797 r/min) and 3HP (motor speed of approximately 1730 r/min). The datasets CWRU-DE-1 (0HP) and CWRU-DE-2 (3HP) are acquired by the sensor at the drive end (DE), and the datasets CWRU-FE-1 (0HP) and CWRU-FE-2 (3HP) are captured by the sensor at the fan end (FE). The number of the sample is increased by the data augmentation described in Chapter 3.2.1, so that 3000 samples are collected in each of the four data sets, with 1000 samples for the normal state (N), 1000 samples for inner race fault (IF), and 1000 samples for outer race fault (OF). And each sample has 1,200 sampling points.

The second dataset is from the University of Paderborn, Germany [46]. The test rig consists of an electric motor, a torque measuring shaft, a rolling bearing test module, a flywheel, and a load motor, as shown in Fig. 7 [46]. The motor is operated by an inverter with a 16 kHz switching frequency, which can provide similar conditions in the industry. This dataset is sampled at 64 kHz and includes 6 healthy bearings, 12 bearings with artificial damage, and 14 bearings with damages from accelerated lifetime tests. And the dataset includes three health states: normal (N), outer race fault (OF), and inner race fault (IF). In this study, two sets of artificial damaged data are selected: the dataset for failures caused by EDM is recorded as dataset PDB-EDM, the dataset for failures caused by manual electric engraving is recorded as dataset PDB-EG. EDM is in the same mode as the damage in the CWRU dataset. The damage caused by manual electric engraving has an irregular surface structure and a deeper depth, similar to real pitting damage, and better simulates real working conditions. A dataset of accelerated lifetime tests is selected as the dataset PDB-ALT to validate the diagnostic effect of transfer learning on real working conditions. The samples are also increased by the data augmentation so that 3300 samples are collected for each damage mode, with 1100 samples from each health states (N, IF, OF). 3300 samples are split at a ratio of 10:1 for training and testing. Each sample has 6000 sampling points. The details are shown in Table 3.

According to Table 2, four datasets from the CWRU dataset are used as source domain samples to provide diagnosis knowledge and three datasets from the Paderborn (PDB) dataset are used as target domain samples. As shown in Table 4, 12 sets of transfer learning tasks are constructed to classify the samples in the datasets PDB.

Since the sampling frequency of the CWRU data set is not the same as that of the PDB data set, the samples of the PDB data set are down-sampled to 1200, as same as that in CWRU data set.

4.2. Performance evaluation and comparison analysis

4.2.1. Comparison methods

The proposed framework will be compared with several state-of-the-art methods used in the field of fault diagnosis:

- CNN

CNN is the baseline method, and as a classic framework for deep learning, it does not take any transfer operations. CNN has the same architecture as the backbone in JDFA, which means that the baseline method contains the DFA module with the same structure as JDFA.

- TCA

Table 5
Average testing accuracies (%) of different methods for transfer learning tasks.

Task Name	Baseline (CNN)	TCA [19]	DDC [47]	DAN [21]	JDA [20]	Proposed method
C_1	55.733	54.54	66.233	66.8	63.032	98.615
C_2	64.4	45.228	75.134	68.633	43.033	98.284
C_3	68.034	63.652	70.867	73.301	65.168	99.834
C_4	61.801	56.792	65.2	78.899	56.433	99.850
C_5	64.33	35.75	68.301	77.534	35.964	98.753
C_6	49.631	35.202	63.301	75.132	36.343	98.732
C_7	63.733	41.522	71.401	78.035	35.087	98.635
C_8	67.932	39.847	75.701	76.167	43.577	98.448
C_9	68.034	32.056	85.2	84.299	31.093	99.884
C_{10}	76.532	42.219	84.768	88.933	46.597	99.800
C_{11}	66.433	40.969	83.7	82.3	50.619	99.649
C_{12}	65.67	34.027	84.034	87.132	35.15	99.652
Average	64.355	43.484	74.487	78.097	45.175	**99.178**

Table 6
Baseline in-domain error of datasets PDB-EDM, PDB-EG, and PDB-ALT.

Target domain datasets	$e_b(T,T)$
PDB-EDM	0.132%
PDB-EG	1.301%
PDB-ALT	0.099%

TCA[19] is a shallow transfer learning method, which needs to extract statistical features from the raw data, then conduct unsupervised domain adaptive, and finally, make the diagnosis decision by the classifier. It reduces the distribution discrepancy between the source and target domains by marginal distribution adaption.

- DDC

DDC[47] is a deep transfer learning method, in which CNN is the backbone to extract features automatically and domain adaptation with discrepancy constraint is applied in the fully connected layers. The discrepancy constraint is constructed in previous layers of the classifier using MMD metric as one term in the network loss function.

- DAN

DAN[21] is similar to DDC and extends DDC. Instead of adding only one adaptive layer to the DDC method, DAN adds more adaptive layers simultaneously. Meanwhile, DAN uses a multi-kernel MMD metric (MK-MMD) with better characterization capability instead of the single-kernel MMD of the DDC method.

- JDA

JDA[20] is also a shallow transfer learning method, but compared to TCA it focuses on adapting both the marginal distribution and the conditional distribution, which means that the word joint in JDA does not refer to the direct adaptation of the joint distribution.

4.2.2. Metrics

Three metrics are devoted to evaluating the proposed model compared with other models, average accuracy, transfer loss, and transfer ratio [48].

- Average accuracy

The average accuracy is the mean value of testing accuracy in ten trials, referring to the accuracy measure between the predicted results and the actual labels for the unlabeled testing samples.

- Transfer loss (TL)

Before defining transfer loss and transfer ratio, the transfer error $e(S, T)$ is devoted to representing the test error obtained by conducting the training phase on the source domain S and running the testing phase on the target domain T. And the baseline in-domain error $e_b(T,T)$ represents the testing error obtained by a baseline model trained and test on the target domain T. Therefore, transfer loss TL for source domain S and target domain T is defined as the difference between the transfer error and the baseline in-domain error, i.e.

$$TL(S,T) = e(S,T) - e_b(T,T). \tag{16}$$

When the transfer loss is smaller, the model is considered to have a better performance.

- Transfer ratio (TR)

Transfer ratio TR is defined by

$$TR = \frac{1}{m \times n} \times \sum_{j=1}^{n} \left[\sum_{i=1}^{m} \frac{\left[1 - e(S_i, T_j)\right]}{\left[1 - e_b(T_j, T_j)\right]} \right] \tag{17}$$

which represents the overall transfer performance on $m \times n$ transfer learning tasks [23]. Contrary to transfer loss, the larger transfer ratio represents better transfer performance.

