A Novel Information Entropy-Guided Source-Free Open-Set Fault Diagnosis Approach

Lanjun Wan, Hui Chen, Hongwei Tan, Jiaen Ning, Wei Ni, and Keqin Li, *Fellow, IEEE*

***Abstract*—In actual industrial applications, due to the data privacy protection, the source domain (SD) data are often not directly accessible, while the fault classes that do not belong to the SD often appear on the target domain (TD). To address these challenges, a novel information entropy-guided source-free open-set fault diagnosis (IESOFD) approach is proposed. First, a graph convolutional network (GCN)-based source model is pretrained to extract robust features. The pretrained source model is transferred to the TD to obtain the target model, which is fine-tuned to achieve the knowledge transfer from SD to TD. Second, a multigroup partition pseudo-label (MGPPL) strategy is designed. It initially partitions the TD samples into multiple groups with the pseudo-labels, selects some samples from each group as the known and unknown class sample sets, respectively, and minimizes and maximizes the information entropy of the known and unknown class sample sets, respectively, aiming to separate the unknown fault classes in the TD and improve the reliability of the pseudo-labels. Third, a difficulty class learning (DCL) strategy is designed, which reduces the intraclass distances of the difficulty classes and uses different weights to learn each fault class by information entropy, so as to more effectively separate the known and unknown fault classes. Finally, extensive experiments are conducted to verify the effectiveness of the proposed approach, and the results confirm that the proposed approach can accurately identify the known and unknown fault classes in the TD while protecting the SD data privacy.**

***Index Terms*—Fault diagnosis, information entropy, open-set, pseudo-labels, source-free.**

I. Introduction

THE stable operation of rotating machinery is a key factor in ensuring productivity, and the intelligent fault diagnosis (FD) plays a key role in ensuring the stable operation of rotating machinery [1]. From the perspective of measurement science, FD is essentially a process of intelligently measuring and identifying the health states of equipment. Traditional measurement methods rely on feature extraction and pattern recognition of sensor signals, while modern intelligent FD methods further transform these signals into high-dimensional features to achieve accurate "soft measurement" of fault classes. In recent years, the rotating machinery FD (RMFD) methods based on unsupervised domain adaptation (UDA) have attracted much attention. For instance, Qin et al. [2] developed a deep joint distribution alignment framework, improving the performance of cross-domain RMFD. Wang et al. [3] combined an improved deep reinforcement learning model and a correlation alignment metric, realizing the cross-domain compound FD tasks. Li et al. [4] designed a bi-generator cooperative domain-adversarial model that can generate better domain-invariant features, improving the RMFD accuracy. Ghorvei et al. [5] designed a deep subdomain adaptation method based on graph convolutional network (GCN), which better considers the geometric structure of sample features through GCN, improving the accuracy of RMFD. Sun et al. [6] designed a pseudo-label-guided loss to construct a more explicit fault classification boundary, improving the accuracy of cross-domain RMFD. Yu et al. [7] developed a multisource domain adaptation (DA) framework based on feature transfer and domain attributes, alleviating the negative transfer problem. Xie et al. [8] designed an improved prototype-oriented DA method, which can maximize the interclass distance and minimize the intraclass distance through a bi-directional prototype alignment mechanism. Chen et al. [9] studied a dual-adversarial-guided multisource DA network to enhance domain confusion, improving the performance of RMFD. Wan et al. [10] provided a multisource domain deep transfer learning method, increasing the cross-domain RMFD performance. Liu et al. [11] provided a multisource DA ensemble model based on dual-branch feature matching

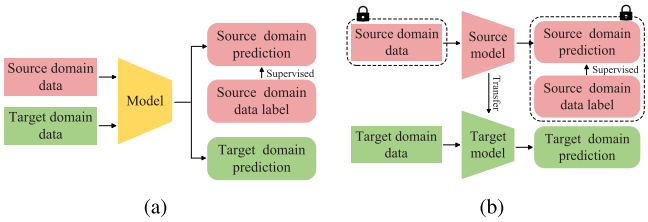


Fig. 1. Comparison of (a) UDA and (b) SFDA.

Received 8 January 2026; revised 19 March 2026; accepted 4 April 2026. Date of publication 27 April 2026; date of current version 4 May 2026. This work was supported in part by the Scientific Research Foundation of Hunan Provincial Education Department under Grant 24A0391; in part by the Natural Science Foundation of Hunan Province under Grant 2026JJ50235, Grant 2025JJ70030, and Grant 2023JJ30217; in part by the National Natural Science Foundation for Young Scientists of China under Grant 61702177; and in part by the Open Project of Hunan Key Laboratory of Intelligent Information Perception and Processing Technology under Grant 2025KF03. The Associate Editor coordinating the review process was Dr. Hongtian Chen. *(Corresponding author: Lanjun Wan.)*

Lanjun Wan, Hui Chen, Hongwei Tan, and Wei Ni are with the School of Computer Science and Artificial Intelligence, Hunan University of Technology, Zhuzhou 412007, China (e-mail: wanlanjun@hut.edu.cn; chenhui@stu.hut.edu.cn; tanhongwei@stu.hut.edu.cn; niwei@hut.edu.cn).

Jiaen Ning is with the School of Electronic Information, Central South University, Changsha 410075, China (e-mail: jiaenning@csu.edu.cn).

Keqin Li is with the Department of Computer Science, The State University of New York, New Paltz, NY 12561 USA (e-mail: lik@newpaltz.edu).

Digital Object Identifier 10.1109/TIM.2026.3687340

that can dynamically adjust the marginal and conditional distributions, reducing the issue of negative transfer. Although the existing UDA approaches perform well in RMFD, they generally assume that both source domain (SD) and target domain (TD) data are directly accessible during model training, as shown in Fig. 1(a). In actual industrial scenarios, due to the scarcity of fault samples and the high cost of labeling, it is usually difficult for a single enterprise to accumulate enough high-quality fault samples. For this reason, some enterprises may require the third-party enterprise to provide enough fault samples as the SD for model training. However, due to the limitations in data security and privacy protection, it is usually not possible to directly access the SD data provided by the third-party enterprise. This means that the SD data are not accessible. As a result, the promotion of traditional UDA approaches in practical industrial applications is limited. To solve this problem, the third-party enterprise utilizes the local SD data to pretrain a source model. Subsequently, it provides the pretrained model rather than SD data to those enterprises in need, meaning that there is no need to directly access the SD data. Therefore, the source-free DA (SFDA) approaches have been explored, as depicted in Fig. 1(b).

To address this issue, researchers have explored the SFDA approaches for RMFD. In the SFDA approaches, a source model is usually pretrained using the SD data provided by the third-party enterprise and transferred to the TD, and then the model is fine-tuned through the TD data to obtain a target model that can perform FD tasks well. For example, Jiao et al. [12] proposed an SFDA method, which uses the normalized symmetric loss to improve the quality of pseudo-labels, achieving superior FD performance. Tian et al. [13] designed a source model based on a deep generalized neural network, which eliminates the dependence on the SD data by using a flow mixing module, solving the overfitting problem. Cao et al. [14] constructed a two-stage SFDA framework, which can better extract transferable features by combining local semantic constraint restrictions and global adversarial feature alignment. Wang et al. [15] constructed an improved feature extractor that can extract transferable features, enhancing the generalization ability of the model. Li et al. [16] designed a filter that can improve the accuracy of pseudo-labels using the Hausdorff distance, increasing the FD accuracy of rotating machinery. Liu et al. [17] devised an improved information entropy strategy for effectively estimating the uncertainty of pseudo-labels, realizing a robust RMFD. Liu et al. [18] designed a multisource SFDA method based on graph structure learning to capture and propagate the potential structure relationships of sample features through a graph neural network, improving the accuracy of cross-domain FD. Yu et al. [19] studied a contrastive learning-based strategy for reducing the effect of pseudo-label noises, boosting the performance of RMFD. Gao et al. [20] designed a self-training mechanism in the multisource SFDA for enhancing the learning of the TD samples, raising the recognition ability of the model. Lin et al. [21] investigated an SFDA method with the uncertainty-based metric to calibrate the risk samples at a fine-grained level, improving the accuracy and efficiency of RMFD. Ma et al. [22] studied a prototype-based pseudo-labeling strategy that can align the source and TD features, improving the FD performance of the model. Li et al. [23] put forward a calibrated SFDA method, which combines the samples with pseudo-labels and the determinant mutual information loss for the model adaptation, effectively improving the accuracy of RMFD. Jiao et al. [24] proposed a source-free black-box adaptation method, which utilizes the decoupled self-distillation to better transfer the SD knowledge to the TD, significantly enhancing cross-domain FD performance. The above studies show that the existing SFDA methods can effectively deal with RMFD in the scenario where the SD data cannot be directly accessed. In the field of instrumentation and measurement, the SFDA methods have strong practical significance. Because they can realize the transfer of diagnosis knowledge without directly accessing the original measurement data. These methods meet the requirements of the measurement system for data security and privacy protection in the industrial field.

To address the issues of the scarcity of fault samples and the high cost of labeling, recently, there have been some studies on the simulation data-driven DA in the field of RMFD. Xia et al. [25] constructed a dynamically updated digital twin model to generate the simulated fault data close to actual operating conditions, which are used as the SD data, effectively achieving accurate cross-domain FD. Kumar et al. [26] designed a digital twin-enabled DA framework that can transfer knowledge from the simulation domain (i.e., SD) to the real domain (i.e., TD), improving the accuracy and generalization of RMFD. Fang et al. [27] developed a digital twin-assisted DA network to improve the performance of RMFD, where the distribution discrepancies between the simulation and real domains are reduced significantly by the dynamic joint distributed DA strategy. The above approaches can effectively address the problem of scarce SD samples in DA by generating massive simulation samples that cover different fault classes. However, the construction of simulation models heavily relies on expert knowledge and precise physical parameters, resulting in poor scalability and high modeling costs. In contrast, SFDA exhibits stronger practicality and robustness, it does not rely on the specific simulation model and only utilizes the pretrained source model and unlabeled TD data to achieve the fine-tuning of the target model.

Both the above UDA and SFDA methods belong to the closed-set DA (CSDA) methods, which are applicable to RMFD under the closed-set scenarios and require the SD and TD to have the same label space. However, in practical measurement applications, sensor signals are often affected by factors such as changes in working conditions, which may lead to the emergence of fault classes that are not recorded during the training stage. This results in the label spaces of the SD and TD generally being heterogeneous, meaning that the TD often contains fault classes that do not belong to the SD, which is called the open-set scenario. Therefore, how to effectively address the open-set FD (OSFD) is a more challenging problem. OSFD is of great significance for enhancing the practicability and reliability of the measurement system.

To solve the OSFD problem, some researchers have recently conducted studies on the open-set DA (OSDA). For example, Liang et al. [28] utilized the pseudo-margin vector to handle the unknown fault classes in the TD, improving the accuracy of multisource domain RMFD. Mei et al. [29] effectively solved the OSFD problem of rotating machinery by separating the unknown fault classes through the extreme value theory and empirical threshold. Wang and Zhang [30] proposed a 1D-ResNet-based OSFD method, which can separate unknown samples through the cosine similarities and Mahalanobis distances between the input sample features and the class feature centers. Zhu et al. [31] constructed a multiadversarial learning-based DA model that can align features from different perspectives, improving the performance of OSFD. Weng et al. [32] designed a dynamic open-set subdomain adaptive module, which can separate unknown fault classes and better align known fault classes. Jin et al. [33] combined adversarial learning with the weighted DA network, which can effectively separate the private class features and accurately identify unknown fault classes. Pan et al. [34] designed an adaptive unknown fault separation scheme for increasing the RMFD accuracy by separating the unknown classes through a hyperplane. Zheng et al. [35] proposed a multiadversarial OSDA network to mitigate the negative transfer effect through adversarial learning. Weng et al. [36] designed an outlier pseudo-label ensemble filter that can separate the unknown classes through the high-quality soft labels, alleviating the uncertainty of pseudo-labels. Li et al. [37] studied a centroid contrastive multisource DA method, which can effectively alleviate the domain shift and class shift by combining the centroid contrastive and domain discrimination losses. Wang et al. [38] presented a self-supervised cross-domain OSFD method that uses the high-quality unknown fault samples as the supervisory information, improving the performance of RMFD. The above research can effectively address the challenge of unknown fault classes in the TD and provide useful exploration for the OSFD problem. However, most existing OSFD methods still rely on the direct access of the SD and have not fully taken into account the data isolation and privacy constraints commonly found in industrial measurement scenarios. How to implement OSFD without directly accessing the SD is an important challenge faced by intelligent FD in the current measurement field.

