A Novel Deep Model with Meta-learning for Rolling Bearing Few-shot Fault Diagnosis

Xiaoxia Liang[1,2], Ming Zhang[3,*], Guojin Feng[1,2], Yuchun Xu[3], Dong Zhen[1,2], Fengshou Gu[4]

[1] College of Mechanical Engineering, Hebei University of Science and Technology, Tianjin 300401, China

[2] Advanced Equipment Research Institute Co., Ltd. of HEBUT, Tianjin, China

[3] College of Engineering and Physical Sciences, Aston University, Birmingham B4 7ET, UK

[4] Centre for Efficiency and Performance Engineering, University of Huddersfield, Queensgate, Huddersfield HD1 3DH, UK

*Corresponding author: m.zhang21@aston.ac.uk

Machine learning, especially deep learning, has been highly successful in data-intensive applications, however, the performance of these models will drop significantly when the amount of the training data amount does not meet the requirement. This leads to the so-called Few-Shot Learning (FSL) problem, which requires the model rapidly generalize to new tasks that containing only a few labeled samples. In this paper, we proposed a new deep model, called deep convolutional meta-learning networks (DCMLN), to address the low performance of generalization under limited data for bearing fault diagnosis. The essential of our approach is to learn a base model from the multiple learning tasks using a support dataset and finetune the learnt parameters using few-shot tasks before it can adapt to the new learning task based on limited training data. The proposed method was compared to several few-shot learning methods, including methods with and without pre-training the embedding mapping, and methods with finetuning the classifier or the whole model by utilizing the few-shot data from the target domain. The comparisons are carried out on one-shot and ten-shot tasks using the CWRU bearing dataset and a cylindrical roller bearing dataset. The experimental result illustrates that our method has good performance on the bearing fault diagnosis across various few-shot conditions. In addition, we found that the pre-training process does not always improve the prediction accuracy.

Keywords: Few-shot learning, Meta-learning, Deep model, Fault diagnosis, Bearing

1 Introduction

With the advancement in technology, the structure of modern machinery and equipment is becoming increasingly complex, meanwhile, the high requirements of reliability and increased precision must be met. In recent years, deep learning-based intelligent fault diagnosis techniques have attracted a lot of attention due to their merits,

such as robust feature extraction capability, effective processing models, cost-effective in calculation and analysis [1]–[5]. However, the excellent performance of these deep models is based on massive amounts of labeled data [6], [7]. However, in practice, industrial equipment and systems, especially the critical ones, are usually not allowed to work in a severe fault state due to safety reasons and maintenance cost considerations. Therefore, in many cases, it is difficult or even impossible to obtain sufficient labeled training samples to make the classifier robust for every fault type.

The capability of these deep models may be degraded significantly when the amount of training data does not meet the requirement. This is referred to a few-shot learning (FSL) problem [8], in which a model is required to be trained and generalized well even using one or a few data samples. The history of FSL can be traced back to the early 2000s [9], [10]. The early FSL approaches, for example, Congealing algorithm by E. G. Miller et al. [9] and Variational Bayesian framework (VBF) by L. Fei-Fei et al. [11], are non-deep models and these models are mainly based on the generative model. By applying labeled data in the source domain, this kind of model tries to estimate the joint distribution $P(X,Y)$ or the conditional distribution $P(X|Y)$. The model can make predictions for test samples using Bayesian theory, based on very few observed samples for training. In 2015, G. Koch et al. [12] firstly applied deep learning for FSL problems by proposing a Siamese convolutional network (CNN) to learn a class-irrelevant similarity metric on pairwise samples. This symbols a new era for FSL [10].

Apart from the aforementioned generative model, the discriminative model based FSL approach is getting increasingly popular recently. Two main approaches in the discriminative model are metric learning and meta-learning [10]. Metric learning tries to learn a pairwise similarity metric $S(\cdot,\cdot)$. It will give a high score if a sample pair is similar, and a low score if the pair is dissimilar. Meta-learning aims at training a model based on a variety of learning tasks and adapting a new different but related task with very limited labeled data sample. A typical model is the model agnostic meta-learning (MAML) proposed by FINN et al. [13]. The model trains a set of initial parameters, then carries out gradient adjustment in one or more steps to achieve the purpose of quickly adapting to new tasks with only a small amount of data. However, the MAML is quite sensitive to neural network structure and requires time-consuming hyperparameters search to stabilize training and improve model generalization capability [14]. Antoniou et.al [15] and Nichol et.al. [16] optimized MAML's robustness, training stability, and efficiency. Li et al. [17] proposed a meta-learning fault diagnosis method based on the MAML model, and the results show that this method has certain advantages in solving small sample fault classification problems under complex working conditions.

Although the metric learning methods are effective for the few-shot fault diagnosis tasks to a certain degree, when being taught, the models only focus on the relative similarity information from sample groups while the attribute information of each specific category is ignored, hence resulting the labeled source data not being fully exploited. To better transfer the knowledge learnt from the source domain to deal with problems in the target domain, in this paper, we prefer to solve the few-shot fault diagnosis problem using the meta-learning method. Besides, considering that the deep CNN model is proven to be robust and efficient in feature extraction, a hybrid method that combines the merit of deep

CNN supervised learning and meta-learning is proposed. In the hybrid model, the first several layers are the same as supervised learning, that is trained by source data and extracts features to recognize different fault types. These layers are fixed after training process and then worked as a feature extractor that transforms raw data to the basic feature space. Finally, meta-learning is employed to train the rest of the model with the extracted features. In this way, not only the relative information between data pairs but also the supervision information from the source domain will be well exploited by the proposed model.

In this paper, a meta-learning based deep convolutional meta-learning networks (DCMLN) is proposed and presented for few-shot fault diagnosis tasks using the CWRU dataset and a cylindrical roller bearing dataset. The background and framework is introduced and explained in Section 2 and 3, respectively. In Section 4, experiments using the CWRU dataset and a cylinder bearing dataset are demonstrated, the bearing fault prediction accuracy are listed, confusion matrix of the prediction results are illustrated, and t-SNE technique is employed to visualize the feature embedding. Finally, Section 5 presents the conclusions and discussions.

2 Background

2.1 Few-shot Learning

Traditional supervised learning has achieved significant performance when there are sufficient data. However, it often fails to make the correct prediction under the circumstances when only limited data is available. The FSL method is assigned to address such problem of identifying the unobserved categories under little support information without the certainly necessary process of retuning the model, since recollecting the data may be too expensive or sometimes even not possible.

As shown in Fig. 1, the training and test data are drawn from the same domain, and the unlabeled samples all belong to the categories already known. On the contrary, the test mission in the few-shot learning derived from the different domains, in which only very few labeled data of each category can be used. The categories of the target domain are certainly unseen in the source domain and all the unlabeled samples should be part of these unfamiliar categories.

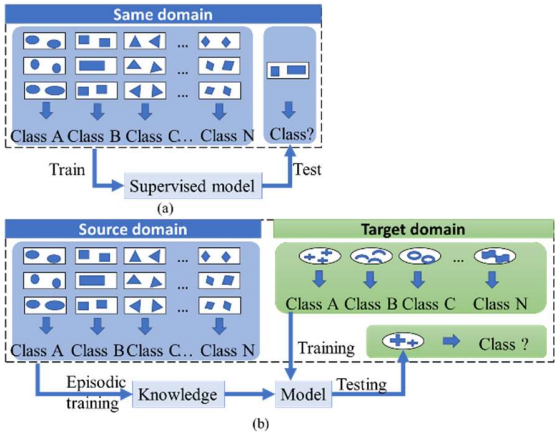


Fig. 1. Few-shot learning problem (a) traditional supervised learning method, (b) FSL method.

To be more specific, few-shot learning with k-shot and N-class occupy k labeled data samples of each N category in the supporting dataset and there is no explicit limitation of sample number in the test dataset which aims at accessing the performance of the trained model. In the typical few-shot learning setting, k usually equals 1 or 5 and N is regarded with the specific tasks.

2.2 Meta-learning

As one effective method for overcoming the few-shot challenge, meta-learning aims at training a model based on a variety of learning tasks and then quickly adapts to a

new different but related task with a very limited labeled data sample, where is the few-shot learning setting. In a meta-learning setting, there are three different metasets, which are D_{meta_train}, $D_{meta\_validate}$, and D_{meta_test}, each of them is a specific few-shot task. The meta-learner can be trained by using the D_{meta_train}, then test it on the D_{meta_test}, meanwhile, the hyper-parameter will be tuned with the $D_{meta\_validate}$. In this paper, we concentrate on the few-shot learning circumstance of different categories by utilizing the meta-learning strategy.