4.2.3. Results and performance analysis

In model comparison experiments, the baseline model is with CNN only. The average testing accuracies of contrastive methods are shown in Table 5. The testing accuracy of the proposed method on each task is over 98%, the average accuracy of all tasks reaches 99.178%, which is superior to all comparison methods. Besides, the proposed method can achieve relatively steady and excellent transfer accuracy for the transfer task under different conditions. The accuracies of the baseline method CNN are around 60%, which means basic CNN without distribution discrepancy constraint can hardly realize the fault identification of complex transfer tasks. It means that only utilizing basic CNN to extract features is not sufficient to deal with serious distribution discrepancy between source domain and target domain. The results from TCA and JDA are just as bad (about 45% on average testing accuracy), proving that the shallow transfer learning methods are not advantageous in these complex transfer tasks. These shallow transfer learning methods use simple non-linear mappings to extract features, which makes them difficult to fit complex and serious distributions. When dealing with more drastic domain shifts like transfer across different machines (TDM) scenarios, these methods have difficulty extracting powerful and rich features to narrow the distribution discrepancy and result in insufficient error correction between the source and target domains. As a result, they deliver poor results on the target domain. Relatively speaking, DDC and DAN, which are based on deep transfer learning, perform well in terms of accuracy, transfer loss, and transfer ratio. The accuracies reach about 70% on tasks 1–8 and about 90% on tasks 8–12, which is since deep neural network have certain capability to extract hierarchical features from vibration data and MMD can achieve the goal of narrowing the distribution discrepancies between the source and target domains, to improve the effectiveness of transfer learning. However, their results are still inferior to our proposed model, because MMD only considers the marginal distribution but does not joint distribution. MMD is not sufficient and appropriate enough to measure distribution discrepancy accurately for complex TDM scenario. Besides, DDC and DAN make use of basic CNN architecture without optimized operations to extract features that is not adequate to extract advantaged features for fault diagnosis under TDM scenarios. On the aspect of the average accuracy of all

Table 7
Transfer loss (TL) of different methods for transfer learning tasks.

Task Name	Baseline (CNN)	TCA [19]	DDC [47]	DAN [21]	JDA [20]	Proposed method
C_1	44.135	45.328	33.635	33.068	36.836	1.253
C_2	35.468	54.64	24.734	31.235	56.835	1.584
C_3	31.834	36.216	29.001	26.567	34.7	0.034
C_4	38.067	43.076	34.668	20.969	43.435	0.018
C_5	34.535	63.115	30.564	21.331	62.901	0.112
C_6	49.234	63.663	35.564	23.733	62.522	0.133
C_7	35.132	57.343	27.464	20.83	63.778	0.230
C_8	30.933	59.018	23.164	22.698	55.288	0.417
C_9	31.867	67.845	14.701	15.602	68.808	0.017
C_{10}	23.369	57.682	15.133	10.968	53.304	0.101
C_{11}	33.468	58.932	16.201	17.601	49.282	0.252
C_{12}	34.231	65.874	15.867	12.769	64.751	0.249
Average	35.189	56.061	25.058	21.448	54.37	**0.367**

Table 8
Transfer ratio (TR) of different methods.

Methods	Baseline (CNN)	TCA [19]	DDC [47]	DAN [21]	JDA [20]	Proposed method
Transfer ratio (TR)	0.6551	0.4243	0.7956	0.8081	0.4536	0.9963

tasks, DDC is 74.487% and DAN is 78.097%, where the difference is not obvious, proving that in these transfer tasks, the adaptation with multi-layer MDD in the full connection layers does not show a significant improvement over that with single-layer MMD.

The datasets PDB-EDM, PDB-EG, and PDB-ALT are used to calculate the baseline in-domain errors, which are shown in Table 6.

With the result of the baseline in-domain errors shown in Table 6, the transfer loss and transfer ratio are calculated in Table 7 (calculated by Eq. (17)) and Table 8 (calculated by Eq. (18)), which show consistent conclusions with the average accuracy. The average transfer loss of the proposed JDFA is only 0.367, which is lower than the value of any other method (CNN is 35.189, TCA is 56.061, JDA is 53.304, DDC is 25.058 and DAN is 21.448), indicating that JDFA has a smaller inter-domain transfer error when tackling distribution discrepancy in TDM scenario.

Furthermore, the transfer ratio shown in Table 8 is 0.9963, which is upper than the others, showing that JDFA performs higher transfer efficiency for all transfer tasks. The excellent performance on both transfer loss and transfer ratio proves that JDFA also shows superior results in terms of transfer effects compared to other comparison methods. The improvement of transfer effects in view of transfer loss and transfer ratio mainly benefits from the capability of DFA module for feature extraction and aggregation, and JMMD constraint for evaluating distribution discrepancy during transfer network training. The results of the comparison methods in transfer loss and transfer ratio are also consistent with the accuracy demonstrated.

In order to present a more visualized performance comparison, experimental results are illustrated in Fig. 8 in the meantime. As shown in Fig. 8 (a-c), the proposed method JDFA shows absolute advantages in three metrics (accuracy, TL, and TR) compared with other transfer learning algorithms.

The experimental results show that the proposed model JDFA has superior diagnostic accuracy and transfer effects than other comparative models, which is due to its powerful feature extraction module DFA and narrowing the domain distribution discrepancy by joint distribution adaptation. Compared to the proposed JDFA, CNN ignores the distribution discrepancy of the training and testing sets, TCA and JDA have difficulty extracting powerful and rich features for solving the problem of large differences in feature space, DDC and DAN also struggle to achieve good diagnostic results when faced with problems that vary more widely in distribution. Nevertheless, there is still room for improvement in the proposed model, for example, the performance on certain tasks and the stability of different tasks could be improved.

4.2.4. Feature visualization

For visualization, t-distributed stochastic neighbor embedding (t-SNE) [49] is introduced to understand the effects of transfer learning intuitively. T-SNE is used to reduce the dimension of data so that the distribution of the features can be visually displayed on a two-dimensional plane. Taking task 1 for example, the learned features from the source and the target domain are shown in Fig. 9(a-f) via t-SNE, which are obtained by CNN(a), TCA(b), DDC(c), DAN(d), JDA(e) and JDFA(f) respectively.