In summary, the existing SFDA approaches perform well when facing RMFD in the scenario where the SD data cannot be directly accessed, but they pay less attention to the problem of the unknown fault classes appearing on the TD. Although the existing OSFD approaches solve the issue of the unknown fault detection in the TD to some extent, they have not considered how to effectively identify the known and unknown classes without directly accessing the SD data. However, in actual RMFD, the following challenges are often encountered simultaneously. First, the SD data are directly inaccessible because of the data privacy protection. Second, the TD contains the fault classes that do not belong to the SD. When the SD data cannot be directly accessed, it is difficult to accurately identify known and unknown fault classes on the TD. The core reason is the inability to obtain the knowledge, such as feature boundaries and confidence thresholds to distinguish known and unknown classes, when SD data are inaccessible. In addition, when the SD data cannot be directly accessed, the unknown fault samples are more likely to be misclassified as known classes. The most existing OSDA approaches rely on the SD data to establish decision boundaries between known and unknown classes, whereas these decision boundaries will be difficult to build when the SD data are inaccessible. Based on the above challenges, there is an urgent need for a unified approach that can address both the SD data cannot be directly accessed, and the unknown fault classes appear on the TD. Therefore, a novel information entropy-guided source-free OSFD (IESOFD) approach is proposed to address the above challenges. The proposed approach can still achieve reliable measurement and identification of the health states of target equipment without direct access to the SD, making it particularly suitable for complex measurement scenarios in industrial settings that have high data privacy requirements and open fault modes.

The main contributions of this article are as follows.

1) An IESOFD framework is proposed, which can effectively realize the cross-domain RMFD under the scenario where the SD data cannot be accessed directly and the unknown fault classes appear on the TD. In the framework, the pretrained GCN-based source model is transferred to the TD to obtain the target model, which is fine-tuned to achieve the knowledge transfer from SD to TD.
2) A multigroup partition pseudo-label (MGPPL) strategy is designed. In this strategy, the pseudo-labels of the TD samples are generated according to the prediction results obtained from the target model. The TD samples are initially partitioned into multiple groups with the pseudo-labels. Some samples are selected from each group as the known and unknown class sample sets, respectively, which can effectively realize the multigroup partitioning of known and unknown classes in the TD. Meanwhile, the information entropy of the known and unknown class sample sets are minimized and maximized, respectively, which can effectively separate the unknown fault classes in the TD and improve the reliability of pseudo-labels.
3) A difficulty class learning (DCL) strategy is designed. In this strategy, the intraclass distances of the difficulty classes with larger differences in information entropy are reduced. The different weights are employed to learn each fault class according to information entropy. This strategy can achieve more accurate identifications of known and unknown classes.
4) Extensive experiments are performed on the bearing and gearbox datasets. The results prove that the proposed approach gains superior FD performance under the scenario where the SD data cannot be directly accessed, and the unknown fault classes appear on the TD.

The remainder of this study is organized as follows. Section II introduces the basic theory. Section III describes

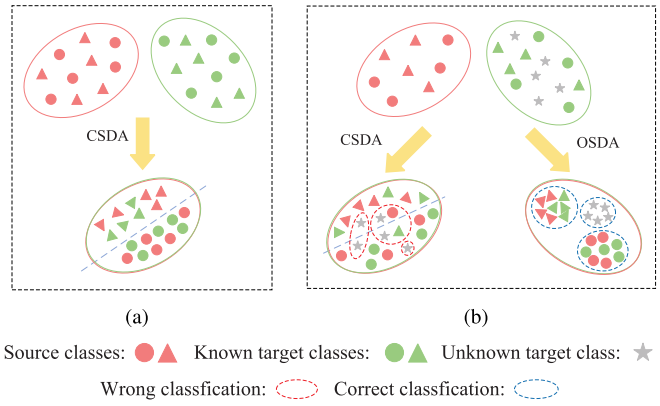


Fig. 2. Illustrations of (a) CSDA and (b) OSDA.

the proposed approach. Section IV presents the experimental results and analysis. Section V provides the conclusions.

II. Basic Theory

A. Domain Adaptation

The DA is intended for addressing the domain shift issue by transferring knowledge from the labeled SD to the unlabeled TD. The DA draws the labeled SD data and the unlabeled TD data for realizing the distribution alignment between SD and TD. Most of the existing DA methods belong to the CSDA methods assuming that the SD and TD have the same label space, as shown in Fig. 2(a). Although the SD and TD have the same label space, their feature distributions are discrepancy, thus the CSDA methods require alleviating the negative transfer problem caused by domain shift. The existing CSDA methods usually mitigate the domain shift effectively through the explicit or implicit distribution alignment.

However, the TD often has the classes that do not appear on the SD in practical applications. This scenario, in which the unknown classes appear, is called the open-set scenario. The label space of SD is just a subset of that of TD in the open-set scenario. If the CSDA methods are applied in the open-set scenario, the features of known classes in the SD and those of known and unknown classes in the TD will inevitably be misaligned, which causes the unknown classes to mix in the different known classes, thus generating classification errors. In view of this, the OSDA methods are suitable for the open-set scenario, as they can effectively handle the distribution discrepancies between different domains as well as the SD and TD with different label spaces. The OSDA aims to correctly classify the TD samples into the known and unknown classes, as shown in Fig. 2(b).

B. Graph Convolutional Network

The GCN is a convolutional neural network that can act directly on an undirected graph and utilize its structural information. A graph G is generally defined by

$$G = (V, E) \tag{1}$$

where V and E denote the node and edge sets, respectively. $V = \{v_1, v_2, \ldots, v_n\}$ is the set of n nodes. $e_{ij} = (v_i, v_j)$ is the edge connected between nodes v_i and v_j. The adjacency matrix $A \in \mathbb{R}^{n \times n}$ indicates the relationships between nodes and edges. If there exists an edge between v_i and v_j, $A_{ij} = 1$; otherwise, $A_{ij} = 0$. The node feature vector matrix $X \in \mathbb{R}^{n \times d}$, where d is the node feature dimension. The propagation between graph convolution (GConv) layers in the GCN is defined by

$$H^{(l+1)} = \mathrm{ReLU}\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right). \tag{2}$$

In (2), $H^{(l)}$ and $W^{(l)}$ are the node feature matrix and the trainable weighting matrix of the lth layer, respectively, ReLU$(\cdot)$ indicates the rectified linear unit activation function, $\tilde{A}$ indicates the adjacency matrix with added self-connections, and $\tilde{D}$ denotes the degree matrix of $\tilde{A}$. $\tilde{A}$ and $\tilde{D}_{ii}$ can be calculated by

$$\tilde{A} = A + I_n \tag{3}$$

and

$$\tilde{D}_{ii} = \sum_j \tilde{A}_{ij} \tag{4}$$

respectively, where I_n is the identity matrix. The softmax function is adopted for realizing the node classification after the propagation of multiple layers, and the final output graph convolutional signal matrix Z is described as

$$Z = f(X, A) = \mathrm{softmax}\left(\hat{A}\mathrm{ReLU}\left(\hat{A}XW^{(0)}\right)W^{(l)}\right) \tag{5}$$

where $\hat{A} = \tilde{D}^{-(1/2)}\tilde{A}\tilde{D}^{-(1/2)}$ is the symmetrically normalized adjacency matrix.

III. Proposed Approach

A. Problem Definition

In the FD scenario, where the SD data are not accessible and the unknown fault classes appear on the TD, first, a source model M_s is trained through the labeled SD data $D_\mathrm{s} = \{x_i^\mathrm{s}, y_i^\mathrm{s}\}_{i=1}^{n_\mathrm{s}}$, where x_i^s and y_i^s denote the ith sample in D_s and its corresponding label, respectively, and n_s is the number of samples in D_s. Second, M_s is transferred to the TD and used as the target model M_t. Finally, the unlabeled TD data $D_\mathrm{t} = \{x_i^\mathrm{t}\}_{i=1}^{n_\mathrm{t}}$ are utilized to fine-tune M_t to obtain the final target model, where x_i^t denotes the ith sample in D_t, and n_t denotes the number of samples in D_t. During the training of M_t and D_s are unavailable. Not only there are distribution discrepancies between D_s and D_t, but D_s and D_t have different label spaces. The fault classes not found in D_s might appear on D_t. The C_k indicates the fault classes that are common to both D_s and D_t, which are referred to as the known fault classes. The C_u denotes the fault classes that exist in D_t but not in D_s, which are called the unknown fault classes. The goal is to train M_t that can accurately recognize the C_k and C_u.

B. Design of IESOFD Framework

To effectively perform cross-domain RMFD under the scenario where the SD data cannot be directly accessed and the TD has the unknown fault classes, an IESOFD framework is designed. Fig. 3 depicts the overall design of the IESOFD framework. The framework mainly includes three steps: data acquisition and sample division, source model pretraining, and target model fine-tuning. The details are as follows.

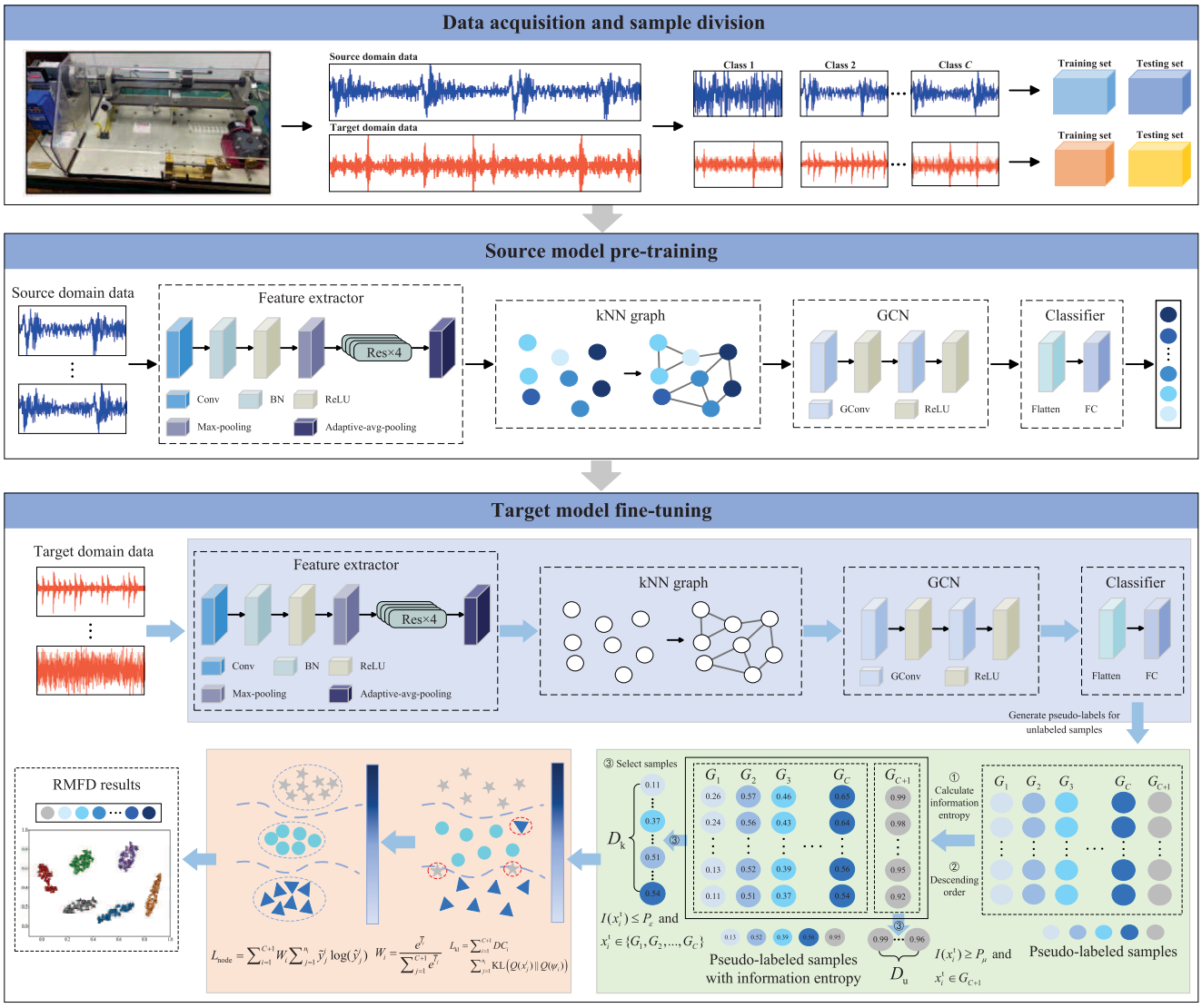


Fig. 3. Proposed IESOFD framework.

Step 1: Data acquisition and sample division. First, the raw vibration signals are acquired in real time by multiple sensors. Second, the raw vibration signals are segmented at the constant length and divided into the D_s and D_t, where D_s need to be labeled and D_t do not need to be labeled. Third, D_s and D_t are normalized by the Z-score, which is beneficial for improving the initialization and convergence speed of the model. Finally, D_s and D_t are split into the training and testing sets, respectively.