The metric-based meta-learning uses a set-to-set approach [10], [18] to address the fault diagnosis problem with limited available labeled data. Suppose $S = \{(x_i, y_i)\}_{i=1}^{k}$ is a support set with k labeled samples, the model can be regarded as a probability distribution that maps $P(\hat{y}|\hat{x}, S)$ from the input samples $\hat{x}$ to the output labels $\hat{y}$. Assuming there is a new support set S' for the few-shot problem, with application of the new set S', the model can directly output the label $\hat{y}$ for each test sample $\hat{x}$: $P(\hat{y}|\hat{x}, S')$. The meta-learning model based on metric can be expressed as:

$$\hat{y} = \sum_{i=1}^{k} a(\hat{x}, x_i) y_i \tag{1}$$

where x_i, y_i is from the support set $S = \{(x_i, y_i)\}_{i=1}^{k}$, and a is the metric-based attention kernel. The metric-based meta-learning outperforms typical supervised learning models in that it is a non-parametric method and it can swiftly adapt to any new support set.

2.3 Matching Networks

In the initial method of metric-based meta-learning method, a conditional classification model with the support set is defined by Matching Networks [19], which explicitly choose the softmax over the cosine distance as the similarity function α for the few-shot test sample $\hat{x}$ and the support sample x_i. The similarity kernel function is given by:

$$\alpha(\hat{x}, x_i; \theta) = \frac{exp\,[d(f(\hat{x}), f(x_i))]}{\sum_{j=1}^{k} exp\,[d(f(\hat{x}), f(x_i))]} \tag{2}$$

where θ denotes the embedding function in an appropriate neural network f; d refers to the cosine distance function.

Since the Matching Networks will directly perform on a new support set S′ which has never been seen in the period of training, the training process of which needs to be carefully designed to match inference for testing unseen few-shot tasks. Firstly, we define a few-shot task $\mathcal{T}$ from the labeled training dataset D_{meta_train}. Then, the support set S and a test batch B need to be sampled from $\mathcal{T}$ in order to shape an episode to update the model. The goal of the Matching Network is to minimize the predicted error for the test batch B under the assistance of the support set S. More precisely, the objective function of the Matching Network is as follows:

$$\underset{\theta}{maximize} \underset{\mathcal{T}\sim p(\mathcal{T})}{\mathbb{E}} \left[\underset{S, B\sim p(\mathcal{T})}{\mathbb{E}} \mathrm{A} \right] \tag{3}$$

$$A = \sum_{(x,y)\in B} log\,(P(y|x, S; \theta)) \tag{4}$$

where θ is the parameters of the embedding neural network optimized by the stochastic gradient descent (SGD) approach. After the training procedure, the trained model will work on the new few-shot task $\mathcal{T}'$. The essential of a Matching Network is related to metric learning, and since the such method is to minimize the loss function based on learns to learn from the support set over the

batch set, which is considered as one of the meta-learning methods.

3 Proposed DCMLN Model Framework

Within this paper, we focus on a few-shot learning problem in the bearing fault diagnosis under a variety of limited data conditions. The bearing fault diagnosis is usually regarded as a supervised learning problem, the goal of which is to identify the different fault locations, including inner race, outer race, and the rolling element. With sufficient training data, such problem can be easily solved. However, in real-world applications, it is difficult to collect a large amount of data samples for each category, where the few-shot learning problem happens. Note that in our works, the number of categories may be different following certain few-shot learning tasks of fault diagnosis, so we use a non-parameters metric-based meta-learning method, Matching Networks, to overcome the challenge in few-shot fault diagnosis.

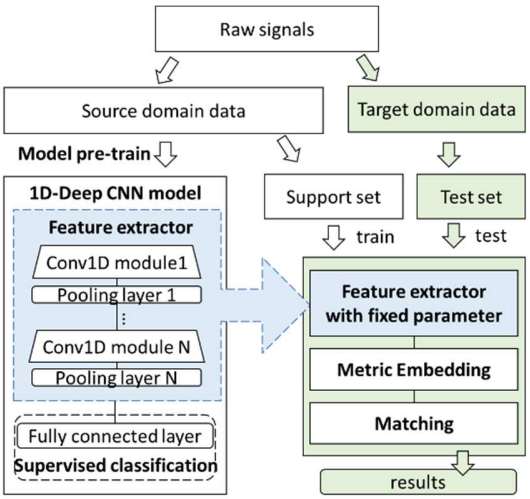


Fig. 2. Framework of the proposed DCMLN model

The framework of the proposed DCMLN model is presented in Fig. 2, which consists of three parts, including embedding network, pre-train term, and matching term. Firstly, the supervised learning method is utilized to initiate the embedding network with the Pre-train dataset $X_s = (x_i^s, y_i^s)_{i=1}^{n^s}$. Then, the metric-based meta-learning method is employed to train the embedding network. Different from the typical pre-train dataset, the proposed DCMLN approach needs to extract the amount of few-shot tasks, each of which owns a support set S_i and a test batch set B_i. When training process is completed, few-shot tasks are tested directly using embedding networks and the matching term on data sets from the unseen target domain.

3.1 Embedding Mapping Function

The vibration signal is widely employed for the diagnosis of the mechanical fault. It is a typical one-dimensional signal, and hence, the embedding mapping plays a critical part in the proposed method, which is composed of several one-dimensional convolution layers and a fully connected layer. More specifically, the one-dimensional convolutional layer is defined by Equation (5):

$$C_{ij}^{l} = \phi(k_{n\times 1}^{j} \cdot x_{i:i+n}^{i} + b_{ij}) \tag{5}$$

where, $k_{n\times 1}^{j}$ is the jth kernel belonging to the kernels $\mathrm{K}_{\mathrm{j}}^{\mathrm{l}}$ with size $n\times 1\times j$ of the l-th convolution layer; $x_{i:i+n}^{i}$ is the i th input segment; b_{ij} refers to the bias; ϕ represents the activation function; C_{ij}^{l} is the i-th feature point of the j -th kernel in the l -th convolution layer.

Following the convolutional layer, it is the fully connected layer, which can be expressed as:

$$y^{l} = \phi(W^{l} y^{l-1} + b^{l}) \tag{6}$$

where y^{l-1} is the feature map of the upper layer, b^l is the bias for the current layer; W^l is the weight matrix between the upper layer and the current layer.

3.2 Pre-train Term

In our method, we firstly use the supervised learning method to pretrain the embedding mapping with enough training dataset with labels. In order to implement this step, an output layer of K categories with the Softmax activation function needs to be attached to the embedding network. The objective function L_c of the pretrain model is the cross-entropy which is defined as follows:

$$L_c = -\frac{1}{n^s}\sum_{i=1}^{n^s}\sum_{k=1}^{K}\mathbb{I}_{y_i} = k \cdot log\frac{e^{wf(x_i)+b}}{\sum_{j=1}^{K} e^{wf(x_i)+b}} \quad (7)$$

where $\mathbb{I}$ is the indicator function; f is the embedding neural network; K is the number of categories.

3.3 Matching Term

The main loss function to overcome the few-shot challenge in the fault diagnosis circumstance is the matching loss, the essential of which is to learn the invariant distance representation in each class. In our proposed method, we train the embedding mapping based on the pretrained parameters, then minimize the matching loss to obtain the shared embedding representation of feature distance in each category. The objective function of this term is defined as follows:

$$L = \frac{1}{N_C N_Q}\sum_{k=1}^{K}[d\big(f(\hat{x}), g(x_i)\big) - log\sum_{i=1}^{k} e^{d(f(x),g(x_i))}] \quad (8)$$

where, N_C is the number of categories per episode; N_Q is the number of query samples per category; f is the embedding networks; g is the alternative mapping which is equal to f in Matching Networks; and d is the distance function. The first item of the objection function is the distance between the current class and its corresponding class in the support set, while the second represents the sum of the distance between the current class and the other class in the support set. Based on the above, we can understand this objective loss is to minimize the distance between the samples in the same category, at the same time, maximize the distance between different categories. The ultimate goal is to acquire the invariant feature mapping in each class, which also has the mapping ability of large divergence between different classes.

3.4 Model implementation

In this section, the implementation of the proposed model for the fault diagnosis task is detailed as follows:

Step 1. Data preparation

Divide the collected data into two parts. One part is the source domain X^S and one part is the target domain X^T. Since the model needs to be pre-trained on the source domain, the data in the source domain requires enough labels and sufficient data volume.

Step 2. Supervised training using deep CNN in source domain

This step aims to train a feature extractor in a supervised way using deep CNN with the fully labeled dataset X^S from the source domain. After training, the structure and parameters of the learned feature extractor was fixed.

Step 3. Episodic training in source domain

Reform the data samples of source domain X^S as the few-shot learning tasks $\{S_i^S, Q_i^S\}_{i=1}^n$, which has similar data structure with target C-way, K-shot M-test fault diagnosis tasks.

Then, the basic features extracted by feature extractor in the second step are collected and applied to train the metric embedding module to get metric features. Following this, fault classification of the query samples is conducted by matching metric features to the support ones.