Through Fig. 9(a-f), the transfer effectiveness of each model can be more intuitively demonstrated. From Fig. 9(a), when the samples from the source domain can be classified, the features obtained by CNN training show large distribution discrepancies between the source domain and target domain. Thus, CNN is difficult to classify the samples from the target domain through the classifier trained with source domain samples. From Fig. 9 (b) and Fig. 9 (e), the features obtained by TCA and JDA training exhibit very poor distributions, and it is difficult to realize the distinction between different categories. Nevertheless, JDA also demonstrates an advantage over TCA in that the source domain features obtained by JDA can be separated but those obtained by TCA cannot be. From Fig. 9(c-d), DDC and DAN have the same problem as CNN, even though the categories of the source samples are well separated, the distances between the categories of the target domain samples are too small to classify the target samples correctly. However, DDC and DAN conduct layer distribution adaptation through MMD to narrow the distribution discrepancies between the source domain and target domain. Thus, DDC and DAN can achieve a better classification effect for samples in the target domain than CNN. Fig. 9(f) shows that the proposed method JDFA achieves excellent transfer results, and not only the samples with different labels are well separated, but also the samples from different domains are well-matched in distribution.

4.3. Sensitivity analysis

4.3.1. Parameter sensitivity

In the proposed model, the trade-off parameters λ will have a serious impact on the performance of the transfer model. Thus, the impact of trade-off parameters λ is investigated. In our study, the parameter λ is calculated by

$$\lambda = \frac{2}{\left(1 + e^{-\gamma(i/iteration)}\right)} - 1 \tag{18}$$

then λ is set by searching $\gamma \in \{10, 1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$.

Another important hyper-parameter is the trade-off parameter α of the L2 regularization term. The L2 regularization term is attached to the overall loss function for optimization by limiting the neural network to learn the high-frequency component and preferring a low-frequency smooth function to alleviate overfitting. α is searched from $\{10^{-9}, 10^{-10}, 10^{-11}, 10^{-12}\}$.

Task 1 and Task 5 are chosen as examples to show the transfer performance of the proposed model as γ and α change. The mean values of testing accuracy in ten trials on Task1 and Task 5 are presented in Fig. 10, and the darkest color are shown in red circles, which corresponds to the highest accuracy. Through experimental verification, it can be observed that changes in parameters do have an impact on test performance, and the proposed method achieves the highest average accuracy when γ and α are set as 0.1 and 1×10^{-10} respectively.

4.3.2. Condition sensitivity

In this section, the influence of the different working conditions on

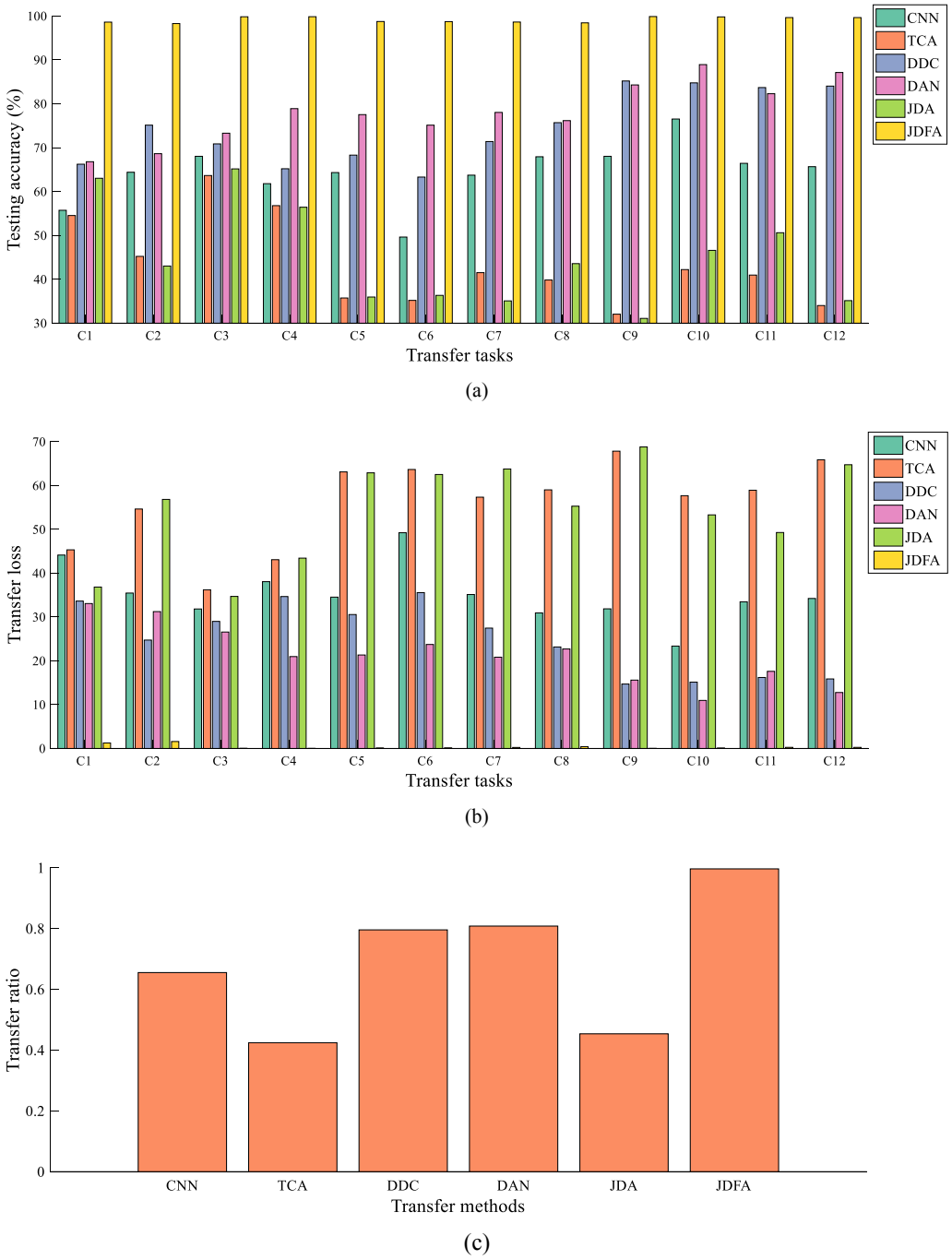

Fig. 8. (a) Testing accuracy, (b) transfer loss, (c) transfer ratio of the proposed method compared with the comparison methods on tasks 1–12.

diagnostic performance is investigated. Fig. 11 presents the testing accuracy of the proposed method under various working conditions. It can be observed that in 12 tasks, the testing accuracy rate is more than 98%, which means that the model can achieve stable and excellent results under complex working conditions. That validates the robustness of the proposed model. Meanwhile, we also find that transfer performance is stable overall with local fluctuation appearing. More detailed analyses are shown as follows:

1) For tasks where the source domains are only from different operating conditions, such as Task 1&2 or 3&4, there is no great difference in testing accuracy between them, which means that changes in operating conditions do not have a significant effect on the transfer performance.
2) For tasks where the target domain is from a different damage mode, for example in the comparison of Tasks 5&6&7&8 and with Tasks 9&10&11&12, will show greater differences in transfer performance, implying that it will be easier to exist fluctuation in transfer effects across datasets when there is serious inconsistency in the distribution discrepancy in the datasets.