Step 2: Source model pretraining. The source model M_s is trained with D_s to classify the SD samples. First, each batch of samples from D_s are input into a feature extractor consisting of a convolutional (Conv) layer, a batch normalization (BN) layer, a max-pooling layer, four residual (Res) blocks, and an adaptive-avg-pooling layer. The features of the samples obtained via the feature extractor are used as the features of the nodes. The node set V is constructed in the form of one sample corresponding to one node. Second, the k-nearest neighbor (kNN) algorithm is adopted for constructing the kNN graph. Determining whether the nodes are connected or not according to the Euclidean distances between nodes. The Euclidean distance between nodes is calculated by

$$D_{ij} = \sqrt{\sum_{k=1}^{d} \left(v_{ik} - v_{jk}\right)^2} \tag{6}$$

where D_{ij} denotes the distance between the nodes v_i and v_j, v_{ik} and v_{jk} represent the kth feature values of v_i and v_j, respectively. The adjacency matrix A is constructed according to the relationships between nodes. If v_i is one of the K nearest neighbor nodes of v_j, $A_{ij} = 1$; otherwise, $A_{ij} = 0$. Third, the A and the node feature vector matrix X are input into the GCN consisting of two GConv layers and two ReLU activation functions to update the node features. Finally, the classification results are obtained by a classifier consisting of one flatten layer and one fully connection (FC) layer. The source model is iteratively updated according to the cross-entropy loss to obtain the final source model M_s.

Step 3: Target model fine-tuning. The M_s is transferred to the TD and used as the target model M_t. M_t is fine-tuned with D_t to obtain the final target model, so as to achieve

accurate identifications of C_k and C_u. First, each batch of samples from D_t is input into M_t, and the proposed MGPPL strategy is used to generate the pseudo-labels of the samples and classify the samples into the known and unknown class sample sets. Second, the information entropy loss is calculated by minimizing the information entropy of the known class sample set and maximizing the information entropy of the unknown class sample set. Third, the proposed DCL strategy is used to better aggregate the samples of the same class, and the intraclass distances are reduced by calculating the Kullback–Leibler (KL) divergence loss and the node classification loss. Fourthly, both the node similarity and the structural similarity are considered to calculate the structural consistency loss. Finally, the total loss is calculated based on the above four losses and M_t is iteratively updated accordingly.

Algorithm 1 Proposed MGPPL Strategy

Input: The TD data D_t, the target model M_t, the threshold ω, and the parameters P_ε and P_μ.
Output: The information entropy loss L_{ie}.

```
Get a batch of B samples from D_t;
for i = 1 to B do
    Get M_t(x_i^t) by M_t;
    Generate ỹ_i^t by Eq. (9);
end for
Partition the pseudo-labeled samples into {G_1, G_2, ..., G_C, G_{C+1}};
for j = 1 to C + 1 do
    Calculate the information entropy of each sample in G_j by Eq. (10);
    Sort all the samples in G_j in descending order by information entropy;
    if 1 ≤ j ≤ C then
        Select each sample whose information entropy is less than or equal to P_ε from G_j into D_k;
    else
        Select each sample whose information entropy is greater than or equal to P_μ from G_j into D_u;
    end if
end for
Calculate L_ie by Eq. (13);
```

C. Design of MGPPL Strategy

In the open-set scenario, the unknown classes may appear on the TD, and it is difficult to determine whether the TD samples belong to C_k or C_u only through the initial target model. Therefore, the MGPPL strategy is proposed, as shown in Fig. 4. The MGPPL strategy described in Algorithm 1 will be executed during each training epoch of the target model M_t. The strategy mainly consists of a phase of generating pseudo-labels for the TD samples and a phase of multigroup partitioning. First, M_t is utilized for generating the pseudo-labels for the TD samples to initially classify all the samples. Second, the multigroup partitioning is used to select the samples to form the known class sample set D_k and the unknown class sample set D_u. Finally, the information entropy loss is calculated to separate the C_u and improve the reliability of pseudo-labels.

In the phase of generating the pseudo-labels for the TD samples, first, the M_s is transferred to the TD and used as the M_t. Second, each batch of samples from D_t are input into M_t, and the class prediction probability vector of the TD samples x_i^t obtained by M_t is expressed as

$$M_t\left(x_i^t\right) = \left(S_1^i, S_2^i, \ldots, S_C^i\right) \tag{7}$$

where S_j^i indicates the probability of x_i^t belonging to the jth class predicted by M_t, and $1 \le j \le C$. Third, the maximum class prediction probability $S_{\max}^i$ for x_i^t is defined as

$$S_{\max}^i = \max\left(S_1^i, S_2^i, \ldots, S_C^i\right). \tag{8}$$

Finally, the pseudo-label is generated for x_i^t based on $S_{\max}^i$. When the $S_{\max}^i$ of x_i^t is greater than or equal to the threshold ω, the label of the fault class corresponding to $S_{\max}^i$ is generated for x_i^t. Otherwise, the label of the fault class $C+1$ is generated for x_i^t, where $C+1$ is called the unknown fault class. The generation of the pseudo-label $\tilde{y}_i^t$ of x_i^t can be described as

$$\tilde{y}_i^t = \begin{cases} \arg\max_{j\in\{1,2,\ldots,C\}} S_j^i, & \text{if} S_{\max}^i \ge \omega; \\ C+1, & \text{otherwise.} \end{cases} \tag{9}$$

In the phase of multigroup partitioning, first, the samples with pseudo-labels are partitioned into $C+1$ groups $\{G_1, G_2, \ldots, G_C, G_{C+1}\}$, where G_j denotes the group of all the samples belonging to the jth known fault class and $1 \le j \le C$, and G_{C+1} denotes the group of all the samples belonging to the unknown fault class. Second, for each group, the information entropy of each sample in the group is calculated, and all samples in the group are sorted from high to low according to the information entropy. The information entropy $I(x_i^t)$ of x_i^t is calculated by

$$I\left(x_i^t\right) = -\sum_{j=1}^{C} S_j^i \log S_j^i \tag{10}$$

where the value range of $I(x_i^t)$ is $[0, \log C]$. When the probability that x_i^t belongs to a certain class is 1 and the probability that x_i^t belongs to other classes is 0, the minimum value of $I(x_i^t) = -(1\cdot\log 1 + 0\cdot\log 0 + \ldots + 0\cdot\log 0) = 0$. When the probability that x_i^t belongs to each class is the same, i.e., $\forall S_j^i = (1/C)$, the maximum value of $I(x_i^t) = -((1/C)\cdot\log(1/C) + (1/C)\cdot\log(1/C) + \ldots + (1/C)\cdot\log(1/C)) = \log C$. The information entropy of a sample can indicate whether the sample belongs to C_k or C_u to some extent. In general, the information entropy of a sample belonging to C_k is lower, whereas the information entropy of a sample belonging to C_u is higher. Third, the samples in each group of $\{G_1, G_2, \ldots, G_C\}$ whose information entropy is less than or equal to P_ε are selected to form the known class sample set D_k, where P_ε is a percentile and $50 \le \varepsilon \le 100$. In other words, the bottom $(100-\varepsilon)\%$ of samples in each group of $\{G_1, G_2, \ldots, G_C\}$ are selected after being sorted in descending order of information entropy. Fourthly, the samples in G_{C+1} whose information entropy is greater than or equal to P_μ are selected to form the unknown class sample set D_u, where P_μ is a percentile and

$1 \le \mu \le 50$. That is to say, the top $\mu\%$ of samples in G_{C+1} are selected after being sorted in descending order of information entropy. If all the TD samples from D_t are sorted from high to low according to the information entropy, all the samples with information entropy less than or equal to P_ε are selected as D_k, and all the samples with information entropy greater than or equal to P_μ are selected as D_u, it will lead to the fact that the samples in D_k tend to be concentrated in a few fault classes, resulting in an uneven distribution of the samples. This can also lead to too many samples in D_u, which can easily cause the over-fitting problem in model training, thus reducing the FD accuracy of the model. However, the D_k constructed by the multigroup partitioning can contain the samples corresponding to each known fault class, i.e., the distribution of samples will be more uniform, and the number of samples in D_k and D_u are more reasonable. Fifthly, the information entropy loss L_{ie}^k for D_k and the information entropy loss L_{ie}^u for D_u are calculated by

$$L_{ie}^k = \sum_{x_i^k \in D_k} I\left(x_i^k\right) \tag{11}$$

and

$$L_{ie}^u = \sum_{x_i^u \in D_u} I\left(x_i^u\right) \tag{12}$$

respectively, where x_i^k and x_i^u denote the ith sample in D_k and D_u, respectively. Finally, L_{ie}^k and L_{ie}^u are combined to calculate the information entropy loss L_{ie}

$$L_{ie} = L_{ie}^k - L_{ie}^u. \tag{13}$$

By minimizing L_{ie}^k and maximizing L_{ie}^u, the classifier can be allowed to reject the samples of unknown classes in the TD and reduce the distances between the samples of the same class, while the reliability of pseudo-labels can be improved.

D. Design of DCL Strategy

The TD samples can be classified into the known and unknown classes through the MGPPL strategy. However, there may be cases where the information entropy varies too much among the samples of certain fault classes. In one case, the information entropy of most samples is higher but that of individual samples is lower for a certain fault class. In another case, the information entropy of most samples is lower but that of individual samples is higher for a certain fault class. These fault classes that fall into these two cases are called the difficulty classes. It is difficult to reduce the distances between samples only through the MGPPL strategy for the difficulty classes. Therefore, a DCL strategy is proposed, as shown in Fig. 5. The strategy consists of a phase of reducing the intraclass distances for the difficulty classes and a phase of calculating the node classification loss. Algorithm 2 describes the execution of the DCL strategy in one training epoch. Then, the node classification loss is calculated using the different weights for each fault class in D_t, thus allowing the FD model to achieve better diagnosis results.

Algorithm 2 Proposed DCL Strategy

Input: The TD data D_t, the target model M_t, and the threshold ϱ.
Output: The KL divergence loss L_{kl} and the node classification loss L_{node}.

```
for i = 1 to C + 1 do
    Calculate Ī_i by Eq. (14);
    for j = 1 to n_i do
        Calculate φ_j^i by Eq. (15);
    end for
    Calculate φ̄_i by Eq. (16);
    sum = 0;
    for j = 1 to n_i do
        if |φ_j^i − φ̄_i| ≥ ϱ then
            sum + = 1;
        end if
    end for
    if sum ≥ n_i × 10% then
        DC_i = 1;
    else
        DC_i = 0;
    end if
    Calculate W_i by Eq. (19);
end for
Calculate L_kl by Eq. (18);
Calculate L_node by Eq. (20);
```

In the phase of reducing the intraclass distances of difficulty classes, first, the average value of the information entropy $\bar{I}_i$ of all the samples of the ith fault class is calculated by

$$\bar{I}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} I\left(x_j^i\right) \tag{14}$$

where $I(x_j^i)$ denotes the information entropy of the jth sample of the ith class x_j^i in D_t, n_i is the number of samples of the ith fault class, and $1 \le j \le C+1$. Second, the standard deviation (i.e., the entropy difference) of the information entropy φ_j^i of x_j^i is defined as

$$\varphi_j^i = \left| I\left(x_j^i\right) - \bar{I}_i \right|. \tag{15}$$

The entropy difference can reflect the possibility that the class to which the sample belongs is a difficulty class, and the higher entropy difference indicates a higher possibility that the class to which the sample belongs is a difficulty class, and vice versa. Third, the average value of the entropy difference $\bar{\varphi}_i$ for the ith fault class is calculated by

$$\bar{\varphi}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} \varphi_j^i. \tag{16}$$

Fourth, if the absolute differences between the entropy difference of more than 10% of the samples in the ith fault class and $\bar{\varphi}_i$ are greater than or equal to the threshold ϱ, the class is considered to as a difficulty class. If the ith fault class is a difficulty class, then $DC_i = 1$; otherwise, $DC_i = 0$. The feature distribution of each fault class follows a Gaussian distribution

centered at the class feature mean. The feature distribution $Q(\psi_i)$ of the samples of the ith fault class is expressed as

$$Q(\psi_i) = \mathcal{N}(\xi_i, \Sigma_i) \tag{17}$$

where ψ_i denotes all the samples of the ith class, ξ_i is the class feature mean of the ith fault class, and Σ_i is the covariance matrix corresponding to ξ_i. Finally, the KL divergence loss L_{kl} of the difficulty class is calculated by

$$L_{\mathrm{kl}} = \sum_{i=1}^{c+1} DC_i \sum_{j=1}^{n_i} \mathrm{KL}\left(Q\left(x_j^i\right) \| Q(\psi_i)\right) \tag{18}$$

where $Q(x_j^i)$ denotes the feature distribution x_j^i. By minimizing L_{kl}, the distances between the samples in difficulty classes can be reduced, thus enhancing the ability of the FD model to discriminate C_{k} and C_{u}.

In the stage of calculating the node classification loss, considering that the information entropy concentration intervals of different fault classes are different, different weights are set for the samples of different fault classes when calculating the node classification loss. First, $\bar{I}_i$ is normalized to calculate the weight W_i corresponding to the ith fault class by

$$W_i = e^{\bar{I}_i} \Big/ \sum_{j=1}^{C+1} e^{\bar{I}_j}. \tag{19}$$

The smaller W_i is, it means that the sample features of the ith fault class are easier to learn, and vice versa. Then, the node classification loss L_{node} is calculated by

$$L_{\mathrm{node}} = \sum_{i=1}^{C+1} W_i \sum_{j=1}^{n_i} \tilde{y}_j^i \log\left(\hat{y}_j^i\right) \tag{20}$$

where $\tilde{y}_j^i$ and $\hat{y}_j^i$ are the pseudo-label and predicted label of the jth sample of the ith class, respectively. The calculation of L_{node} using different weights for the samples of different classes can better facilitate the aggregation of samples of the same class, thus achieving further separation of C_{k} and C_{u}.