Step 4. Test in target domain

In this step, the feature extractor, which is fixed after step 2 and the metric embedding module, which is trained in step 3, are applied for target fault diagnosis tasks. All the samples from target tasks are transferred into basic feature space, and a similar matching operation is conducted to predict the fault types of query data based on the provided limited support data with target domain data.

4 Experiment

In this section, two bearing datasets, with one provided by the CWRU and one collected by our group, are applied to evaluate the proposed DCMLN fault diagnosis method. Each dataset is divided into source domain and target domain. Note that sufficient labeled data are provided in the source domain to extract prior knowledge while very limited training data are included in the target domain. In this way, a scenario is provided for the few-shot fault diagnosis task.

Both cases share the same experimental settings. Adam [20] is applied for the optimizer. For the pre-training process, the learning rate is 0.001 [20], batch size is 16, the maximum epoch is set as 50, and early stop duration epoch is set as 15. The number of finetune steps is 100. In the episodic training process, the learning rate is 0.0001 [21], the distance metric uses Cosine function [19], the scale factors are set as 100, the maximum epochs are set as 80. We consider 1-shot or 10-shot tasks, so that the support samples per class (K) equals to 1 or 10. The number of query samples for each class is 25, and 600 evaluation tasks are tested.

To better assess the hybrid DCMLN method, we compare it with other five baseline few-shot learning methods, which are:

1) Finetune Last [22];
2) Finetune Whole [22];
3) Feature Knn [23];
4) Data Space Matching Network (DSMN) [19];
5) Data Space Matching Network with Pre-training (DSMN-Pre) [19];
6) Deep convolutional meta-learning networks (DCMLN, ours).

4.1. Case Study 1: CWRU Bearing dataset

4.1.1 Data description

In this case, a publicly available bearing dataset provided by the bearing center of Case Western Reserve University (CWRU) was applied to test the performance of the proposed fault diagnosis method. The testbed, as shown in Fig. 3, is composed of a motor, accelerometer, torque transducer, dynamometer, and rolling bearings. The rolling bearings were deep groove ball bearing (type SKF6205-2RSJEM) that were installed at the drive end and the fan end of the motor, respectively. The Electrical Discharge Machining (EDM) technique was applied to set up three degrees of faults on the inner ring, outer ring, and roller of the bearing with fault diameters of 0.007, 0.014, and 0.021 in., respectively. The vibration

data was gathered on four different loads (0, 1, 2, and 3 hp) with the sampling frequency of 12 kH. Each fault category contains 500 samples and each sample has 2048 points.

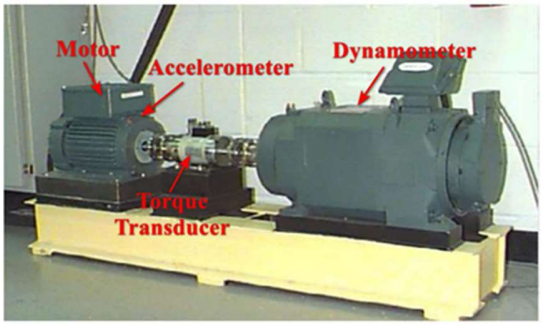


Fig. 3. CWRU testbed

Table 1 Few-shot fault diagnosis scenarios for each load condition of the CWRU bearing dataset.

Scenarios	Source domain (7 categories)	Target domain (3 categories)
1 - Inner Race fault	Healthy; Outer race fault (0.007, 0.014 and 0.021 in.); Ball fault (0.007, 0.014 and 0.021 in.)	Inner race fault (0.007, 0.014 and 0.021 in.)
2 - Outer Race fault	Healthy; Inner race fault (0.007, 0.014 and 0.021 in.); Ball fault (0.007, 0.014 and 0.021 in.)	Outer race fault (0.007, 0.014 and 0.021 in.)
3 - Ball fault	Healthy; Inner race fault (0.007, 0.014 and 0.021 in.); Outer race fault (0.007, 0.014 and 0.021 in.)	Ball fault (0.007, 0.014 and 0.021 in.)
4 - Worst IOB	Healthy; Inner race fault (0.007, 0.014 in.); Outer race fault (0.007, 0.014 in.); Ball fault (0.007, 0.014 in.)	Inner race fault (0.021 in.); Outer race fault (0.021 in.); Ball race fault (0.021 in.)

Table 1 shows the few-shot fault diagnosis scenarios for each load condition of the CWRU bearing dataset. There are four different load conditions (0, 1, 2, 3 hp), and each load condition contains four scenarios, with seven categories in the source domain and three in the target domain.

4.1.2 Results and analysis

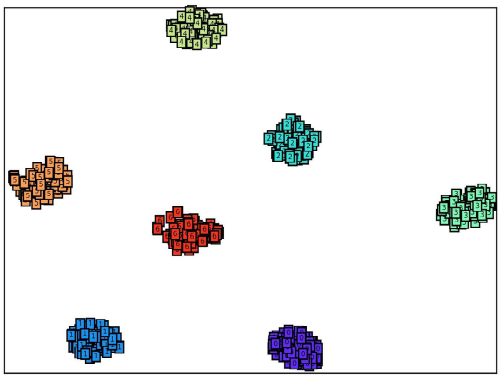
The classification accuracy on 1-shot and 10-shot learning tasks for bearing fault with different types under four different load conditions (0, 1, 2, 3 hp) are presented in Table 2 and Table 3, respectively. As can be seen, the proposed DCMLN fault diagnosis model achieves the best performance among all six models for both 1-shot and 10-shot learning tasks. In addition, the prediction accuracy is apparently improved when the number of shots increased from 1-shot to 10-shot. The proposed DCMLN fault diagnostic method has very good performance on the scenarios, including inner race fault, ball fault, and the worst IOB. Besides, all the MN methods and Feature Knn perform very well in these tasks, and there are no obvious differences between the MN methods. However, the prediction accuracy for the outer race fault scenario is lower compared to those of other three scenarios.

Table 2 Prediction accuracy (%) by DCMLN on 1-shot learning tasks for bearing fault classification using CWRU dataset

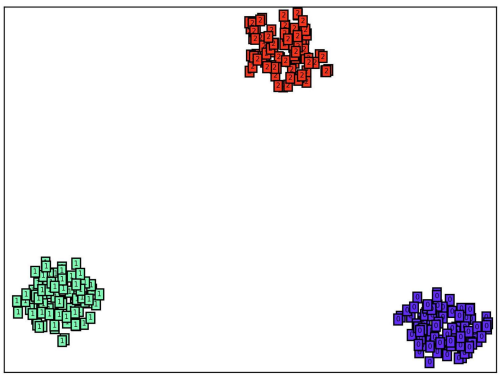
	Load 0			
	Inner Race	Outer Race	Ball	Worst IOB
Finetune Last	86.373±1.124	56.149±1.291	72.400±0.785	73.158±1.452
Finetune Whole	97.340±0.450	61.776±0.922	88.613±0.624	91.693±0.742
Feature Knn	99.931±0.024	70.484±0.967	72.662±0.493	96.149±0.218
DSMN	100.000±0.000	68.176±0.731	96.249±0.374	98.198±0.200
DSMN-Pre	99.947±0.062	70.309±0.890	83.831±0.753	99.082±0.122
DCMLN(ours)	**100.000±0.000**	**71.720±0.976**	**96.760±0.293**	**99.580±0.074**
	Load 1			
	Inner Race	Outer Race	Ball	Worst IOB
Finetune Last	81.191±1.321	58.498±1.444	91.129±0.814	84.511±1.356
Finetune Whole	99.067±0.246	79.742±1.023	93.058±0.943	96.351±0.486
Feature Knn	100.000±0.000	82.844±0.806	99.431±0.097	99.284±0.091
DSMN	100.000±0.000	77.836±0.974	99.162±0.128	99.438±0.073
DSMN-Pre	99.993±0.010	82.433±0.850	85.036±0.680	97.629±0.241
DCMLN (ours)	**100.000±0.000**	**91.160±0.528**	**99.967±0.017**	**99.696±0.057**
	Load 2			
	Inner Race	Outer Race	Ball	Worst IOB
Finetune Last	78.958±1.556	66.573±1.688	66.431±1.360	70.593±1.582
Finetune Whole	97.929±0.394	84.329±0.753	92.184±0.758	95.278±0.571
Feature Knn	100.000±0.000	93.456±0.508	93.147±0.545	99.993±0.008
DSMN	99.909±0.028	**94.204±0.417**	98.887±0.144	**100.000±0.000**
DSMN-Pre	100.000±0.000	84.442±0.542	98.662±0.196	99.556±0.072
DCMLN(ours)	**100.000±0.000**	84.562±0.772	**99.856±0.044**	99.996±0.006
	Load 3			
	Inner Race	Outer Race	Ball	Worst IOB
Finetune Last	80.029±1.433	70.989±1.534	70.567±1.111	78.929±1.536
Finetune Whole	97.536±0.424	86.453±0.801	85.096±0.852	80.838±0.910
Feature Knn	99.004±0.109	73.809±0.693	95.576±0.355	92.700±0.510
DSMN	99.424±0.072	78.071±0.685	**99.253±0.110**	99.664±0.070
DSMN-Pre	**99.996±0.006**	88.469±0.677	95.120±0.488	**99.991±0.009**
DCMLN(ours)	99.764±0.050	**89.324±0.733**	94.342±0.368	99.971±0.018