4.4. Ablation analysis

In section 3 mentioned above, the proposed JDFA architecture with four modules attached to the CNN-based backbone, including multiple-scale receptive field module, residual module, SE-Module, and channel shuffle [38] module is proposed to achieve optimal performance. To validate the effectiveness of each module, ablation analyses are

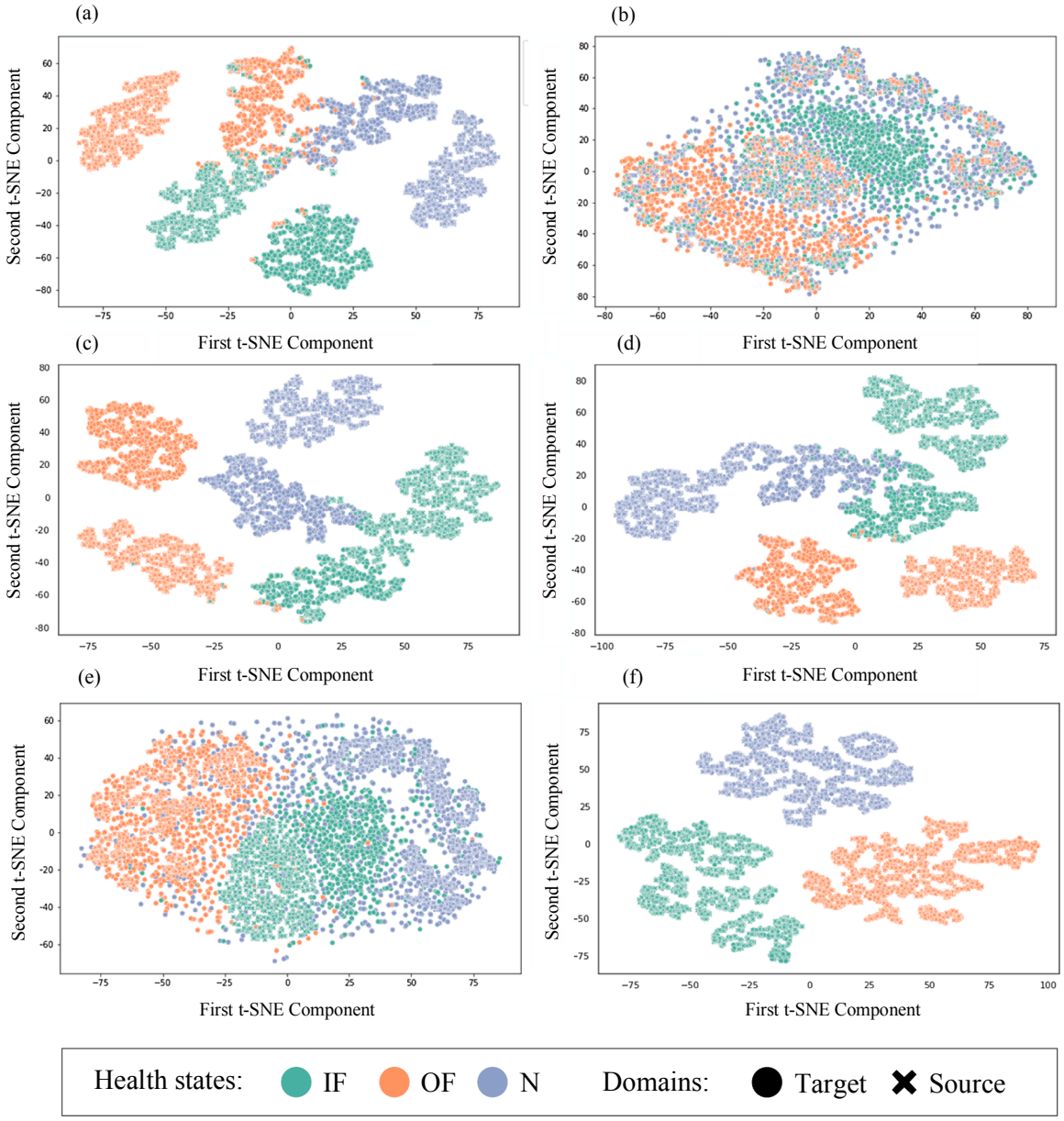

Fig. 9. The visualization of the learned features on the target domain and the source domain on task 1: (a) CNN, (b) TCA, (c) DDC, (d) DAN, (e) JDA, (f) JDFA (the proposed method).

conducted to compare JDFA with the variants in Table 9. The visualized comparison results are shown in Fig. 12.

As shown in Table 9, S_1 refers to the baseline method, which uses CNN only with JMMD. S_6 refers to the proposed model JDFA, which adds multiple-scale feature extractor, residual module, SE-module and, channel shuffle module on the baseline model. S_2 to S_5 represent models without multiple-scale feature extractor, residual module, SE-module and, channel shuffle module respectively. The prediction accuracy is calculated on each task with several experiments and obtain the mean value of each experiment as the metric to facilitate comparison. Meanwhile, TR is also calculated as an important evaluation metric.

Through the analysis of results in Table 9, the method S_6 is proved to have almost 100% prediction accuracy and the highest TR value. That is, the proposed JDFA architecture with four optimized operations added to the CNN backbone can achieve the best performance, which means that combining the four optimized operations makes an essential contribution to the fault diagnosis problem. In contrast, average prediction accuracy and TR value of the method S_1 only with CNN backbone are far lower than the method S_6 , especially in the tasks C_1 , C_5 , C_7 , C_8 . Therefore, it can be inferred that basic CNN architecture does not possess enough ability to extract features and make classification under complex tasks. What's more, it can be found that methods with one optimized operation ablated may perform even worse than basic CNN network under certain tasks through analyzing S_2 - S_5 . For example, the method S_2 which does not contain multiple-scale feature extractor achieves low prediction accuracy less than 80% under tasks C_4 and C_5 . Results of tasks C_1 , C_2 , C_3 , C_4 , C_8 of the method S_3 without residual module are much less than satisfactory. The method S_4 ablating SE-Module cannot perform well especially under the tasks C_1 and C_8 , of which accuracy is lower than 80%. Analyses are conducted for the above situations and corresponding reasons are listed as follows:

Firstly, when the multiple-scale feature extractor module is removed from the overall architecture, the single-scale CNN filter may be inappropriate for extracting features in original data and the number of output feature channels after each layer becomes much less than a complete neural network. That is, fewer types of output features are

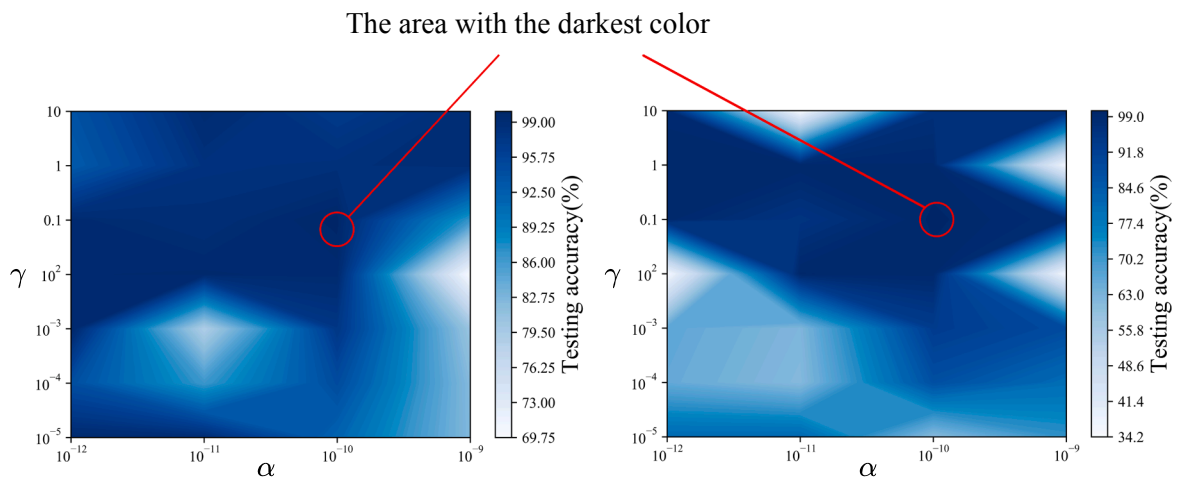

Fig. 10. Testing accuracy on Task1 (a) and Task5 (b) w.r.t γ and α .

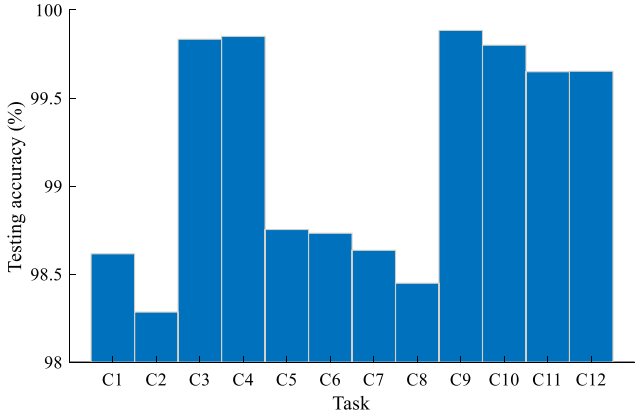

Fig. 11. Testing accuracy of the proposed method under various working conditions (in 12 tasks).

provided to the following network layers. In this condition, adding SE-Module to learn channel weights and residual module to fuse features obtained from different layers may cause high weights allocation for inappropriate features and wrong feature fusions under some tasks. Secondly, when the residual module is ablated from the framework proposed, outputs of shallow layers representing local and detailed features and deep layers representing abstract and semantic features have no opportunity to be fused. Therefore, features from shallow layers may be submerged and abstract features obtained from deep layers will dominate the following prediction process. Furthermore, only adding multiple-scale feature extractor and SE-Module will increase the number of network layers, causing gradient vanishment and network degeneration when training neural network under some tasks. Thirdly, when adopting multiple-scale feature extractor, residual module, channel shuffle module and reject SE-Module from the overall architecture, output channels per network layer will be attached to the same importance even if output channels contain complex and diverse features. It is obvious that various features in different channels have varying degrees of impact on final prediction results, especially when lots of channels exist after multiple-scale feature extractor. Lack of weights for each channel will make channel features with different importance be treated equally, which may fade useful features and emphasize useless features. For the reasons mentioned above, the method S_4 predicts with low accuracy under tasks C_1 and C_8 . Finally, the method S_5 with channel shuffle ablated is proved to have better performance than S_1 - S_4 , which means that the architecture with multiple-scale feature extractor, residual module, and SE-Module have the capability to predict fault types

Table 9
Comparison of JDFA with the variants.

Methods		S_1	S_2	S_3	S_4	S_5	S_6
Multiple-scale feature extractor?				√	√	√	√
Residual module?			√		√	√	√
SE-Module?			√	√		√	√
Channel shuffle?			√	√	√		√
Accuracy	C_1	83.834	85.866	75.400	71.068	92.900	**98.615**
	C_2	89.399	87.843	76.232	83.900	89.466	**98.284**
	C_3	88.067	89.112	75.167	96.435	89.268	**99.834**
	C_4	89.233	76.934	74.000	96.166	93.300	**99.850**
	C_5	86.500	71.934	88.432	84.202	87.334	**98.753**
	C_6	89.834	89.278	92.466	84.366	94.167	**98.732**
	C_7	84.067	84.233	81.634	80.399	83.568	**98.635**
	C_8	78.766	90.667	70.232	76.800	86.300	**98.448**
	C_9	93.066	85.991	92.800	84.033	97.767	**99.884**
	C_{10}	91.501	92.792	92.467	96.667	88.932	**99.800**
	C_{11}	92.801	90.694	88.000	98.499	88.268	**99.649**
	C_{12}	92.301	91.028	91.667	95.500	96.334	**99.652**
	Average	88.281	86.364	83.208	87.336	90.634	**99.178**
Transfer ratio (TR)		0.8867	0.8675	0.8359	0.8771	0.9104	**0.9963**

Table 10
Parameter setting of overall network.