E. Parameter Updating

The parameter update consists of updating the parameters of M_{s} and M_{t}, as described in Algorithm 3. The parameters of M_{s} are updated by minimizing the cross-entropy loss L_{c}, which is calculated by

$$L_{\mathrm{c}} = -\frac{1}{n_{\mathrm{s}}} \sum_{i=1}^{n_{\mathrm{s}}} y_i^{\mathrm{s}} \log\left(\hat{y}_i^{\mathrm{s}}\right) \tag{21}$$

where $\hat{y}_i^{\mathrm{s}}$ is the predicted label of the sample x_i^{s}. By minimizing L_{c} to find the optimal parameters $\dot{\theta}_{\mathrm{G^s}}$ of the feature extractor and the optimal parameters $\dot{\theta}_{\mathrm{C^s}}$ of the classifier in M_{s}

$$\left(\dot{\theta}_{\mathrm{G^s}}, \dot{\theta}_{\mathrm{C^s}}\right) = \arg\min_{\theta_{\mathrm{G^s}}, \theta_{\mathrm{C^s}}} L_{\mathrm{c}} \tag{22}$$

where $\theta_{\mathrm{G^s}}$ and $\theta_{\mathrm{C^s}}$ represent the parameters of the feature extractor and the classifier in M_{s}, respectively.

Algorithm 3 Parameter Updating Procedure of the Proposed IESOFD

Input: The SD data D_{s}, the TD data D_{t}, the number of training epochs for the source model e_{s}, the number of training epochs for the target model e_{t}, and the weight parameters α, β, and γ.

Output: The optimal parameters $\dot{\theta}_{\mathrm{G^s}}$, $\dot{\theta}_{\mathrm{C^s}}$, $\dot{\theta}_{\mathrm{G^t}}$, and $\dot{\theta}_{\mathrm{C^t}}$.

▷ Training of M_{s}
for $i = 1$ **to** e_{s} **do**
  Calculate L_{c} by Eq. (20);
  Update $\dot{\theta}_{\mathrm{G^s}}$ and $\dot{\theta}_{\mathrm{C^s}}$ by Eq. (21);
end for
▷ Training of M_{t}
for $i = 1$ **to** e_{t} **do**
  Calculate L_{ie} by Eq. (13);
  Calculate L_{kl} by Eq. (18);
  Calculate L_{node} by Eq. (20);
  Calculate L_{sc} by Eq. (23);
  Calculate L_{total} by Eq. (27);
  Update $\dot{\theta}_{\mathrm{G^t}}$ and $\dot{\theta}_{\mathrm{C^t}}$ by Eq. (28);
end for

During the training of M_{t}, the more structural information are obtained by calculating the structural consistency loss L_{sc} of the kNN graph of the TD samples. L_{sc} is calculated by

$$L_{\mathrm{sc}} = \sum_{v_i \in V} \sum_{v_j \in V} \left(\mathcal{S}_{\mathrm{node}}^{ij} + \mathcal{S}_{\mathrm{sc}}^{ij}\right). \tag{23}$$

In (23), $\mathcal{S}_{\mathrm{node}}^{ij}$ denotes the node similarity between v_i and v_j and is calculated by

$$\mathcal{S}_{\mathrm{node}}^{ij} = A_{ij} \log \tau\left(\langle \hat{y}_i^{\mathrm{t}}, \hat{y}_j^{\mathrm{t}} \rangle\right) + \left(1 - A_{ij}\right) \log \tau\left(-\langle \hat{y}_i^{\mathrm{t}}, \hat{y}_j^{\mathrm{t}} \rangle\right) \tag{24}$$

where $\hat{y}_i^{\mathrm{t}}$ indicates the predicted label of the ith sample from D_{t}, $\tau(x) = 1/(1 + e^{-x})$ is the sigmoid function, and $\langle \cdot, \cdot \rangle$ represents the inner product. $\mathcal{S}_{\mathrm{sc}}^{ij}$ indicates the structural similarity between v_i and v_j and is calculated by

$$\mathcal{S}_{\mathrm{sc}}^{ij} = M_{ij} \log \tau\left(\langle \hat{y}_i^{\mathrm{t}}, \hat{y}_j^{\mathrm{t}} \rangle\right) + \left(1 - M_{ij}\right) \log \tau\left(-\langle \hat{y}_i^{\mathrm{t}}, \hat{y}_j^{\mathrm{t}} \rangle\right). \tag{25}$$

If the node pair v_i and v_j is one of the top $n \times K$ node pairs with the highest degree sequence similarities in the kNN graph containing n nodes, $M_{ij} = 1$; otherwise, $M_{ij} = 0$, where K is the number of nearest neighbors in the kNN graph. The degree sequence similarity $\mathcal{T}_{ij}$ between v_i and v_j is calculated by

$$\mathcal{T}_{ij} = |\mathcal{D}_i \cap \mathcal{D}_j| \Big/ \sqrt{|\mathcal{D}_i| \times |\mathcal{D}_j|} \tag{26}$$

where $\mathcal{D}_i$ and $\mathcal{D}_j$ denote the neighbor sets of the nodes v_i and v_j, respectively, $|\mathcal{D}_i \cap \mathcal{D}_j|$ indicates the number of common neighbors of the nodes v_i and v_j, $|\mathcal{D}_i|$ and $|\mathcal{D}_j|$ represent the degrees of nodes v_i and v_j, respectively. The predicted similarity between connected nodes and that between structurally similar nodes can be improved by minimizing L_{sc}.

The parameters of M_{t} are updated by minimizing the total loss L_{total}, which is calculated by

$$L_{\mathrm{total}} = \alpha L_{\mathrm{ie}} + \beta L_{\mathrm{sc}} + \gamma L_{\mathrm{kl}} + L_{\mathrm{node}} \tag{27}$$

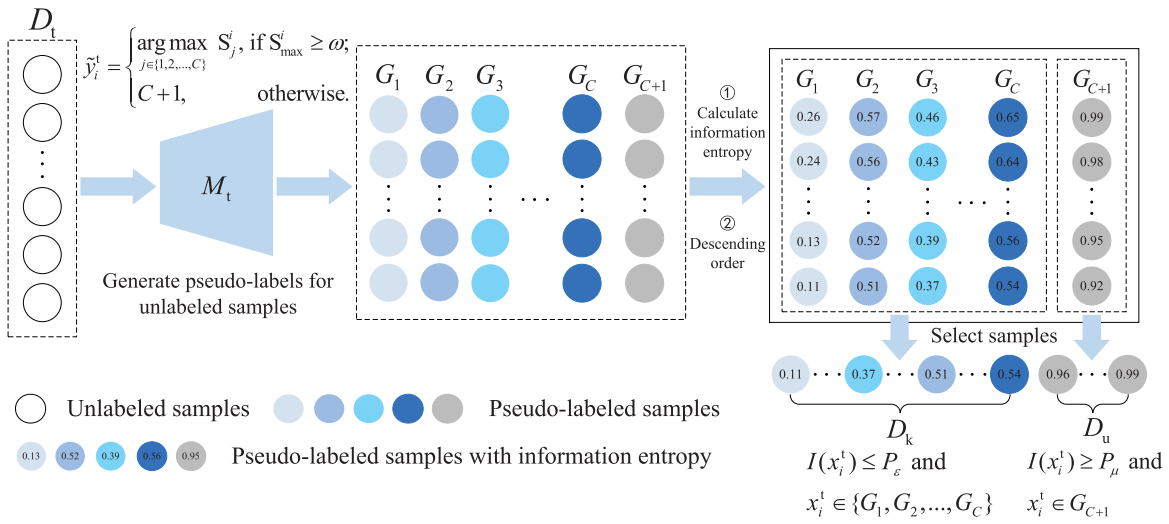


Fig. 4. Proposed MGPPL strategy.

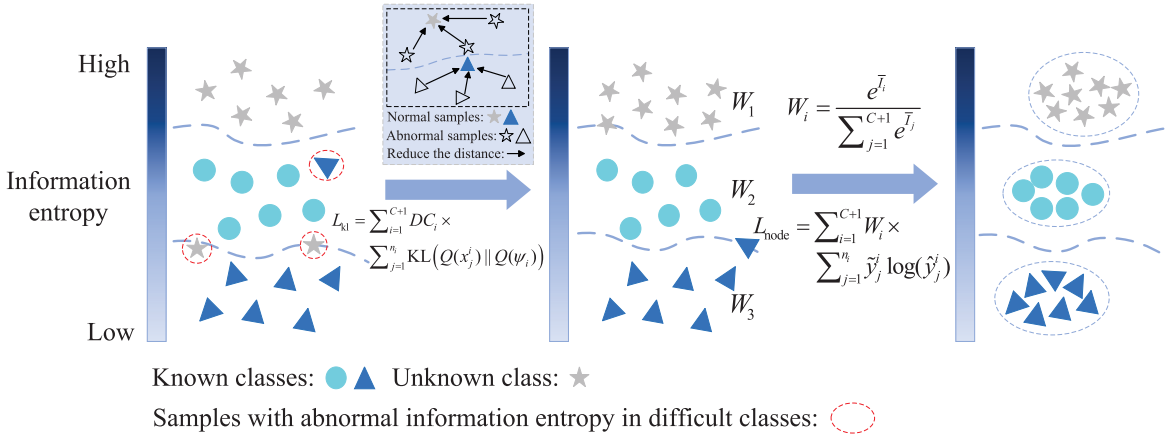


Fig. 5. Proposed DCL strategy.

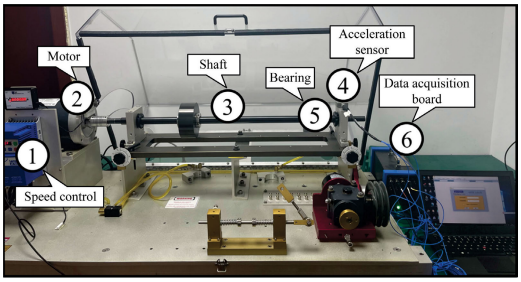


Fig. 6. HUST bearing test rig [39].

TABLE I
HEALTH STATES OF HUST BEARING

Health state	Damage location	Class label
Normal	-	1
Moderate fault	Rolling element	2
	Inner race	3
	Outer race	4
Severe fault	Rolling element	5
	Inner race	6
	Outer race	7

where α, β, and γ are the different weighting parameters, all of which are valued between 0 and 1. During the iterative updating of M_t, the process of searching the optimal parameters $\dot{\theta}_{G^t}$ of the feature extractor and the optimal parameters $\dot{\theta}_{C^t}$ of the classifier in M_t can be expressed as

$$\left(\dot{\theta}_{G^t}, \dot{\theta}_{C^t}\right) = \arg\min_{\theta_{G^t}\theta_{C^t}} L_{\text{total}} \tag{28}$$

where θ_{G^t} and θ_{C^t} are the parameters of the feature extractor and the classifier in M_t, respectively. The source-free OSFD can be effectively realized through the final trained M_t. When the training of M_s and M_t have been completed, the test sample x_i^t from TD is input into M_t. After completing the forward propagation of M_t, the class prediction probability vector $M_t(x_i^t)$ of x_i^t is obtained. The class of x_i^t is determined according to the maximum class prediction probability in $M_t(x_i^t)$, thereby the FD result of x_i^t is obtained.

TABLE II
FD TASKS ON HUST DATASET

Cross-condition	Task	Source classes	Target classes	Openness
65 Hz → 70 Hz	H1	1, 2, 3, 4, 5, 6, 7	1, 2, 3, 4, 5, 6, 7	0
	H2	1, 2, 3, 4, 5, 6		0.14
	H3	1, 2, 3, 4, 5		0.29
	H4	1, 2, 3, 4		0.43
	H5	1, 2, 3		0.57
65 Hz → 75 Hz	H6	1, 2, 3, 4, 5, 6, 7	1, 2, 3, 4, 5, 6, 7	0
	H7	1, 2, 3, 4, 5, 6		0.14
	H8	1, 2, 3, 4, 5		0.29
	H9	1, 2, 3, 4		0.43
	H10	1, 2, 3		0.57

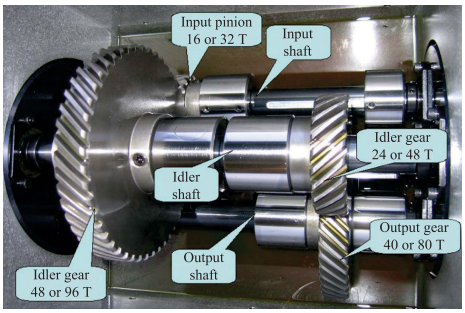


Fig. 7. PHM gearbox test rig [40].