Table 3 Prediction accuracy (%) by DCMLN on 10-shot learning tasks for bearing fault classification using CWRU dataset

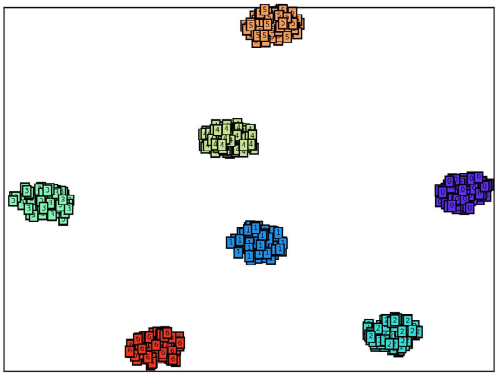
	Load 0			
	Inner Race	Outer Race	Ball	Worst IOB
Finetune Last	84.227±1.292	61.433±1.016	67.218±0.690	84.067±1.391
Finetune Whole	99.869±0.046	87.818±0.368	99.853±0.043	98.660±0.148
Feature Knn	100.000±0.000	86.944±0.329	95.976±0.211	99.453±0.076
DSMN	100.000±0.000	80.933±0.411	**100.000±0.000**	**99.776±0.046**
DSMN-Pre	100.000±0.000	**88.938±0.313**	98.642±0.110	98.487±0.143
DCMLN(ours)	**100.000±0.000**	86.671±0.333	99.996±0.006	99.298±0.087
	Load 1			
	Inner Race	Outer Race	Ball	Worst IOB
Finetune Last	85.347±1.257	70.756±1.144	76.833±1.249	87.038±1.031
Finetune Whole	99.880±0.048	92.458±0.313	98.513±0.167	99.151±0.111
Feature Knn	100.000±0.000	**94.867±0.243**	99.962±0.018	98.947±0.121
DSMN	100.000±0.000	85.456±0.397	**100.000±0.000**	98.220±0.145
DSMN-Pre	100.000±0.000	94.147±0.229	99.960±0.018	99.782±0.048
DCMLN(ours)	**100.000±0.000**	94.318±0.241	99.991±0.009	**100.000±0.000**
	Load 2			
	Inner Race	Outer Race	Ball	Worst IOB
Finetune Last	89.827±1.077	72.302±1.602	87.689±0.956	83.629±1.192
Finetune Whole	99.951±0.026	97.951±0.196	99.011±0.128	99.887±0.040
Feature Knn	99.987±0.011	**98.702±0.114**	98.296±0.128	100.000±0.000
DSMN	99.980±0.017	95.078±0.237	97.171±0.175	100.000±0.000
DSMN-Pre	100.000±0.000	97.818±0.155	**99.804±0.044**	100.000±0.000
DCMLN(ours)	**100.000±0.000**	91.498±0.308	99.073±0.097	**100.000±0.000**
	Load 3			
	Inner Race	Outer Race	Ball	Worst IOB
Finetune Last	82.953±1.192	77.240±1.370	94.544±0.790	82.402±1.248
Finetune Whole	99.684±0.167	98.469±0.183	**99.640±0.076**	98.556±0.164
Feature Knn	99.687±0.053	89.493±0.312	98.920±0.105	96.458±0.192
DSMN	99.956±0.019	91.582±0.289	95.767±0.226	100.000±0.000
DSMN-Pre	99.931±0.024	97.311±0.173	99.122±0.083	**100.000±0.000**
DCMLN(ours)	**99.996±0.022**	**98.493±0.154**	95.244±0.230	99.982±0.012



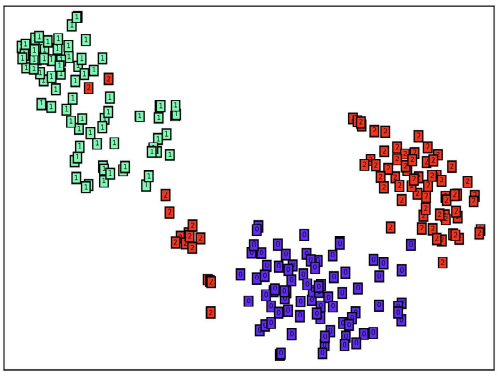
(a)



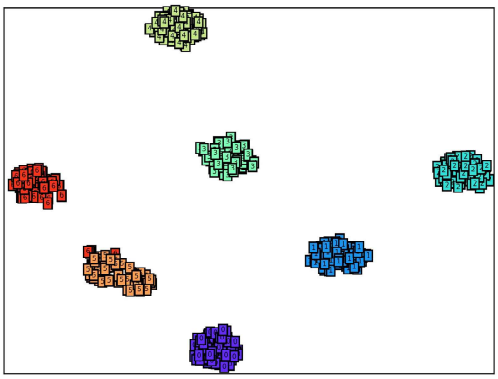
(b)



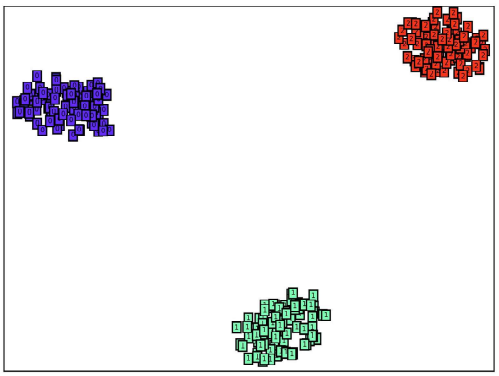
(c)



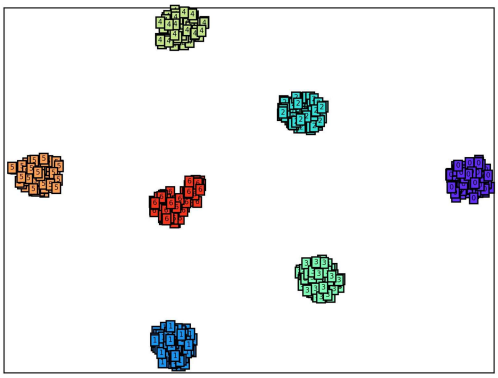
(d)



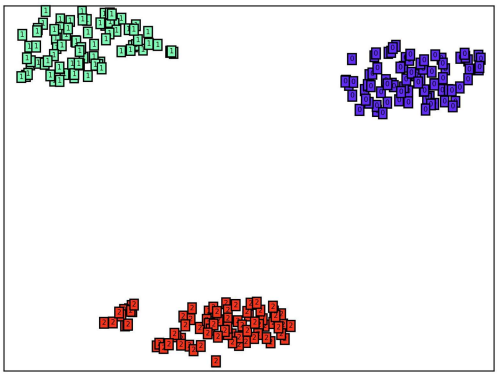
(e)



(f)

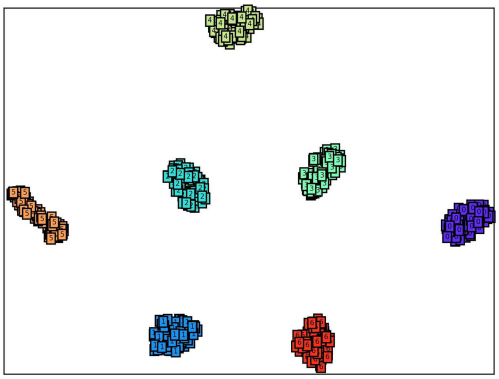


(g)

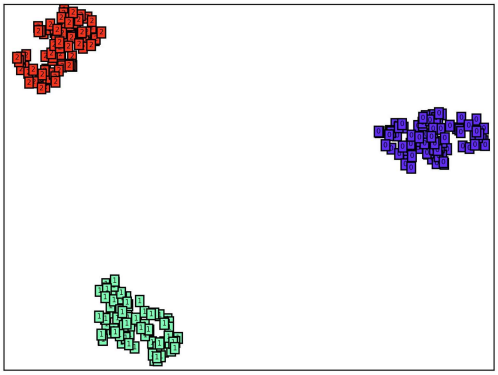


(h)

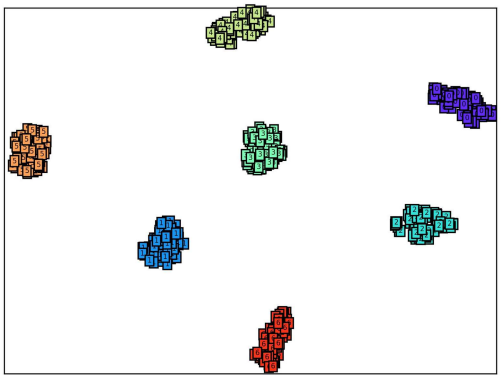
Fig. 4. t-SNE visualization of bearing diagnosis feature embedding derived from DCMLN (1-shot) under load 0. (a), (c), (e), and (g) denote the results of the inner race fault, outer race fault, ball fault and worst IOB from the task source domain, respectively; (b), (d), (f) and (h) denotes the results of the inner race fault, outer race fault, ball fault and worst IOB from the task target domain.