Layers	Modules	Components	Receptive field size	Output channel
Layer1	Multiple-scale	Conv1_1	5 × 1	8
	receptive field	Conv1_2	7 × 1	8
		Conv1_3	9 × 1	8
	Conv1_4	–	5 × 1	24
	Conv1_5	–	5 × 1	24
	SE- Module	Global pooling1	1200	–
		Dense1_1	24	–
		ReLU	–	–
		Dense1_2	2	–
		Sigmoid	–	–
	ReLU	–	–	–
	Avg-pooling1	–	5 × 1	–
Layer2	Multiple-scale	Conv2_1	5 × 1	32
	receptive field	Conv2_2	7 × 1	32
		Conv2_3	9 × 1	32
	Conv2_4	–	5 × 1	96
	Conv2_5	–	5 × 1	96
	SE- Module	Global pooling2	240	–
		Dense2_1	96	–
		ReLU	–	–
		Dense2_2	6	–
		Sigmoid	–	–
	ReLU	–	–	–
	Avg-pooling2	–	5 × 1	–
Layer3	Multiple-scale	Conv3_1	5 × 1	128
	receptive field	Conv3_2	7 × 1	128
		Conv3_3	9 × 1	128
	Conv3_4	–	5 × 1	384
	Conv3_5	–	5 × 1	384
	SE- Module	Global pooling3	48	–
		Dense3_1	348	–
		ReLU	–	–
		Dense3_2	24	–
		Sigmoid	–	–
	ReLU	–	–	–
	Avg-pooling3	–	5 × 1	–
Layer4	Multiple-scale	Conv4_1	5 × 1	512
	receptive field	Conv4_2	7 × 1	512
		Conv4_3	9 × 1	512
	Conv4_4	–	5 × 1	1536
	Conv4_5	–	5 × 1	1536
	SE- Module	Global pooling4	9	–
		Dense4_1	1536	–
		ReLU	–	–
		Dense4_2	96	–
		Sigmoid	–	–
	ReLU	–	–	–
	Avg-pooling4	–	5 × 1	–
Tanh			–	
FC1	–	–	1536	–
FC2	–	–	128	–
Softmax	–	–	–	–

accurately. The channel shuffle module is helpful to make the trained neural network sufficiently stable and generalizable instead of improving final accuracy essentially.

Through analysis of the ablation experiment, the architecture proposed with all additional operations can achieve the best performance. Every module in the overall framework is unique and the ablation of a certain module will cause worse prediction results under all 12 tasks.

5. Discussion

For transfer learning studies, the challenge often lies in lack of labeled data in the target domain, while the proposed method can reduce the discrepancy between the source and the target domain, to conduct fault diagnosis for target domain samples lacking labels through the diagnostic information of source domain samples. In the proposed model, diverse feature aggregation (DFA) and joint distribution adaptation are developed. Through diverse feature aggregation, the CNN backbone is improved to obtain better feature extraction capability. The joint distribution adaptation is realized through JMMD to reduce the cross-domain discrepancy. The proposed method has obvious advantages over other methods in the TDM scenario. JDFA has the capability to output remarkable transfer performance with stability and robustness through case-by-case analysis. For all 12 cross-machine transfer tasks/cases with diverse distribution discrepancies, the proposed JDFA can successfully diagnose the health status of unlabeled PDB samples through labeled CWRU samples with diagnostic accuracy above 98% and transfer loss less than 2. That is, corresponding results show superiority when comparing with other state-of-the-art frameworks even if evaluating performance for each transfer case. Furthermore, the proposed DFA and JMMD can be easily embedded in other machine learning models or deep learning models to achieve the transfer effect.

In order to conduct the study of cross-machine transfer learning, the large discrepancy in sample characteristics between the source and target domains is another challenge. Based on Tables 5 and Fig. 8, it can be learned that the diagnostic performance of either model will always vary on different tasks. For example, tasks 9–12 always perform better than tasks 5–8. This may result from the narrower distribution discrepancy between the source and target domains for tasks 9–12. Obviously, CNN does not take into account the discrepancies in feature distributions, and shallow transfer learning methods (TCA, JDA) are also difficult to obtain good transfer results for large differences in feature space across machines. Deep transfer learning models such as DDC and DAN have better transfer results for tasks with small discrepancies in distributions, but still, fail to obtain satisfactory results for tasks with large discrepancies in distributions. In the task of transfer learning, it is inevitable that there is a huge discrepancy between the source domain and target domain, which is well solved by the proposed model through the joint distribution adaptation combined with diverse feature aggregation. The comparison experiment also proves the superiority of the proposed model. Moreover, due to data augmentation applied in the experiment, a small number of samples from the CWRU data set and PDB data set are expanded, so that cross-machine fault diagnosis also could be realized in the case of small samples. In addition, simple data preprocessing, such as down-sampling and normalization can reduce the distribution discrepancies of sample characteristics between the source domain and target domain.

To some extent, the proposed model JDFA is not only a model for bearing diagnosis but also a set of methods for mechanical fault diagnosis. Its application can be extended to other scenarios, such as spindles blade tools, gearboxes, and other mechanical equipment. We hope to expand its application in more scenarios in future work.

On the other hand, even though the proposed JDFA framework can achieve excellent and steady transfer performance for bearing diagnosis under cross-machine scenarios, fractional experiment results raise unsolved issues that should be considered in future work. Firstly, promotion tests and applications in more actual factory scenarios should be carried out to improve the transferability for complex data of the model. Secondly, through sensitivity analysis and ablation analysis, it can be

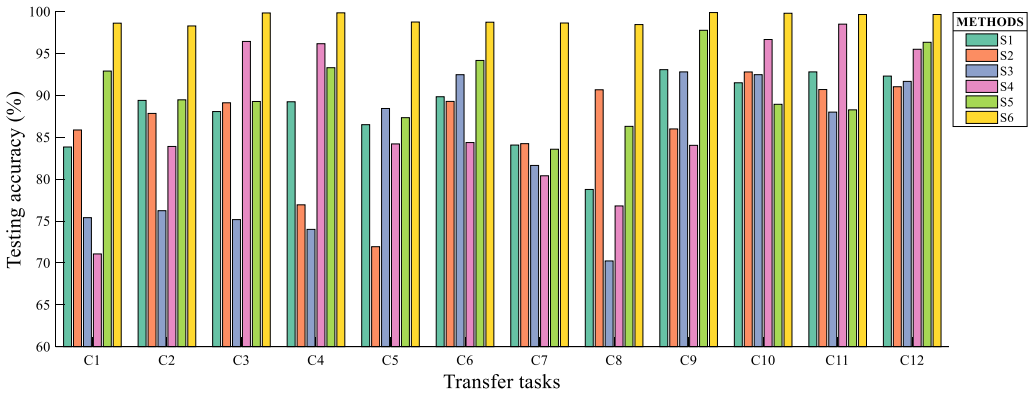

Fig. 12. Testing accuracy of the methods S_1 - S_6 on tasks 1–12.

found that the accuracies for some tasks are relatively poor, and the stability or robustness of the diagnosis framework is needed to be improved in future work when network structure is modified and network parameters are adjusted. This may be crucial for the diagnosis system to tackle more types of transfer tasks/cases or actual factory scenarios with complex data distribution. In the following research, mode advanced neural network structure and more detailed analysis for formulizing distribution discrepancy should be explored. Thirdly, it is still a lack for considering the impact of model computing overhead, which may affect the practical deployment in an actual factory environment. The testing time (diagnosis time) of the proposed framework is about 0.0237 s, considering the impact of model computing load on the actual situation of the factory, a more efficient transfer algorithm should be developed to reduce computing overhead and achieve a tradeoff between transfer performance and computing overhead.