TABLE III
DESCRIPTION OF PHM DATASET

Class	Gear		Bearing			Shaft	
label	24 T	Others	IS:OS	ID:OS	Others	Input	Output
1	G	G	G	G	G	G	G
2	CH	G	G	G	G	G	G
3	BR	G	CO	Inner	G	BS	G
4	G	G	CO	Ball	G	IM	G
5	BR	G	G	Inner	G	G	G
6	G	G	G	G	G	BS	G

IS = Input shaft; OS = Output side; ID = Idler shaft; G = Good; CH = Chipped; BR = Broken; CO = Combination; BS = Bent shaft; IM = Imbalance

IV. EXPERIMENTS

A. Experimental Setup

To validate the effectiveness of the proposed IESOFD approach, extensive experiments are performed on the Huazhong University of Science and Technology (HUST) [39] and Prognostics and Health Management (PHM) [40] datasets. The HUST bearing test rig is depicted in Fig. 6. In this experiment, the bearing vibration data collected at a sampling frequency of 25.6 kHz under the three different operating conditions of 65, 70, and 75 Hz are selected. Table I lists the seven different health states contained in each operating condition, where each health state contains 600 samples. The training and testing sets are split by 8:2. For the HUST dataset, the ten different FD tasks are designed according to different openness, as depicted in Table II.

Fig. 7 illustrates the PHM gearbox test rig. In this experiment, the vibration signals collected at a sampling frequency of 66.67 kHz under the three different operating conditions of 30, 35, and 40 Hz are chosen. Table III lists the six different health states contained in each operating condition. Each health state includes 800 samples. The training and testing sets are split by 8:2. The eight different FD tasks are designed for the PHM dataset according to different openness, as seen in Table IV.

TABLE IV
FD TASKS ON PHM DATASET

Cross-condition	Task	Source classes	Target classes	Openness
30 Hz → 35 Hz	P1	1, 2, 3, 4, 5, 6	1, 2, 3, 4, 5, 6	0
	P2	1, 2, 3, 4, 5		0.17
	P3	1, 2, 3, 4,		0.33
	P4	1, 2, 3		0.50
35 Hz → 40 Hz	P5	1, 2, 3, 4, 5, 6	1, 2, 3, 4, 5, 6	0
	P6	1, 2, 3, 4, 5		0.17
	P7	1, 2, 3, 4		0.33
	P8	1, 2, 3		0.50

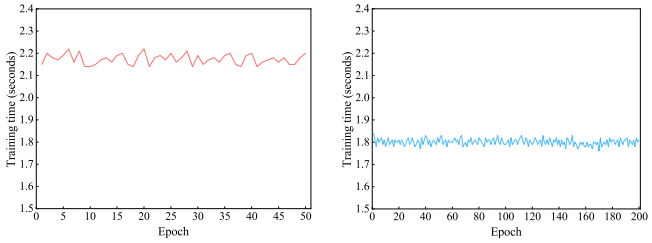
TABLE V
NETWORK STRUCTURE OF THE MODEL

Module	Layer type	Kernel / Channels / Stride	Output
Feature extractor	Conv	7×1 / 64 / 2	64, 512
	BN & ReLU	–	64, 512
	Max-pooling	3×1 / 64 / 2	64, 256
	Res block 1	$\begin{bmatrix} 3 \times 1 \text{ / } 64 \text{ / } 2 \\ 3 \times 1 \text{ / } 64 \text{ / } 1 \end{bmatrix} \times 2$	64, 256
	Res block 2	$\begin{bmatrix} 3 \times 1 \text{ / } 128 \text{ / } 2 \\ 3 \times 1 \text{ / } 128 \text{ / } 1 \end{bmatrix} \times 2$	128, 128
	Res block 3	$\begin{bmatrix} 3 \times 1 \text{ / } 256 \text{ / } 2 \\ 3 \times 1 \text{ / } 256 \text{ / } 1 \end{bmatrix} \times 2$	256, 64
	Res block 4	$\begin{bmatrix} 3 \times 1 \text{ / } 512 \text{ / } 2 \\ 3 \times 1 \text{ / } 512 \text{ / } 1 \end{bmatrix} \times 2$	512, 32
	Adaptive-avg-pooling	32×1 / 512 / 32	512
GCN module	GConv 1 & ReLU	–	512
	GConv 2 & ReLU	–	512
Classifier	Flatten	–	512
	FC	–	C (for M_s) $C+1$ (for M_t)

TABLE VI
SETTING OF HYPERPARAMETERS ADOPTED IN MODEL TRAINING

Parameter name	Parameter description	Parameter value
K	Number of nearest neighbors in a kNN graph	3
ω	Threshold used in MGPPL strategy	0.85
ε	Threshold used to select samples to form D_k	90
μ	Threshold used to select samples to form D_u	15
ϱ	Threshold used to select the difficulty classes	0.1
α	Weighting of L_{ie}	0.5
β	Weighting of L_{sc}	0.3
γ	Weighting of L_{kl}	0.6

In the proposed IESOFD approach, the network structure of the model is shown in Table V. Both the source model M_s and the target model M_t consist of a feature extractor, a GCN module, and a classifier. The M_s and M_t share the same network structure, except for the difference in the classifier structure. The fault classifiers used in the M_s and M_t consist of a flatten layer and an FC layer with the output dimensions



(a) (b)

Fig. 8. Training time of each epoch during different phases. (a) Source model pretraining. (b) Target model fine-tuning.

of C and $C+1$, respectively. The training epochs for M_{s} and M_{t} are set to 50 and 200, respectively. The batch size, learning rate, and optimizer employed in the training of M_{s} and M_{t} are all set to 64, 0.01, and stochastic gradient descent, respectively. The other hyperparameter values are determined through the grid search method, as listed in Table VI. Note that the bottom $(100-\varepsilon)\%$ of samples in each group of $\{G_1, G_2, \ldots, G_C\}$ and the top $\mu\%$ of samples in G_{C+1} are selected after being sorted in descending order of information entropy of each group, respectively.

The hardware environment mainly includes a 10-core Intel i5-13400F CPU, an RTX 4060 Ti GPU with 4352 CUDA cores, 32 GB of RAM, and 8 GB of GPU memory. The software environment mainly includes PyTorch 1.8 and Python 3.9.

In the proposed IESOFD approach, the time complexity of the source and target models is $O(\mathcal{L}\times\mathcal{C}^2+\mathcal{G}\times\mathcal{D}^2)$, where $\mathcal{L}$ indicates the input sequence length, $\mathcal{C}$ denotes the number of channels, $\mathcal{G}$ is the number of graph nodes, and $\mathcal{D}$ represents the feature dimension. The space complexity of the source and target models is $O(\mathcal{P}+\mathcal{L}\times\mathcal{C}+\mathcal{G}\times\mathcal{D}+\mathcal{G}^2)$, where $\mathcal{P}$ is the total number of parameters in the model and it is approximately 3.85×10^6. To better explore the computational complexity of the model, an analysis is conducted on the training time, reasoning time, and hardware resource consumption during the source model pretraining and the target model fine-tuning. Taking the task H2 of the HUST dataset as an example, where six known classes are participated in the source model pretraining, while six known classes and one unknown class are participated in the target model fine-tuning, and each class contains 480 training samples and 120 testing samples. Fig. 8(a) and (b) presents the training time of each epoch during the source model pretraining and target model fine-tuning, respectively. During the source model pretraining, the average training time of each epoch and the total training time are approximately 2.17 and 109 s, respectively. During the target model fine-tuning, the average training time of each epoch and the total training time are approximately 1.80 and 360 s, respectively. The reasoning time of the target model for each sample is approximately 0.23 ms, which allows the target model to meet the real-time diagnosis requirement in the real industry and fit into the lightweight edge devices. The system main memory and GPU memory required for the source model pretraining and the target model fine-tuning are approximately 6.44 and 4.25 GB, respectively. The target model size is approximately 15 MB. The low memory overhead makes the target model suitable for lightweight edge-end deployments. It can be found from the above analysis that the proposed IESOFD has fast training and reasoning speeds, moderate system main memory and GPU memory required for model training, and a smaller mode size, indicating that it has a good prospect in practical industrial deployment.

TABLE VII
EVALUATION METRICS

Metric	Description	Definition
OS	Identification accuracy for all the classes	$\frac{F_{\mathrm{k}}+F_{\mathrm{u}}}{P_{\mathrm{k}}+P_{\mathrm{u}}}$
OS^*	Identification accuracy for the known classes	$F_{\mathrm{k}}/P_{\mathrm{k}}$
UNK	Identification accuracy for the unknown classes	$F_{\mathrm{u}}/P_{\mathrm{u}}$
H	Harmonic mean of OS^* and UNK	$\frac{2\times OS^*\times UNK}{OS^*+UNK}$

B. Evaluation Metrics

In this experiment, the four different evaluation metrics listed in Table VII are adopted for measuring the FD performance of the proposed IESOFD and the comparison approaches. In Table VII, F_{k} and F_{u} represent the number of testing samples of the known and unknown classes correctly identified, respectively. P_{k} and P_{u} denote the total number of testing samples of the known and unknown classes, respectively. H indicates the harmonic mean of OS^* and UNK. The higher the value of H, the better the separation effect for the unknown classes and the identification effect for the known fault classes. To ensure the reliability of the evaluation results, when different RMFD approaches are adopted for conducting experiments on different FD tasks of HUST and PHM datasets, each experiment is repeated 30 times, and the average OS, OS^*, UNK, and H are calculated.

C. Validation of the Effectiveness of GCN Module

To validate the effectiveness of the GCN module, the proposed IESOFD approach adopting the GCN module is compared with the IESOFD-CNN and IESOFD-Res on the FD tasks of HUST and PHM datasets, where IESOFD-CNN and IESOFD-Res are the variants of IESOFD. In IESOFD-CNN, the GConv layers are replaced with the Conv layers in the original GCN module. In IESOFD-Res, the GCN module is eliminated, and only ResNet-18 is adopted as the feature extractor. Fig. 9(a) and (b) shows the OS of the three approaches on the FD tasks of HUST and PHM datasets, respectively. On the ten different FD tasks of the HUST dataset, the average OS of IESOFD, IESOFD-CNN, and IESOFD-Res are 95.42%, 92.16%, and 89.70%, respectively. On the eight different FD tasks of the PHM dataset, the average OS of IESOFD, IESOFD-CNN, and IESOFD-Res are 94.80%, 90.74%, and 88.29%, respectively. The performance improvement of IESOFD compared to IESOFD-CNN and IESOFD-Res is due to its ability to fully utilize structural relationships between samples through GCN to extract more robust features. In IESOFD-CNN and IESOFD-Res, the features are extracted through standard convolutional operations,

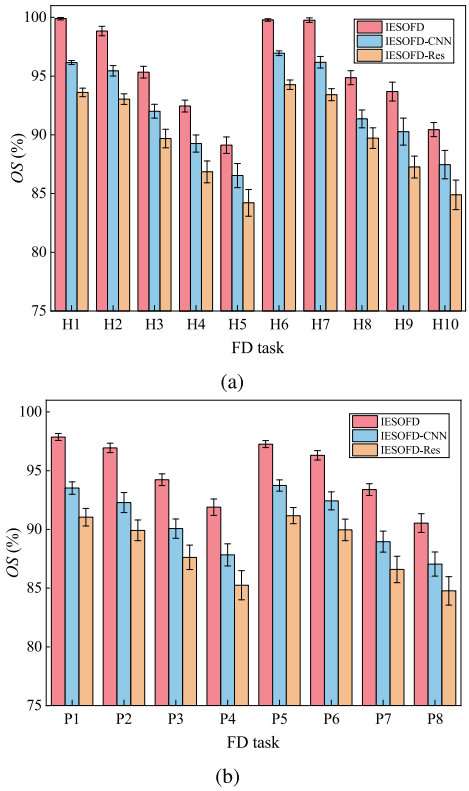


Fig. 9. *OS* of different approaches on the FD tasks of different datasets. (a) *OS* of the HUST dataset. (b) *OS* of PHM dataset.

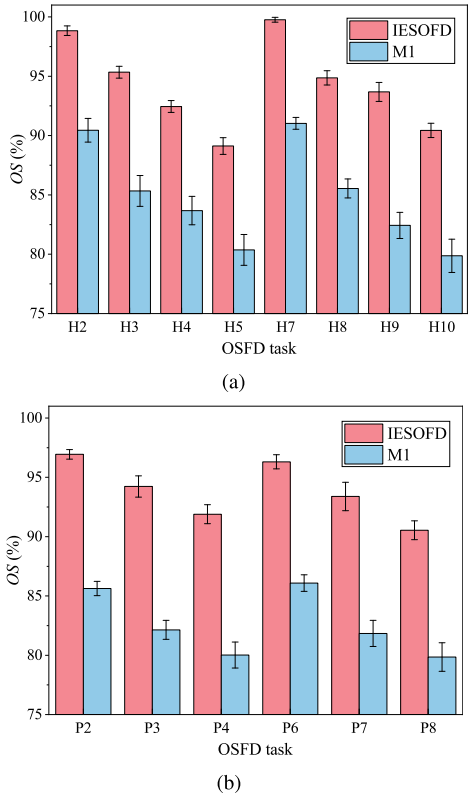


Fig. 10. *OS* of IESOFD and M1 on the OSFD tasks of different datasets. (a) *OS* of the HUST dataset. (b) *OS* of PHM dataset.

leading to the potential correlation between samples of similar fault classes may be ignored. In contrast, the GCN module used in IESOFD fully considers the structural correlation between samples, which can extract more robust and generalized features. In IESOFD, the potential structural relationships between sample features can be captured via GCN, and the more structural information can be obtained via the node similarity and structure similarity in the kNN graph of samples, thereby achieving more accurate identification of known and unknown fault classes.