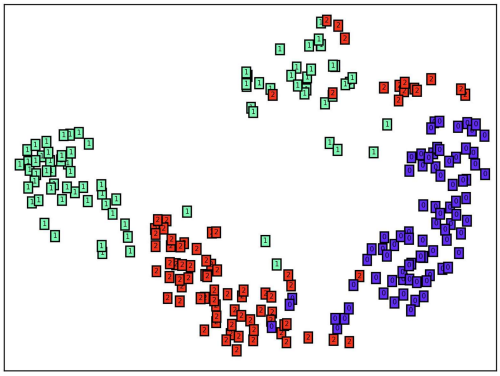
(a)



(b)



(c)



(d)

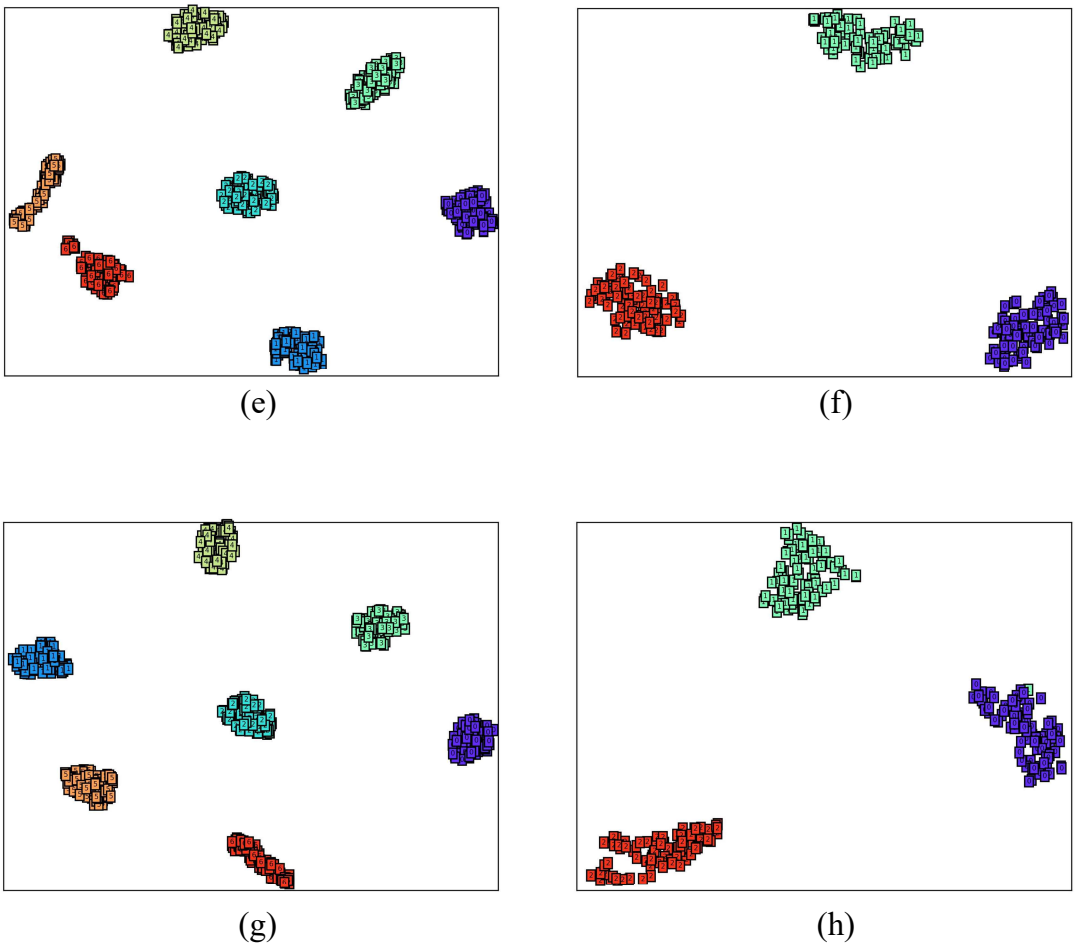


Fig. 5. t-SNE visualization of bearing diagnosis feature embedding derived from DSMN (1-shot) under load 0. (a), (c), (e) and (g) denotes the results of the inner race fault, outer race fault, ball fault, and worst IOB from the task source domain, respectively; (b), (d), (f) and (h) denotes the results of the inner race fault, outer race fault, ball fault and worst IOB from the task target domain.

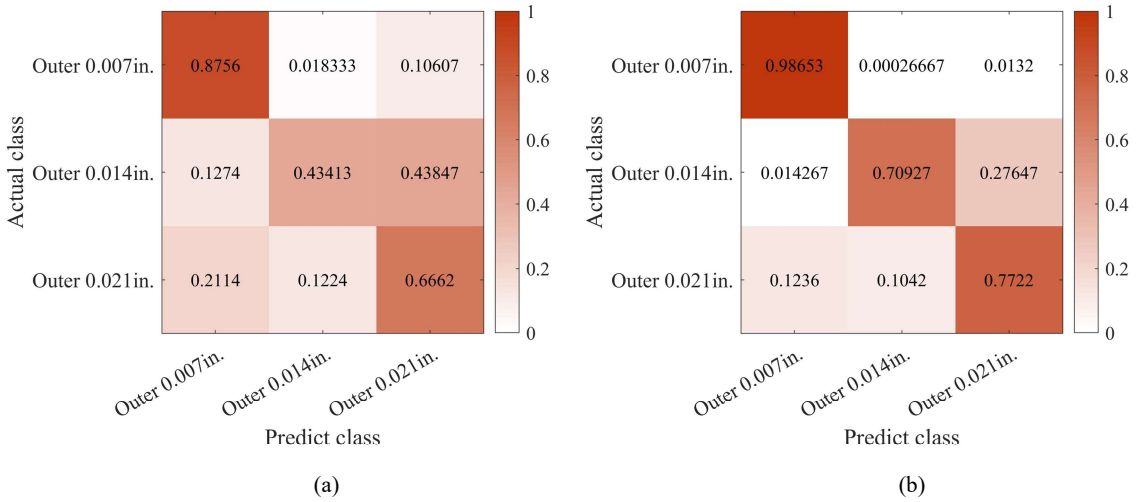


Fig. 6. The prediction accuracy of each class for outer race fault by DCMLN. (a) confusion matrix of 1-shot prediction, (b) confusion matrix of 10-shot prediction

T-SNE is applied to visualize the feature embedding of the source and target domain in two dimensions. Fig. 4 presents the t-SNE visualization of the bearing data feature embedding derived from DCMLN, where (a), (c), (e) and (g) denote the results of the inner race fault, outer race fault, ball fault, and worst IOB from the task source domain, respectively; (b), (d), (f) and (h) denote the results of four scenarios from the task target domain. As a comparison, Fig. 5 shows the t-SNE visualization of bearing data feature embedding derived from DSMN. In both figures, different colours denote different fault categories. From both figures, it can be seen that in the source domain, the embedding feature of a same fault category gathers in a cluster, while features from a different category are apart from each other, showing the DSMN model is well trained in the source domain. Meanwhile, in the target domain, the different categories for the inner race fault scenario in Fig. 4(b) and Fig. 5(b), the ball fault scenario in Fig. 4(f) and Fig. 5(f), and the worst IOB scenario in Fig. 4(h) and Fig. 5(h) are placed at a close area and is notably separated. However, in Fig. 4(d) and Fig. 5(d), at least two degrees of outer race fault scenario in the target domain are interfered with each other.

Considering that the accuracy for the outer race fault classification scenario is generally lower than that for the other three ones, the confusion matrix is provide show the prediction accuracy of each class for the outer race fault scenario by DCMLN. Fig. 6 (a) and (b) are confusion matrices of 1-shot and 10-shot fault classification, respectively, for the outer race fault classification scenario. As can be seen, the prediction accuracy is apparently improved when the number of shots increased. Moreover, the fault with size 0.007 in. can be clearly recognized, while the model was confused with fault size 0.014 and 0.021in.