6. Conclusion

In the paper, we propose a transfer learning framework named JDFA to solve the problem of bearing diagnosis across different machines with drastic domain variance. Based on pre-processed vibration signal from the source and the target domain respectively, the proposed JDFA method extracts hierarchical features by diverse feature aggregation (DFA) extractor, and constrains the distribution discrepancy between source and target domain by formulizing and minimizing JMMD, to realize the health diagnosis for the target domain samples. In order to demonstrate the advantages of the proposed method, the JDFA framework is verified under 12 cross-machine transfer learning tasks based on CWRU and PDB data sets in evaluation phase. Validation results show that JDFA can achieve outstanding transfer performance with 99.178% in diagnosis accuracy, 0.367 for transfer loss, and 0.9963 in view of transfer ratio, which outperform other state-of-the-art diagnosis frameworks for comparison. The above results analysis reflects that the proposed method has promising application prospects for bearing diagnosis in practical industrial scenarios through experimental verification.

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Shiyao Jia: Conceptualization, Methodology, Software, Writing – original draft. **Yafei Deng:** Investigation, Validation. **Jun Lv:** Writing – review & editing. **Shichang Du:** Supervision. **Zhiyuan Xie:** Visualization.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research acknowledges the financial support provided by National Natural Science Foundation of China (Grant No. 51775343) and Shanghai Pujiang Program (Grant No. 18PJC031).

Appendix

The parameter setting of overall feature extracting network, i.e., DFA extractor is presented in detail in Table 10.

Simulation system information: The proposed architecture is constructed and conducted under Ubuntu 16.04 system, hardware platform with 256 GB Memory, Intel Xeon E5-2683 v4 @ 2.1Ghz CPU, and Nvidiamailto:v4@2.1Ghz, Nvidia Geforce GTX 1080Ti 11G GPU.

Meanwhile, following environment and dependent libraries are needed when training and testing the proposed framework, as shown in Table 11.

Table 11
Environment and dependent libraries for experiment.

Environment and dependent libraries	Function
Python 3.8.2	The basic programming language
Pytorch-gpu 1.5.0	The deep learning framework to construct proposed neural network and other network architecture for comparison
Scikit-learn 0.23.2	Construct TCA and JDA framework
Numpy 1.18.1	Complete corresponding matrix operation
Pandas 1.1.4	Complete pre-processing of input data
Seaborn 0.11.0	Plot the distribution of data from source domain and target domain
Tensorboard 2.2.1	View the accuracy-iteration curve during training process
Scipy 1.4.1	Deal with some scientific computing

References

[1] J. Wang, Y. Ma, L. Zhang, R.X. Gao, D. Wu, Deep Learning for Smart Manufacturing: Methods and Applications, Journal of Manufacturing Systems (2018) 144–156.

[2] L. Guo, Y. Lei, S. Xing, T. Yan, and N. Li, "Deep Convolutional Transfer Learning Network: A New Method for Intelligent Fault Diagnosis of Machines With Unlabeled Data," *IEEE Transactions on Industrial Electronics,* vol. PP, pp. 1-1, 2018.

[3] K. Weiss, T.M. Khoshgoftaar, D.D. Wang, A survey of transfer learning, Journal of Big Data 3 (1) (2016) 1–40.

[4] J. Dybala, R. Zimroz, Rolling bearing diagnosing method based on Empirical Mode Decomposition of machine vibration signal, Applied Acoustics vol. 77, no. mar (2014) 195–203.

[5] S. D. Wu, P. H. Wu, C. W. Wu, J. J. Ding, and C. C. Wang, "Bearing Fault Diagnosis Based on Multiscale Permutation Entropy and Support Vector Machine," *Entropy,* vol. 14, no. 8, 2012.

[6] H. Zhou, J. Chen, G. Dong, R. Wang, Detection and diagnosis of bearing faults using shift-invariant dictionary learning and hidden Markov model, Mechanical Systems & Signal Processing 72–73 (2016) 65–79.

[7] Y. Lei, B. Yang, X. Jiang, F. Jia, and A. K. Nandi, "Applications of machine learning to machine fault diagnosis: A review and roadmap," *Mechanical Systems & Signal Processing,* vol. 138, pp. 106587-, 2020.

[8] C. F, Deep Learning mit Python und Keras: Das Praxis-Handbuch vom Entwickler der Keras-Bibliothek. MITP-Verlags GmbH & Co. KG, 2018.

[9] W. Mao, J. He, Y. Li, and Y. Yan, "Bearing fault diagnosis with auto-encoder extreme learning machine: A comparative study," *Proceedings of the Institution of Mechanical Engineers Part C Journal of Mechanical Engineering Science,* p. 0954406216675896, 2016.

[10] Y. Zhang, X. Li, L. Gao, W. Chen, P. Li, Intelligent fault diagnosis of rotating machinery using a new ensemble deep auto-encoder method, Measurement 151 (2019), 107232.

[11] Y. Deng, D. Shichang, J. Shiyao, Z. Chen, X. Zhiyuan, Prognostic study of ball screws by ensemble data-driven particle filters, Journal of Manufacturing Systems 56 (2020/07/01/ 2020,) 359–372, https://doi.org/10.1016/j.jmsy.2020.06.009.

[12] G. Qiu, Y. Gu, Q. Cai, A deep convolutional neural networks model for intelligent fault diagnosis of a gearbox under different operational conditions, Measurement 145 (2019) 94–107.

[13] L. Deng and D. Yu, "Deep Learning: Methods and Applications," *Foundations & Trends in Signal Processing,* vol. 7, no. 3, 2014.

[14] Y. Deng, D. Huang, S. Du, G. Li, C. Zhao, J. Lv, A double-layer attention based adversarial network for partial transfer learning in machinery fault diagnosis, Computers in Industry 127 (2021/05/01/ 2021,), 103399, https://doi.org/10.1016/j.compind.2021.103399.

[15] S.J. Pan, Q. Yang, A Survey on Transfer Learning, IEEE Transactions on Knowledge and Data Engineering (2010).

[16] H. Venkateswara, S. Chakraborty, S. Panchanathan, Deep-Learning Systems for Domain Adaptation in Computer Vision: Learning Transferable Feature Representations, IEEE Signal Processing Magazine 34 (6) (2017) 117–129.