D. Validation of the Effectiveness of the Proposed MGPPL Strategy

To validate the effectiveness of the MGPPL strategy, the proposed IESOFD approach adopting the MGPPL strategy and the M1 approach are compared on the 14 OSFD tasks of the HUST and PHM datasets. M1 is a variant of IESOFD. The difference is that when D_{k} and D_{u} are formed, M1 does not adopt the MGPPL strategy, but adopts the way of global partitioning. Specifically, M1 ranks the information entropy of all the samples from high to low, and selects the samples whose information entropy is less than or equal to P_{ε} as D_{k} and the samples whose information entropy is greater than or equal to P_{μ} as D_{u}. Fig. 10(a) and (b) shows the *OS* of IESOFD and M1 on the OSFD tasks of HUST and PHM datasets, respectively. On the eight OSFD tasks of the HUST dataset, the average *OS* of IESOFD and M1 are 94.31% and 84.71%, respectively. On the six OSFD tasks of the PHM dataset, the average *OS* of IESOFD and M1 are 93.88% and 82.06%, respectively. On the OSFD tasks of HUST and PHM datasets, the average *OS* of IESOFD is increased by 9.60% and 11.82% compared with M1, respectively. This is mainly because the D_{k} formed by M1 adopting the global partitioning only contains the samples of some known fault classes, making M1 overfit for a small number of fault classes and underfit for most fault classes. Fig. 11(a) and (b) depicts the D_{k} and D_{u} formed with IESOFD and M1 on the task H2 of HUST dataset, respectively. In Fig. 11(a) and (b), these samples are selected using IESOFD adopting MGPPL strategy and M1, respectively. In order to more intuitively present the samples belong to which fault class, their real class labels are shown in Fig. 11. As shown in Fig. 11(a), the D_{k} formed by IESOFD adopting the MGPPL strategy contains samples of all known fault classes. As depicted in Fig. 11(b), the D_{k} formed by M1 using the global partitioning only includes samples of three known fault classes, where most of them belong to class 3, a few of them belong to class 2, and an extremely few of them belong to class 6, without including samples of classes 1, 4, and 5. This is mainly because the samples in D_{k} formed by M1 using the global partitioning are directly selected after sorting the information entropy of all samples in descending order. Therefore, it is impossible to select samples of each known fault class. In total, the proposed IESOFD adopting the MGPPL strategy can form a more reasonable D_{k} and D_{u}, thereby promoting the separation of the known and unknown classes.

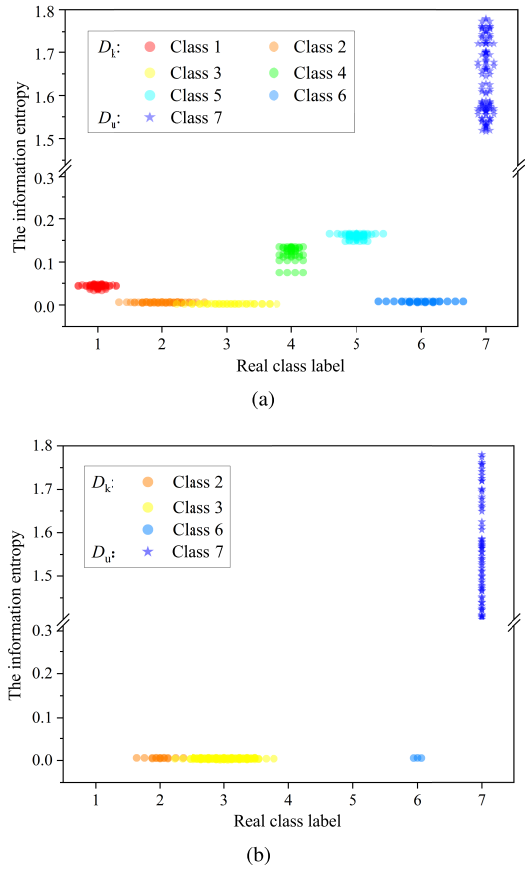


Fig. 11. D_k and D_u formed by different approaches on the task H2 of HUST dataset. (a) IESOFD. (b) M1.

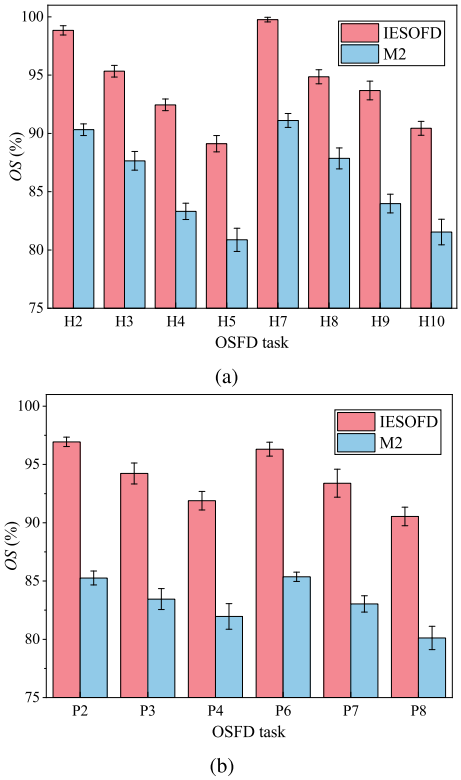


Fig. 12. *OS* of IESOFD and M2 on the OSFD tasks of different datasets. (a) *OS* of the HUST dataset. (b) *OS* of PHM dataset.

E. Validation of the Effectiveness of the Proposed DCL Strategy

To validate the effectiveness of the DCL strategy, the proposed IESOFD approach adopting the DCL strategy and the M2 approach are compared on the 14 OSFD tasks of HUST and PHM datasets. M2 is a variant of IESOFD. The difference is that after M2 adopts the MGPPL strategy to form D_k and D_u, it does not deal with the difficulty classes but instead of calculating directly L_{node} and taking the same weighting for all the fault classes. Fig. 12(a) and (b) reveals the *OS* of IESOFD and M2 on the OSFD tasks of HUST and PHM datasets, respectively. On the eight OSFD tasks of the HUST dataset, the average *OS* of IESOFD and M2 are 94.31% and 85.83%, respectively. On the six OSFD tasks of the PHM dataset, the average *OS* of IESOFD and M2 are 93.88% and 83.20%, respectively. On the OSFD tasks of HUST and PHM datasets, the average *OS* of IESOFD is increased by 8.48% and 10.68% compared with M2, respectively. This is mainly because M2 takes the same weighting for each fault class to calculate L_{node}, leading to the poor effect of intraclass aggregation and interclass separation. However, the DCL strategy adopted in IESOFD fully considers the difference of samples and adopts different weights for different fault classes to calculate L_{node}. This not only shortens the intraclass distances, but also lengthens the interclass distances, thereby better separating different fault classes.

F. Comparison With Other RMFD Approaches

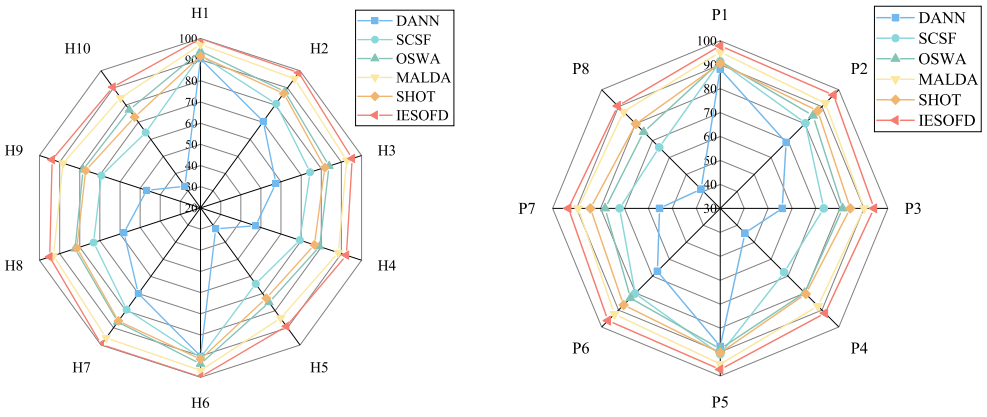
To further validate the effectiveness of the proposed IESOFD approach, IESOFD and the following five advanced RMFD approaches are compared on HUST and PHM datasets: DANN [41], SCSF [22], OSWA [42], MALDA [31], and SHOT [43]. DANN is a CSDA approach that achieves domain alignment through adversarial learning. SCSF is a CSDA approach that achieves domain alignment by generating pseudo-labels through prototypical clustering. OSWA is an OSDA approach that detects the similarity of instances to known classes using an instance-level weight strategy. MALDA is an OSDA approach that recognizes both known and unknown classes through domain-adversarial learning. SHOT is an OSDA approach that achieves domain-invariant feature learning by introducing information maximization and self-supervised learning. In these comparison approaches, DANN, OSWA, and MALDA require access to the SD data, whereas SCSF, SHOT, and IESOFD require no direct access to the SD data.

Table VIII lists the OS^*, *UNK*, and *H* obtained by different approaches on different FD tasks of HUST and PHM datasets. As shown in Table VIII, the average OS^*, *UNK*, and *H* obtained by IESOFD on HUST dataset are 97.72%, 94.51%, and 95.82%, respectively, and the average OS^*, *UNK*, and *H* obtained by IESOFD on PHM dataset are 94.28%, 93.42%, and 93.30%, respectively. The OS^*, *UNK*, and *H* obtained by IESOFD on HUST and PHM datasets are superior to the other comparison approaches. For example, on the HUST dataset, the average OS^* obtained by IESOFD is 15.45%,

TABLE VIII

OS^* (%), UNK (%), AND H (%) OBTAINED BY DIFFERENT APPROACHES ON DIFFERENT FD TASKS OF HUST AND PHM DATASETS

Task	DANN			SCSF			OSWA			MALDA			SHOT			IESOFD		
	OS^*	UNK	H	OS^*	UNK	H	OS^*	UNK	H	OS^*	UNK	H	OS^*	UNK	H	OS^*	UNK	H
H1	90.33±0.89	–	–	92.12±0.77	–	–	93.96±0.81	–	–	97.13±0.43	–	–	91.73±0.57	–	–	99.89±0.13	–	–
H2	88.94±1.02	–	–	90.21±0.97	–	–	92.52±1.01	82.41±1.54	87.17	96.72±0.51	92.43±0.64	94.53	92.16±0.72	84.36±1.02	88.09	98.43±0.25	97.75±0.53	98.09
H3	81.64±1.49	–	–	87.49±1.37	–	–	90.67±1.23	78.67±1.84	84.24	94.84±0.57	90.27±0.77	92.50	90.52±0.78	82.37±0.97	86.25	99.65±0.41	95.77±0.67	97.67
H4	78.39±2.01	–	–	86.05±1.56	–	–	87.96±1.47	76.02±2.01	81.56	93.37±0.85	87.45±1.03	90.31	87.94±1.12	78.39±1.42	83.44	97.34±0.53	92.73±0.88	94.98
H5	74.65±2.42	–	–	83.51±1.79	–	–	85.67±1.69	71.67±2.53	78.05	92.43±0.86	84.82±1.17	88.46	86.58±1.31	75.38±1.64	80.59	95.14±0.86	90.33±1.31	92.67
H6	89.97±1.02	–	–	90.71±1.11	–	–	93.65±0.89	–	–	96.74±0.52	–	–	91.47±0.89	–	–	99.78±0.11	–	–
H7	87.45±1.43	–	–	90.04±1.34	–	–	92.31±1.02	83.06±1.37	87.44	96.21±0.53	93.06±0.88	94.61	90.14±0.94	83.97±1.58	86.95	99.83±0.18	97.50±0.51	98.65
H8	82.51±1.78	–	–	87.54±1.69	–	–	90.34±1.43	80.57±1.74	85.17	95.43±0.67	91.69±0.85	93.52	88.92±0.99	80.24±1.87	84.26	97.53±0.64	96.56±0.79	97.04
H9	76.47±2.15	–	–	84.33±1.89	–	–	88.17±1.66	77.93±2.17	82.73	93.26±0.75	88.51±1.94	90.82	85.83±1.82	77.63±2.03	81.52	94.40±0.89	92.56±0.92	93.47
H10	72.31±2.19	–	–	82.17±2.27	–	–	87.78±1.83	75.04±2.14	80.91	90.14±1.72	85.31±2.13	87.65	83.02±2.32	74.92±2.57	78.76	95.19±1.12	92.88±1.03	94.02
Avg.	82.27	–	–	87.42	–	–	90.30	78.17	83.41	94.62	89.19	91.55	88.83	79.78	83.75	**97.72**	**94.51**	**95.82**
P1	88.17±1.01	–	–	91.32±0.68	–	–	91.37±1.10	–	–	95.27±0.57	–	–	90.52±0.84	–	–	97.87±0.26	–	–
P2	86.57±1.32	–	–	88.32±1.25	–	–	89.73±1.49	78.74±1.02	83.87	94.52±0.64	90.62±0.87	92.53	89.27±0.85	81.53±1.52	85.22	95.97±0.39	96.98±0.48	96.47
P3	80.34±1.68	–	–	86.95±1.51	–	–	86.05±1.85	77.61±1.31	81.61	92.67±0.75	87.35±1.32	89.93	86.94±1.53	79.93±1.42	82.74	93.75±0.87	93.86±0.64	93.80
P4	75.42±2.02	–	–	82.84±1.82	–	–	83.28±2.14	74.52±1.64	78.66	90.19±1.02	84.32±1.68	87.16	83.29±1.57	74.37±1.86	78.68	90.86±0.81	90.25±0.99	90.55
P5	87.63±1.08	–	–	90.88±0.97	–	–	88.69±1.26	–	–	95.03±0.56	–	–	90.28±0.71	–	–	97.26±0.30	–	–
P6	86.37±1.41	–	–	87.61±1.32	–	–	87.58±1.64	76.47±1.37	81.65	94.17±0.63	91.02±0.96	92.57	88.93±1.36	82.14±1.18	85.40	95.32±0.57	96.45±0.57	95.88
P7	79.85±1.77	–	–	85.41±1.67	–	–	85.35±1.96	75.06±1.57	79.87	92.48±0.63	87.66±1.35	90.01	86.53±1.72	78.44±2.38	83.24	93.09±1.16	93.17±0.75	93.13
P8	72.86±2.16	–	–	81.67±1.99	–	–	83.29±2.05	73.19±1.96	77.91	89.17±1.46	83.93±2.12	84.51	83.26±1.64	74.69±2.14	78.74	90.13±0.83	89.83±1.26	89.98
Avg.	82.15	–	–	86.88	–	–	86.92	75.93	80.60	93.19	87.48	89.45	87.38	78.35	82.34	**94.28**	**93.42**	**93.30**