4.2. Case Study 2: cylindrical roller bearing dataset

4.2.1 Data description and experimental setup

The cylindrical roller bearing fault test bench was set up as shown in Fig. 7, which

consisted of an electric generator, couplings, an AC motor, an intermediate shaft, bearing supports, and cylindrical roller bearings. In this case, ten cylindrical roller bearings (N406), with a pitch diameter of 59 mm, bore diameter of 30mm, roller diameter of 14mm, and 9 rollers, were utilized to produce 9 types of faults. One of the bearings was left with no fault and was considered as a reference bearing. Three kinds of faults, which were roller(ball), inner race and outrace faults were simulated by EDM to create certain degrees of scratches 100%, 60% and 30% in length from the edge respectively, each bearing had one fault, all faults were with 1mm in depth and 0.18 mm in width.

Data collection approximately took a month starting on 26/09/2010 and finishing on 29/10/2010. The bearing rig ran for five minutes for warming up before starting data collection. Each case of bearing conditions was tested three times to ensure that the signals obtained are consistent. All experiments were conducted under full speed and 50% of torsion load, however, the radial load varied between 0, 1, 2, 3, 4, and 5 respectively in every test, the sampling rate was 96 kHz and data length was 960000 points with a duration of 10 seconds.

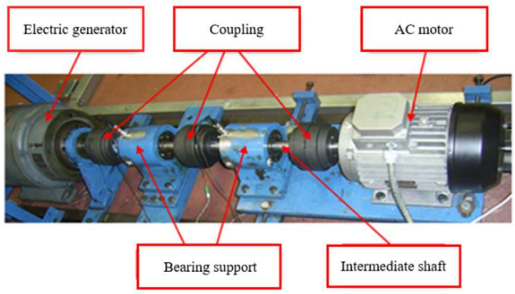


Fig. 7. The cylindrical roller bearing fault test bench

4.2.2 Results and analysis

The few-shot setup and detailed experimental settings are the same as those in case 1. The results of 1-shot and 10-shot learning fault diagnosis tasks are presented in Table 4 and Table 5, respectively. The proposed method, DCMLN, is compared with the other 5 models, including Finetune Last, Finetune Whole, Feature Knn, DSMN and DSMN-Pre.

In Table 4 and Table 5, all models perform better in 10-shot tasks than those in 1-shot tasks. Comparing the accuracy values of the six models, it is clear that the last three methods (DSMN, DSMN-Pre, and DCMLN) perform much better than the first three ones (Finetune Last, Finetune Whole, Feature Knn). An interesting result is that DSMN shows the best performance in both 1-shot and 10-shot tasks. Furthermore, the models (DSMN-Pre and DCMLN) that have pre-training process work not as well as the ones without pre-training.

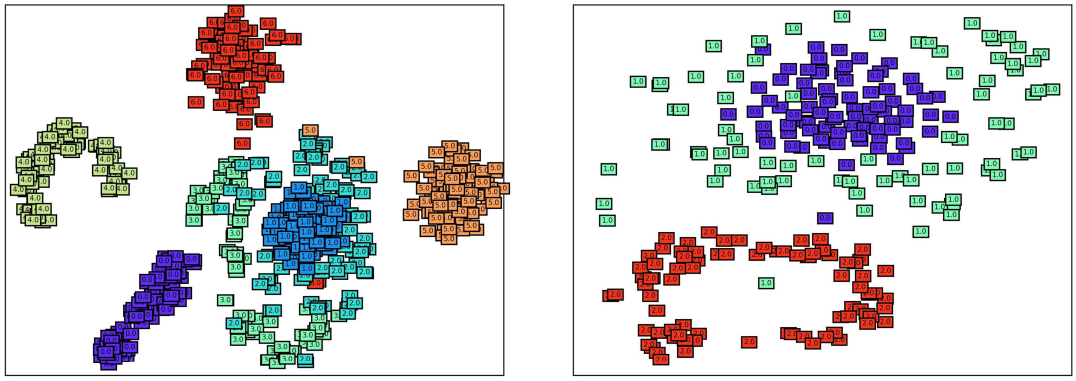
Table 4. Prediction accuracy (%) by DCMLN on 1-shot learning tasks for case 2 bearing fault classification

	Inner Race	Outer Race	Ball	Worst IOB
Finetune Last	51.953±1.146	54.156±1.236	42.209±0.971	40.518±0.768
Finetune Whole	62.833±0.979	60.998±0.958	51.378±0.938	42.818±0.822
Feature Knn	81.204±0.629	80.102±0.746	59.124±0.740	59.960±0.531

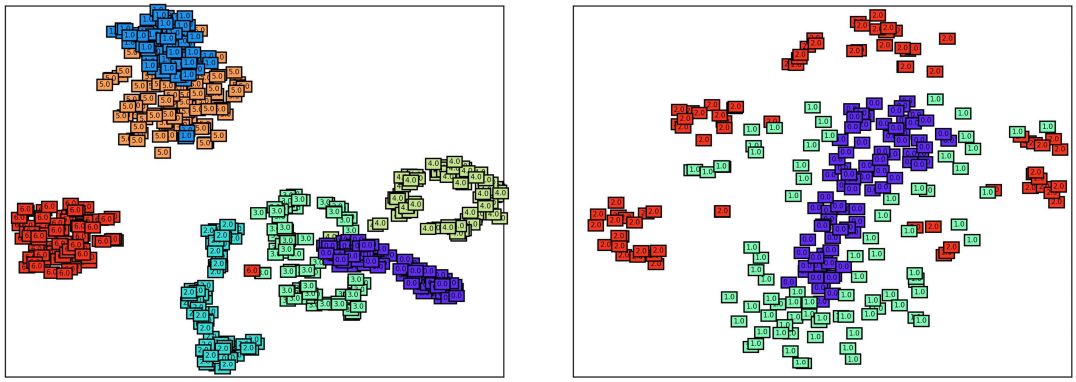
DSMN	**96.340±0.340**	**92.764±0.503**	**68.416±0.998**	61.973±0.474
DSMN-Pre	93.520±0.542	81.940±0.738	48.087±0.864	**64.664±0.436**
DCMLN(ours)	90.227±0.646	78.816±0.811	54.207±0.873	63.107±0.397

Table 5. Prediction accuracy (%) by DCMLN on 10-shot learning tasks for case 2 bearing fault classification

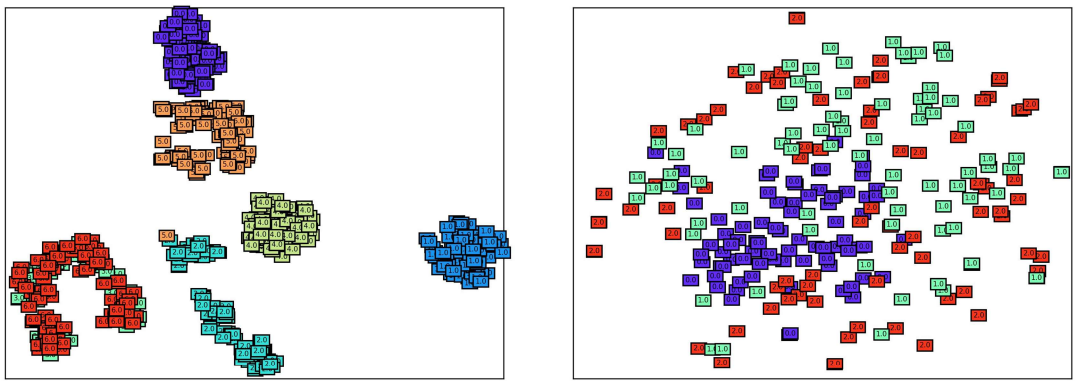
	Inner Race	Outer Race	Ball	Worst IOB
Finetune Last	67.604±0.668	63.151±0.787	47.031±0.705	46.560±0.682
Finetune Whole	85.771±0.504	77.411±0.605	61.018±0.599	58.047±0.553
Feature Knn	98.738±0.107	84.491±0.313	67.620±0.492	57.533±0.421
DSMN	**99.416±0.073**	**96.396±0.180**	**68.558±0.488**	**65.251±0.385**
DSMN-Pre	97.531±0.156	87.996±0.327	65.631±0.500	61.742±0.401
DCMLN (ours)	90.004±0.289	91.093±0.267	67.349±0.482	64.864±0.374