[17] D. Wang and T. F. Zheng, "Transfer Learning for Speech and Language Processing," *Computer Science,* 2015. [Online]. Available: http://arxiv.org/abs/1511.06066v1.

[18] W. Dai, Q. Yang, G. R. Xue, and Y. Yu, "Boosting for Transfer Learning," in *Machine Learning, Proceedings of the Twenty-Fourth International Conference (ICML 2007), Corvallis, Oregon, USA, June 20-24, 2007*, 2007.

[19] S.J. Pan, I.W. Tsang, J.T. Kwok, Q. Yang, Domain Adaptation via Transfer Component Analysis, IEEE Transactions on Neural Networks 22 (2) (2011) 199–210.

[20] M. Long, J. Wang, G. Ding, J. Sun, P.S. Yu, Transfer Feature Learning with Joint Distribution Adaptation. in *Proceedings of the 2013 IEEE International Conference on Computer Vision*, 2013.

[21] M. Long, Y. Cao, J. Wang, M.I. Jordan, Learning transferable features with deep adaptation networks. presented at the Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37, 2015.

[22] E. Tzeng, J. Hoffman, K. Saenko, T. Darrell, Adversarial Discriminative Domain Adaptation. in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[23] B. Yang, Y. Lei, F. Jia, S. Xing, An intelligent fault diagnosis approach based on transfer learning from laboratory bearings to locomotive bearings, Mechanical Systems and Signal Processing 122 (2019) 692–706.

[24] Z. Wu, H. Jiang, K. Zhao, X. Li, An adaptive deep transfer learning method for bearing fault diagnosis, Measurement 151 (2019), 107227.

[25] M. Sun, H. Wang, P. Liu, S. Huang, P. Fan, A sparse stacked denoising autoencoder with optimized transfer learning applied to the fault diagnosis of rolling bearings, Measurement 146 (2019/11/01/ 2019,) 305–314, https://doi.org/10.1016/j.measurement.2019.06.029.

[26] W. Qian, S. Li, P. Yi, K. Zhang, A novel transfer learning method for robust fault diagnosis of rotating machines under variable working conditions, Measurement 138 (2019/05/01/ 2019,) 514–525, https://doi.org/10.1016/j.measurement.2019.02.073.

[27] X. Li, W. Zhang, and Q. Din, "A robust intelligent fault diagnosis method for rolling element bearings based on deep distance metric learning," *Neurocomputing,* vol. 310, no. OCT.8, pp. 77-95, 2018.

[28] F. Zhuang, Z. Qi, K. Duan, D. Xi, and Q. He, "A Comprehensive Survey on Transfer Learning," *Proceedings of the IEEE,* vol. PP, no. 99, pp. 1-34, 2020.

[29] L. Wen, L. Gao, X. Li, "A New Deep Transfer Learning Based on Sparse Auto-Encoder for Fault Diagnosis," *IEEE Transactions on Systems, Man, and Cybernetics: Systems,* vol, PP 99 (2017) 1–9.

[30] W. Qian, S. Li, and J. Wang, "A New Transfer Learning Method and Its Application on Rotating Machine Fault Diagnosis under Variant Working Conditions," *IEEE Access,* pp. 1-1, 2018.

[31] T. Han, C. Liu, W. Yang, and D. Jiang, "Deep transfer network with joint distribution adaptation: A new intelligent fault diagnosis framework for industry application," *Isa Transactions,* 2019.

[32] X. Wang, C. Shen, M. Xia, D. Wang, J. Zhu, and Z. Zhu, "Multi-scale deep intra-class transfer learning for bearing fault diagnosis," *Reliability Engineering & System Safety,* vol. 202, Oct 2020, Art no. 107050, doi: 10.1016/j.ress.2020.107050.

[33] P. L. A, C. S. A, D. W. B, L. C. A, Z. Z. C, and Z. Z. A, "A new transferable bearing fault diagnosis method with adaptive manifold probability distribution under different working conditions - ScienceDirect," *Measurement,* 2020.

[34] Z. Zhang, H. Chen, S. Li, Z. An, Unsupervised domain adaptation via enhanced transfer joint matching for bearing fault diagnosis, Measurement 165 (2020), 108071.

[35] A. Gretton, K.M. Borgwardt, M. Rasch, B. Schlkopf, A. Smola, A Kernel Two-Sample Test, Journal of Machine Learning Research 13 (2012) 723–773.

[36] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-Excitation Networks," *IEEE Transactions on Pattern Analysis & Machine Intelligence,* vol. PP, no. 99, 2017.

[37] K. He, X. Zhang, S. Ren, S. Jian, Deep Residual Learning for Image Recognition. in *IEEE Conference on Computer Vision & Pattern Recognition*, 2016.

[38] X. Zhang, X. Zhou, M. Lin, J. Sun, ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[39] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the Devil in the Details: Delving Deep into Convolutional Nets," *Computer Science,* 2014. [Online]. Available: http://arxiv.org/abs/1405.3531v4.

[40] A. G. Howard, "Some Improvements on Deep Convolutional Neural Network Based Image Classification," *Computer ence,* 2013.

[41] S. Le, B. Boots, S.M. Siddiqi, G.J. Gordon, A.J. Smola, Hilbert space embeddings of hidden Markov models, Proc Interspeech 2 (2) (2010) 140–144.

[42] J. Yosinski, J. Clune, Y. Bengio, H. Lipson, How transferable are features in deep neural networks?. in *International Conference on Neural Information Processing Systems*, 2014.

[43] N. Rahaman, et al., On the Spectral Bias of Neural Networks (2018).

[44] G. Hinton, N. Srivastava, and K. Swersky, "Neural networks for machine learning lecture 6a overview of mini-batch gradient descent," *Coursera Lecture slides,* 2012.

[45] K. A. Loparo. "Bearing vibration data: Case western reserve university bearing data center website." https://csegroups.case.edu/bearingdatacenter/home (accessed.

[46] C. Lessmeier, J.K. Kimotho, D. Zimmer, W. Sextro, 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 *European Conference of the Prognostics and Health Management Society*, 2016.

[47] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep Domain Confusion: Maximizing for Domain Invariance," *Computer Science,* 2014. [Online]. Available: http://arxiv.org/abs/1412.3474v1.

[48] X. Glorot, A. Bordes, and Y. Bengio, "Domain Adaptation for Large-Scale Sentiment Classification: A Deep Learning Approach," in *ICML*, 2011.

[49] V.D.M. Laurens, G. Hinton, Visualizing Data using t-SNE, Journal of Machine Learning Research 9 (2605) (2008) 2579–2605.