(a) (b)

Fig. 13. OS (%) achieved by different approaches on different datasets. (a) OS achieved on the HUST dataset. (b) OS achieved on the PHM dataset.

10.30%, 7.42%, 3.10%, and 8.89% higher than DANN, SCSF, OSWA, MALDA, and SHOT, respectively. This is mainly due to the fact that both DANN and SCSF belong to CSDA approaches, which are unable to recognize the unknown fault classes, thus incorrectly predicting the unknown fault classes as the known fault classes. Although OSWA, MALDA, and SHOT belong to OSDA approaches, all of them fail to completely separate the unknown fault classes, resulting in a small number of unknown fault classes being recognized as the known fault classes. On the PHM dataset, the average UNK achieved by IESOFD is 93.42%, 93.42%, 17.49%, 3.85%, and 10.96% higher than DANN, SCSF, OSWA, MALDA, and SHOT, respectively. This indicates that the proposed IESOFD approach can accurately identify the known and unknown classes. DANN and SCSF belong to CSDA approaches, which cannot recognize the unknown classes. OSWA, MALDA, and SHOT belong to the OSDA approaches, which have the ability to recognize the unknown classes. The comparison with SHOT is worth further analysis. SHOT regards open-set recognition as a problem of clustering the known classes and rejecting the unknown classes. SHOT filters out the unknown class samples based on the k-means clustering and the information maximization loss. The unknown class samples do not participate in the optimization of the objective function during training. This design is effective when there is a significant discrepancy in feature distributions between known and unknown class samples. However, it is difficult to separate the unknown classes when the unknown class samples are scattered near the known class samples. In contrast, IESOFD can fully learn the features of known and unknown class samples by maximizing the information entropy of unknown classes and minimizing the information entropy of known classes, thereby achieving more accurate recognition of known and unknown classes. The reasons why IESOFD performs better than all the other comparison methods are as follows. First, IESOFD better separates the unknown fault classes mixed in the known fault classes by adopting the MGPPL strategy. Second, IESOFD effectively reduces the intraclass distances and constructs better decision

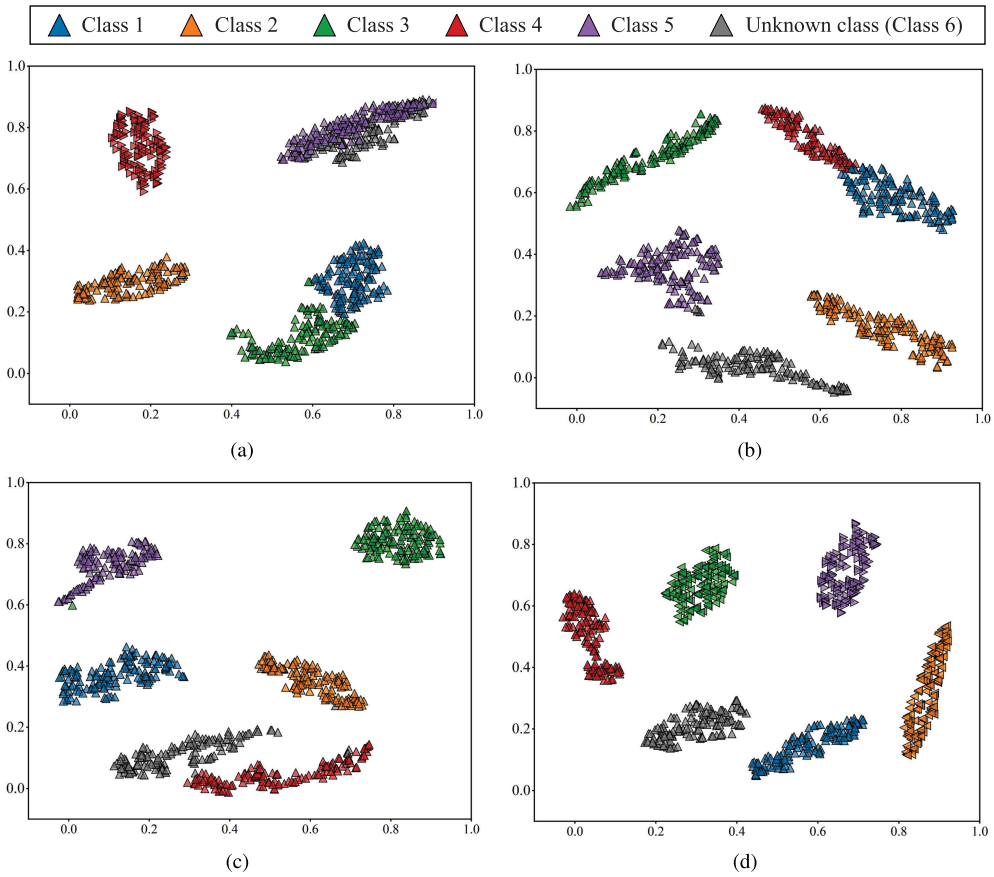


Fig. 14. Feature visualizations of FD results of different approaches on the task P2. (a) DANN. (b) OSWA. (c) SHOT. (d) IESOFD.

boundaries between different fault classes by adopting the DCL strategy. Third, IESOFD better extracts the fault features by the GCN.

Fig. 13(a) and (b) displays the *OS* achieved by different approaches on HUST and PHM datasets, respectively. On the HUST dataset, the average *OS* achieved by DANN, SCSF, OSWA, MALDA, SHOT, and IESOFD are 59.51%, 75.73%, 83.80%, 91.59%, 81.83%, and 95.42%, respectively. On the PHM dataset, the average *OS* obtained by DANN, SCSF, OSWA, MALDA, SHOT, and IESOFD are 63.62%, 77.73%, 82.75%, 91.14%, 85.56%, and 94.80%, respectively. This further indicates that the proposed IESOFD approach better identifies the known and unknown classes.

Fig. 14(a)–(d) illustrates the feature visualizations of FD results of DANN, OSWA, SHOT, and IESOFD on the task P2, respectively, where the t-SNE technique is used to visualize the features. As seen from Fig. 14(a), the unknown fault class is mixed with the known fault classes in the FD results of DANN, indicating DANN cannot separate the unknown fault class. As shown in Fig. 14(b) and (c), OSWA and SHOT can accurately identify most fault classes, but they have not established good decision boundaries, resulting in the mixing of samples from a few fault classes. As depicted in Fig. 14(d), the proposed IESOFD approach not only establishes clear decision boundaries between different fault classes, but also has good clustering within the same fault class. This indicates that IESOFD achieves better OSFD performance.

Fig. 15(a)–(d) shows the confusion matrices of DANN, OSWA, SHOT, and IESOFD on the task H2, respectively. As observed from Fig. 15(a), DANN incorrectly predicts 20%, 5%, and 75% of the samples of class 7 (i.e., the unknown fault class) as class 1, class 3, and class 4, respectively, which is because DANN has no capability to distinguish the unknown fault class. As shown in Fig. 15(b), OSWA incorrectly predicts 5%, 10%, and 3% of the samples of class 7 as class 1, class 3, and class 5, respectively. As depicted in Fig. 15(c), SHOT incorrectly predicts 2%, 4%, and 10% of the samples of class 7 as class 1, class 3, and class 6, respectively. The results show that OSWA and SHOT have the capability to recognize the unknown fault class, but because they fail to construct the clear classification boundaries, causing some samples of the unknown and known classes to be difficult to distinguish. As seen from Fig. 15(d), IESOFD has a higher FD accuracy for each class, where the FD accuracy of class 7 reaches 97%, which is superior to the other comparison approaches. This indicates that IESOFD has the capability to accurately classify the known and unknown classes.

G. Parameter Sensitivity Analysis

To evaluate the impact of the threshold ω used in the MGPPL strategy on the FD performance of IESOFD, the comparison experiments are performed on different FD tasks of HUST and PHM datasets by using different values of ω.

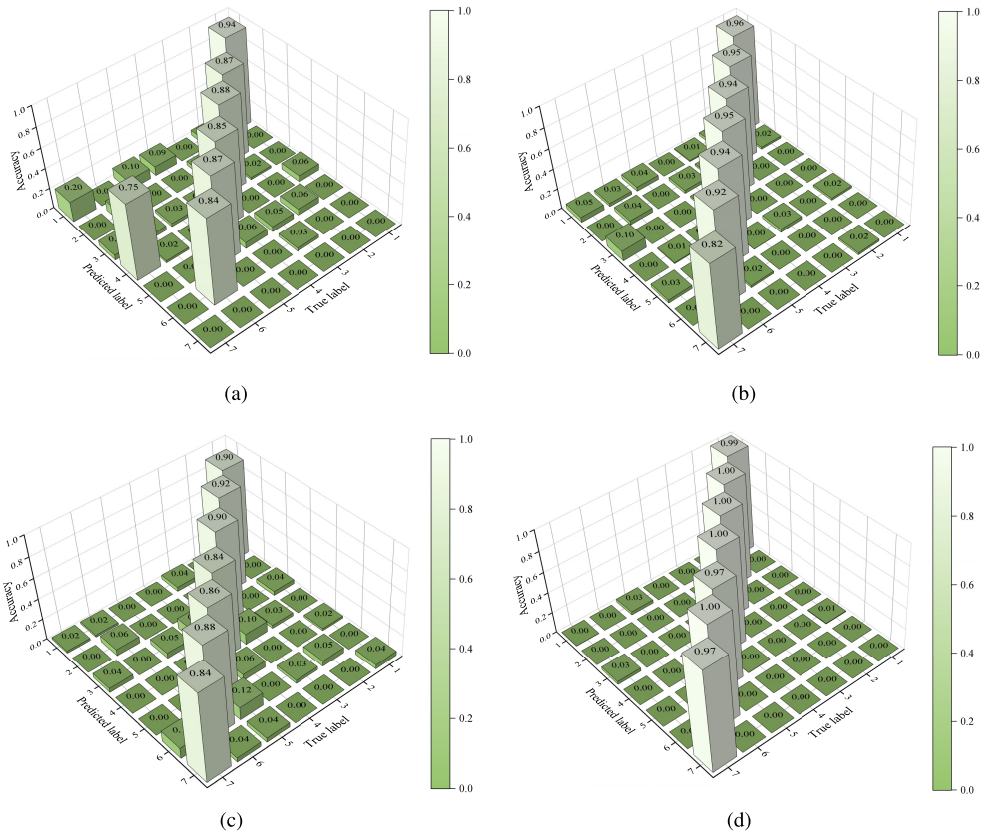


Fig. 15. Confusion matrices of different approaches on the task H2. (a) DANN. (b) OSWA. (c) SHOT. (d) IESOFD.