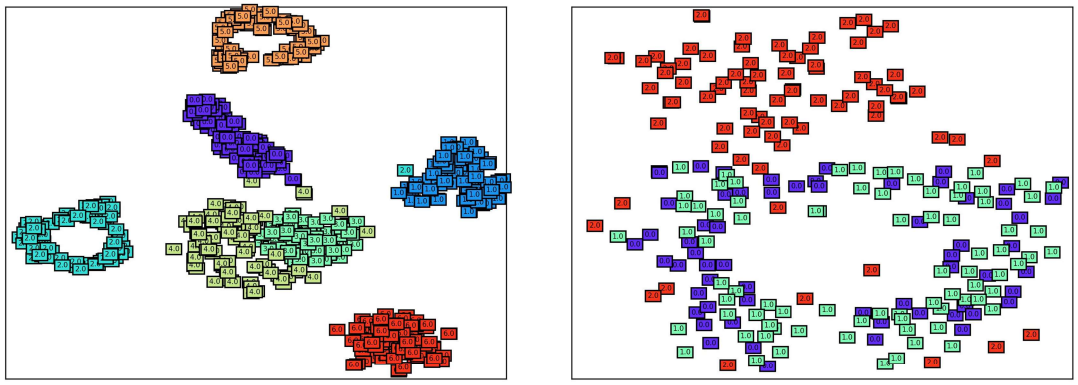
(a) (b)



(c) (d)

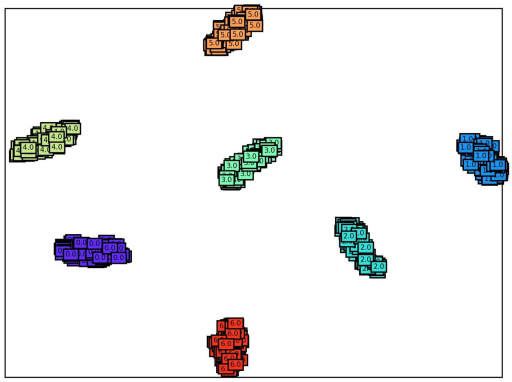


(e) (f)

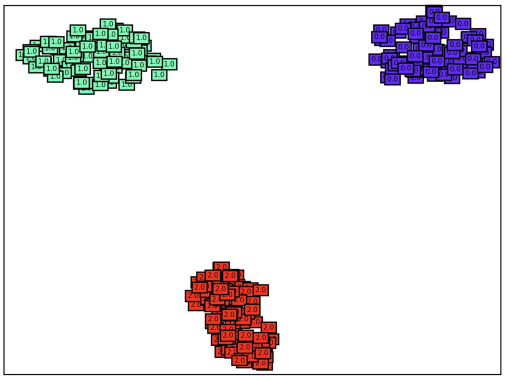


(g) (h)

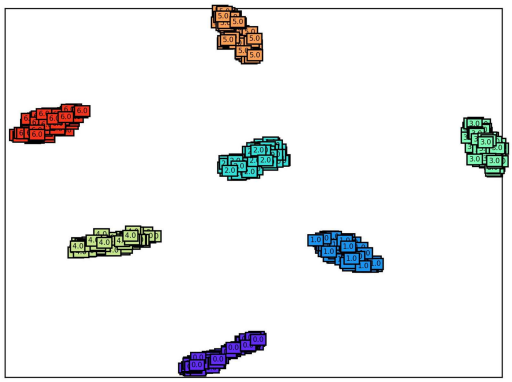
Fig. 8. t-SNE visualization of cylindrical roller bearing data feature embedding derived from DCMLN. (a), (c), (e) and (g) denotes the results of the inner race fault, outer race fault, ball fault and worst IOB from the task source domain, respectively; (b), (d), (f) and (h) denotes the results of the inner race fault, outer race fault, ball fault and worst IOB from the task target domain.



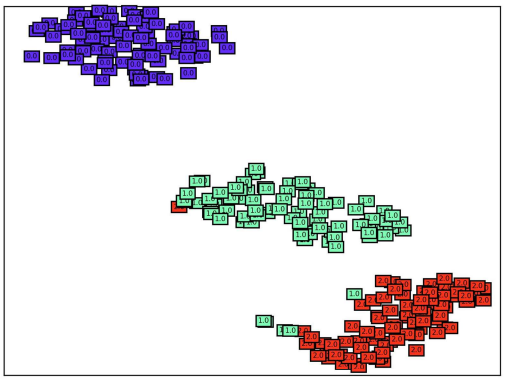
(a)



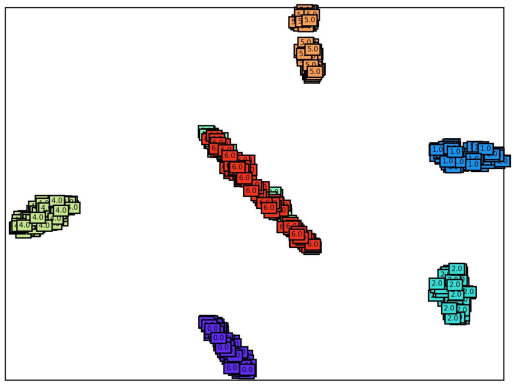
(b)



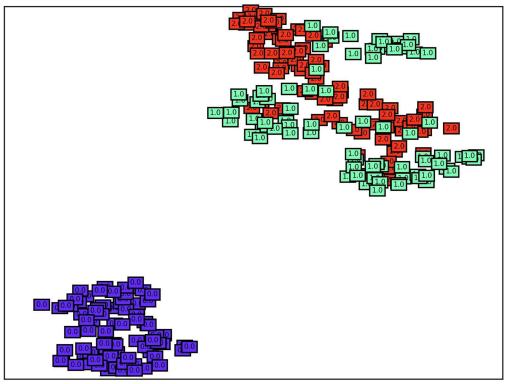
(c)



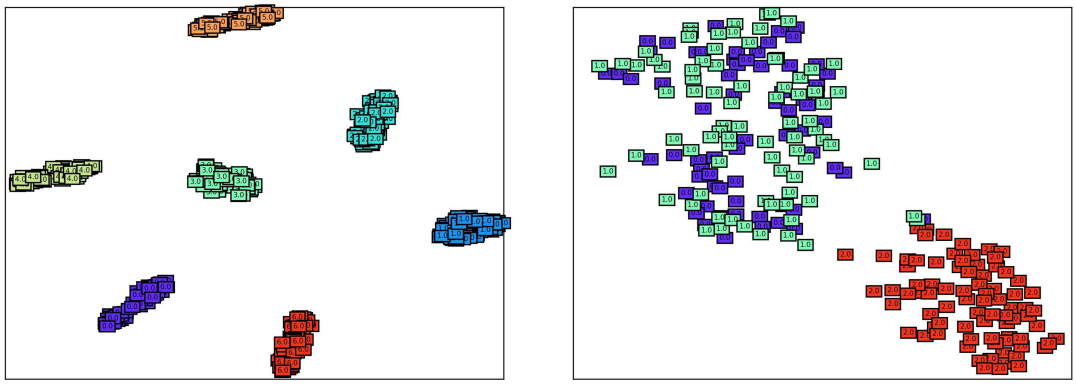
(d)



(e)

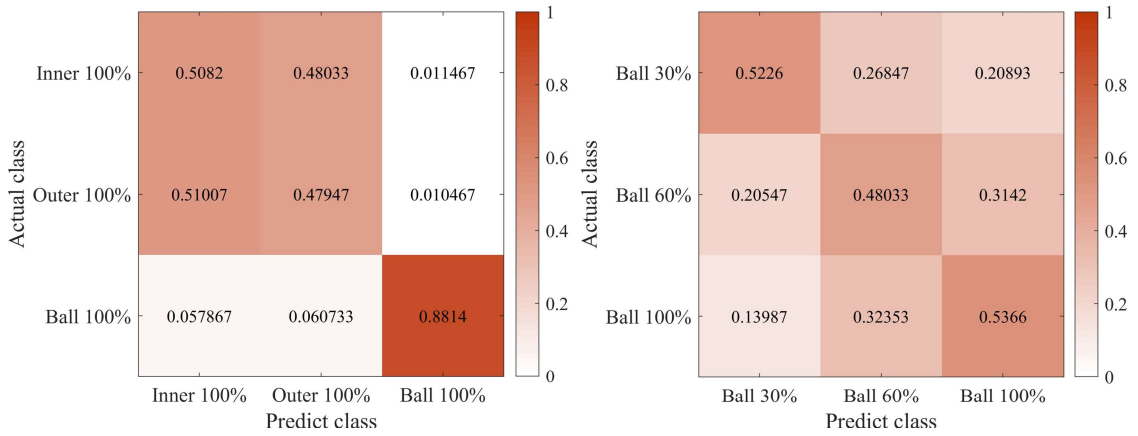


(f)



(g) (h)

Fig. 9. t-SNE visualization of cylindrical roller bearing data feature embedding derived from DSMN. (a), (c), (e) and (g) denotes the results of the inner race fault, outer race fault, ball fault and worst IOB from the task source domain, respectively; (b), (d), (f) and (h) denotes the results of the inner race fault, outer race fault, ball fault and worst IOB from the task target domain.



(a) (b)

Fig. 10. The prediction accuracy for each class by DCMLN. (a) confusion matrix of worst IOB prediction, (b) confusion matrix of ball faults prediction

From Table 4 and Table 5, all models have relatively lower accuracy in classifying the ball fault and worst IOB. Hence, t-SNE is applied to visualize the feature embedding for these two scenarios. Fig. 8 visualizes the t-SNE results derived from the DCMLN method, where (a), (c), (e), and (g) denote the results of the inner race fault, outer race fault, ball fault and worst IOB from the task source domain, respectively; (b), (d), (f) and (h) denotes those of the four scenarios from the task target domain. As a comparison, Fig. 9 visualizes the t-SNE result derived from the DSMN method, where (a), (c), (e), and (g) denote the results of the inner race fault, outer race fault, ball fault and worst IOB from the task source domain, respectively; (b), (d), (f) and (h) denotes the results of the four scenarios from the task target domain. Comparing these two figures, there exists some intersection between the features of both source and target with the DCMLN method, while most classes can be clearly separated by DSMN.

To better understand the prediction results by DCMLN, the confusion matrix is employed to show the classification accuracy for each class. As the ball fault and the worst IOB have the worst classification accuracy in all four scenarios, we present the confusion matrix of these two scenarios in Fig. 10 (a) and Fig. 10 (b), respectively. In the worst IOB scenario, the ball fault is clearly recognized while the model is confused with the inner and outer race fault. In the ball fault scenario, all three classes are not clearly distinguished.

5 Conclusion and discussion

To deal with fault diagnosis problems with limited labeled data, a hybrid method (DCMLN) that combines the merit of deep CNN supervised learning and meta-learning is proposed. The model is able to transfer the knowledge learnt from the source domain to deal with problems in the target domain. Experimental verifications are carried out on 1-shot and 10-shot tasks using the CWRU bearing dataset and a cylindrical roller bearing dataset. We compared our model with the other five models, which are "Finetune Last", "Finetune Whole", "Feature Knn", "Data Space Matching Network" and "Data Space Matching Network with Pre-training". The proposed DCMLN fault diagnostic method has very good performance on the CWRU dataset. Besides, diagnosis methods with pre-training have better performance than those without. However, from our experiment, pre-training does not always bring better results. When tested on the cylindrical roller bearing dataset in case 2, models with pre-training does not as good as those without pre-training. We believe it is because this task is more challenging to tackle. The weak performance of pre-training might be because the pre-training is not aware of the task of interest, resulting it fails to adapt. Google Brain Team [24] compared pre-training and self-training in the application of computer vision. Their experimental results demonstrated the limitation of pre-training and highlight the important advantages of self-training. Therefore, a future work is to replace pre-training with self-training.

Acknowledgment

This research was funded by RECLAIM project 'Remanufacturing and Refurbishment of Large Industrial Equipment' and received funding from the European Commission Horizon 2020 research and innovation programme under Grant Agreement No. 869884. The authors also acknowledge the support of The Efficiency and Performance Engineering Network International Collaboration Fund Award 2022 (TEPEN-ICF 2022) project "Intelligent Fault Diagnosis Method and System with Few-Shot Learning Technique under Small Sample Data Condition".

References

[1] A. Zhang, S. Li, Y. Cui, W. Yang, R. Dong, and J. Hu, 'Limited Data Rolling Bearing Fault Diagnosis With Few-Shot Learning', *IEEE Access*, vol. 7, pp. 110895–110904, 2019, doi: 10.1109/ACCESS.2019.2934233.

[2] Y. Lei, B. Yang, X. Jiang, F. Jia, N. Li, and A. K. Nandi, 'Applications of machine learning to machine fault diagnosis: A review and roadmap', *Mech. Syst. Signal Process.*, vol. 138, p. 106587, 2020.

[3] M. Zhang, D. Wang, W. Lu, J. Yang, Z. Li, and B. Liang, 'A deep transfer model with wasserstein distance guided multi-adversarial networks for bearing fault diagnosis under different working conditions', *IEEE Access*, vol. 7, pp. 65303–65318, 2019.

[4] J. Sun, X. Gu, J. He, S. Yang, Y. Tu, and C. Wu, 'A Robust Approach of Multi-sensor Fusion for Fault Diagnosis Using Convolution

Neural Network', *J. Dyn. Monit. Diagn.*, vol. 1, no. 2, Jun. 2022.

[5] X. Chen, M. Ma, Z. Zhao, Z. Zhai, and Z. Mao, 'Physics-Informed Deep Neural Network for Bearing Prognosis with Multisensory Signals', *J. Dyn. Monit. Diagn.*, vol. 1, no. 4, Jun. 2022.

[6] Z. Ren *et al.*, 'A novel model with the ability of few-shot learning and quick updating for intelligent fault diagnosis', *Mech. Syst. Signal Process.*, vol. 138, p. 106608, 2020.

[7] D. Wang, M. Zhang, Y. Xu, W. Lu, J. Yang, and T. Zhang, 'Metric-based meta-learning model for few-shot fault diagnosis under multiple limited data conditions', *Mech. Syst. Signal Process.*, vol. 155, p. 107510, Jun. 2021, doi: 10.1016/j.ymssp.2020.107510.

[8] Y. Wang, Q. Yao, J. Kwok, and L. M. Ni, 'Generalizing from a Few Examples: A Survey on Few-Shot Learning'. arXiv, Mar. 29, 2020. Accessed: Feb. 06, 2023. [Online]. Available: http://arxiv.org/abs/1904.05046

[9] E. G. Miller, N. E. Matsakis, and P. A. Viola, 'Learning from one example through shared densities on transforms', in *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, 2000, vol. 1, pp. 464–471.

[10] J. Lu, P. Gong, J. Ye, and C. Zhang, 'Learning from very few samples: A survey', *ArXiv Prepr. ArXiv200902653*, 2020.

[11] L. Fe-Fei, 'A Bayesian approach to unsupervised one-shot learning of object categories', in *proceedings ninth IEEE international conference on computer vision*, 2003, pp. 1134–1141.

[12] G. Koch, R. Zemel, and R. Salakhutdinov, 'Siamese neural networks for one-shot image recognition', in *ICML deep learning workshop*, 2015, vol. 2, no. 1.

[13] C. Finn, P. Abbeel, and S. Levine, 'Model-agnostic meta-learning for fast adaptation of deep networks', in *International conference on machine learning*, 2017, pp. 1126–1135.

[14] M. Al-Shedivat, T. Bansal, Y. Burda, I. Sutskever, I. Mordatch, and P. Abbeel, 'Continuous adaptation via meta-learning in nonstationary and competitive environments', *ArXiv Prepr. ArXiv171003641*, 2017.

[15] A. Antoniou, H. Edwards, and A. Storkey, 'How to train your MAML', *ArXiv Prepr. ArXiv181009502*, 2018.

[16] A. Nichol, J. Achiam, and J. Schulman, 'On first-order meta-learning algorithms', *ArXiv Prepr. ArXiv180302999*, 2018.

[17] C. Li, S. Li, A. Zhang, Q. He, Z. Liao, and J. Hu, 'Meta-learning for few-shot bearing fault diagnosis under complex working conditions', *Neurocomputing*, vol. 439, pp. 197–211, 2021.

[18] D. Wang, Y. Cheng, M. Yu, X. Guo, and T. Zhang, 'A hybrid approach with optimization-based and metric-based meta-learner for few-shot learning', *Neurocomputing*, vol. 349, pp. 202–211, 2019.

[19] O. Vinyals, C. Blundell, T. Lillicrap, and D. Wierstra, 'Matching networks for one shot learning', *Adv. Neural Inf. Process. Syst.*, vol. 29, 2016.

[20] J. Wu, Z. Zhao, C. Sun, R. Yan, and X. Chen, 'Few-shot transfer learning for intelligent fault diagnosis of machine', *Measurement*, vol. 166, p. 108202, Dec. 2020, doi: 10.1016/j.measurement.2020.108202.

[21] T. Hu, T. Tang, R. Lin, M. Chen, S. Han, and J. Wu, 'A simple data augmentation algorithm and a self-adaptive convolutional architecture for few-shot fault diagnosis under different working conditions', *Measurement*, vol. 156, p. 107539, May 2020.

[22] Radenović F, Tolias G, Chum O. CNN image retrieval learns from BoW: Unsupervised fine-tuning with hard examples[C]//Computer

Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14. Springer International Publishing, 2016: 3-20.

[23] B. Wang and D. Wang, 'Plant Leaves Classification: A Few-Shot Learning Method Based on Siamese Network', *IEEE Access*, vol. 7, pp. 151754–151763, 2019, doi: 10.1109/ACCESS.2019.2947510.

[24] B. Zoph *et al.*, 'Rethinking Pre-training and Self-training'. arXiv, Nov. 15, 2020. [Online]. Available: http://arxiv.org/abs/2006.06882