TABLE IX

OS (%) OBTAINED BY IESOFD WITH DIFFERENT VALUES OF ω ON DIFFERENT FD TASKS OF HUST DATASET

ω	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	Avg.
0.70	97.31 ± 0.68	96.75 ± 0.43	82.64 ± 0.68	75.01 ± 1.24	70.53 ± 1.35	97.51 ± 0.54	95.02 ± 0.57	81.78 ± 0.74	75.77 ± 0.61	71.32 ± 1.04	84.36
0.75	97.74 ± 0.52	96.87 ± 0.68	83.79 ± 0.98	79.68 ± 1.09	76.41 ± 1.14	97.52 ± 0.60	97.17 ± 0.83	83.99 ± 0.89	79.68 ± 1.02	76.41 ± 1.13	86.93
0.80	98.06 ± 0.41	97.31 ± 0.38	85.31 ± 0.53	85.19 ± 1.08	83.21 ± 1.17	98.34 ± 0.59	97.83 ± 0.59	86.32 ± 0.83	86.01 ± 0.90	83.91 ± 1.23	90.15
0.85	**99.89 ± 0.13**	**98.84 ± 0.25**	**95.34 ± 0.41**	**92.45 ± 0.53**	**89.12 ± 0.86**	**99.78 ± 0.11**	**99.76 ± 0.18**	**94.86 ± 0.64**	**93.68 ± 0.89**	**90.44 ± 1.12**	**95.42**
0.90	97.98 ± 0.50	91.10 ± 0.46	89.42 ± 0.69	86.20 ± 0.81	68.79 ± 0.97	97.31 ± 0.73	91.89 ± 0.55	90.22 ± 0.83	87.80 ± 1.06	70.00 ± 1.25	87.07
0.95	96.33 ± 0.66	82.70 ± 0.63	79.61 ± 0.96	77.30 ± 1.20	60.89 ± 0.99	96.43 ± 0.76	83.83 ± 0.68	79.40 ± 0.94	77.01 ± 1.36	62.51 ± 1.28	79.60

TABLE X

OS (%) OBTAINED BY IESOFD WITH DIFFERENT VALUES OF ω ON DIFFERENT FD TASKS OF PHM DATASET

ω	P1	P2	P3	P4	P5	P6	P7	P8	Avg.
0.70	96.05 ± 0.61	92.95 ± 0.53	85.64 ± 0.68	76.01 ± 1.19	95.86 ± 0.32	91.86 ± 0.41	85.14 ± 0.63	76.62 ± 1.02	87.52
0.75	96.63 ± 0.49	95.12 ± 0.58	87.47 ± 0.67	79.94 ± 1.13	96.27 ± 0.27	94.52 ± 0.46	88.67 ± 0.52	80.99 ± 0.83	89.95
0.80	97.02 ± 0.35	96.11 ± 0.42	90.74 ± 0.74	83.05 ± 1.16	96.93 ± 0.33	95.58 ± 0.46	89.47 ± 0.89	82.73 ± 1.25	91.45
0.85	**97.87 ± 0.27**	**96.94 ± 0.35**	**94.23 ± 0.48**	**91.89 ± 0.74**	**97.26 ± 0.22**	**96.31 ± 0.37**	**93.39 ± 0.51**	**90.54 ± 0.82**	**94.80**
0.90	96.55 ± 0.47	90.47 ± 0.56	86.73 ± 0.84	80.04 ± 1.18	96.19 ± 0.44	89.84 ± 0.85	85.95 ± 1.03	78.72 ± 1.32	88.06
0.95	94.89 ± 0.73	85.83 ± 1.23	79.83 ± 1.14	72.69 ± 1.29	94.74 ± 0.65	84.69 ± 0.92	79.41 ± 1.32	71.73 ± 1.42	82.98

To select an appropriate threshold ω, at first, the grid search is performed between 0 and 1 in steps of 0.1. It is found that IESOFD can achieve a relatively high *OS* when the value of ω is between 0.70 and 0.95. Then, the grid search is performed between 0.70 and 0.95 in steps of 0.05. Table IX lists the *OS* obtained by IESOFD with different values of ω on different FD tasks of the HUST dataset. When ω is set to 0.70, 0.75, 0.80, 0.85, 0.90, and 0.95, respectively, the average *OS* achieved by IESOFD is 84.36%, 86.93%, 90.15%, 95.42%, 87.07%, and 79.60%, respectively. Table X presents the *OS* obtained by IESOFD with different values of ω on different FD tasks of the PHM dataset. When ω is set to 0.70, 0.75, 0.80, 0.85, 0.90, and 0.95, the average *OS* achieved by IESOFD is 87.52%, 89.95%, 91.45%, 94.80%, 88.06%, and 82.98%, respectively. When ω is lower than 0.85, the *OS* achieved by IESOFD on different tasks increases with the increase of ω. When ω is higher than 0.85, the *OS* gained by IESOFD on different tasks decreases with the increase of ω. This is mainly because

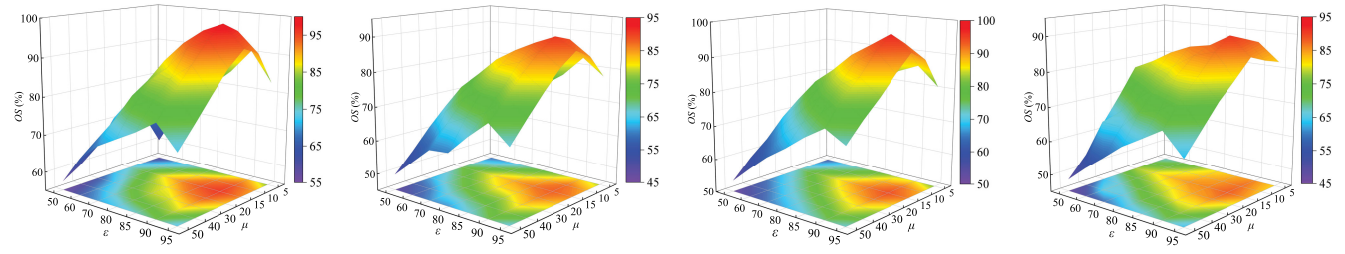


Fig. 16. *OS* achieved by IESOFD using different values of ε and μ on four different FD tasks. (a) H2. (b) H10. (c) P2. (d) P8.

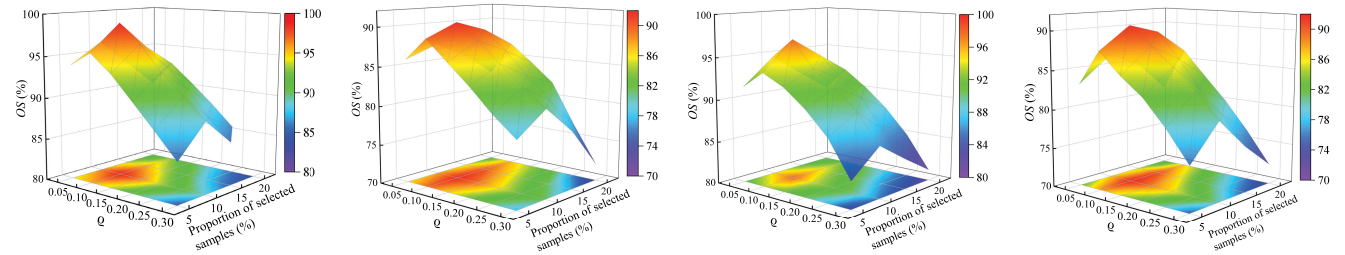


Fig. 17. *OS* achieved by IESOFD with different values of ϱ and different proportions of selected samples on four different FD tasks. (a) H2. (b) H10. (c) P2. (d) P8.

TABLE XI

OS (%) OBTAINED BY IESOFD WITH DIFFERENT VALUES OF *K* ON DIFFERENT FD TASKS OF HUST DATASET

K	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	Avg.
1	95.21 ± 0.49	93.05 ± 0.59	90.93 ± 0.69	87.74 ± 0.84	84.57 ± 1.13	95.04 ± 0.42	92.59 ± 0.50	90.21 ± 0.73	86.99 ± 0.91	83.05 ± 0.94	89.94
2	94.97 ± 0.51	92.38 ± 0.57	89.77 ± 0.74	86.43 ± 1.18	82.43 ± 1.07	94.61 ± 0.50	91.57 ± 0.47	88.91 ± 0.93	85.87 ± 1.14	82.57 ± 1.26	88.95
3	**99.89 ± 0.13**	**98.84 ± 0.25**	**95.34 ± 0.41**	**92.45 ± 0.53**	**89.12 ± 0.86**	**99.78 ± 0.11**	**99.76 ± 0.18**	**94.86 ± 0.64**	**93.68 ± 0.89**	**90.44 ± 1.12**	**95.42**
4	94.36 ± 0.37	92.75 ± 0.64	88.98 ± 0.87	86.25 ± 0.91	81.14 ± 1.15	94.38 ± 0.37	91.83 ± 0.58	89.21 ± 0.79	86.90 ± 1.25	83.01 ± 1.41	88.88
5	95.48 ± 0.43	93.62 ± 0.60	90.33 ± 0.94	88.58 ± 1.25	85.31 ± 1.08	95.87 ± 0.58	92.86 ± 0.64	89.64 ± 0.85	94.13 ± 1.05	84.22 ± 1.33	91.00

the value of ω is too low or too high, which will reduce the quality of pseudo-labels, thus affecting the FD performance of IESOFD. When ω is set to 0.85, IESOFD achieves the highest *OS* on each task, thus, ω is selected as 0.85.

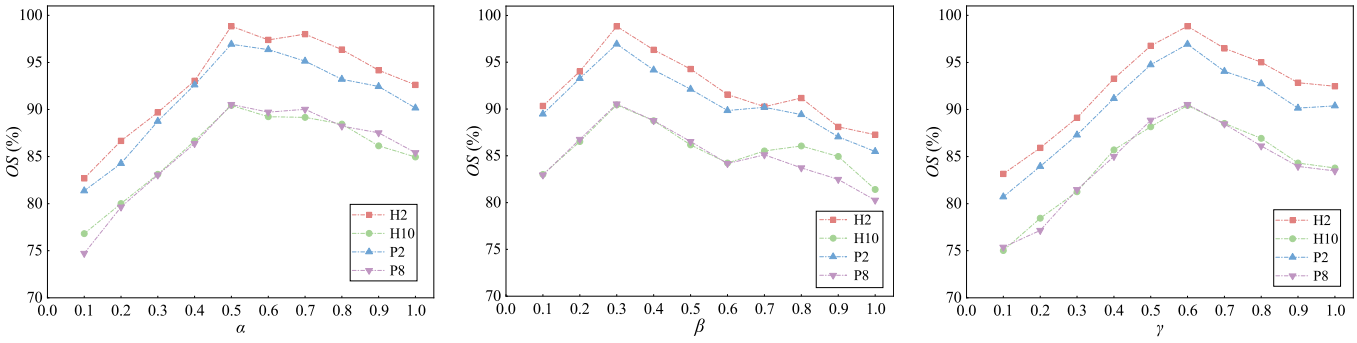
To analyze the impact of the thresholds of ε and μ adopted in MGPPL strategy to determine the percentiles P_ε and P_μ on the FD performance of IESOFD, the comparison experiments are carried out on four different FD tasks, where ε is set to 50, 60, 70, 80, 85, 90, and 95, respectively, and μ is set to 5, 10, 15, 20, 30, 40, and 50, respectively. Fig. 16 demonstrates the *OS* achieved by IESOFD with different values of ε and μ on four different FD tasks. As shown in Fig. 16(a)–(d), when ε and μ are set to 90 and 15, respectively, IESOFD achieves the highest *OS* on four different FD tasks. The results are mainly because the number of selected samples of known and unknown classes is different due to different values of ε and μ. When ε is too high and μ is too low, the selected known and unknown class samples are too few, causing the features of known and unknown classes cannot be fully learned. When ε is too low and μ is too high, too many known and unknown class samples are selected, which reduces the quality of the selected samples, causing the overfitting problem. Therefore, ε and μ are selected as 90 and 15, respectively.

To evaluate the impact of the threshold ϱ and the proportion of selected samples used in DCL strategy on the FD performance of IESOFD, the comparison experiments are conducted on four different FD tasks, where ϱ is set to 0.05, 0.10, 0.15, 0.20, 0.25, and 0.30, respectively, and the proportion of selected samples is set to 5%, 10%, 15%, and 20%, respectively. Fig. 17 depicts the *OS* achieved by IESOFD with different values of ϱ and different proportions of selected samples on four different FD tasks. As illustrated in Fig. 17(a)–(d), when the value of ϱ is lower than 0.10, the *OS* achieved by IESOFD on different FD tasks increases with the increase of ϱ. When the value of ϱ is higher than 0.10, the *OS* achieved by IESOFD on different FD tasks decreases with the increase of ϱ. When the value of ϱ and the proportion of selected samples are set to 0.1 and 10%, respectively, IESOFD achieves the highest *OS* on four different tasks. This is mainly because when the value of ϱ and the proportion of selected samples are different, the selected difficult class samples are different. Too many or too few difficult class samples and the quality of different difficult class samples will affect the clustering effect between different classes, thereby affecting the FD performance of IESOFD. Therefore, the value of ϱ and the proportion of selected samples are selected as 0.1 and 10%, respectively.

TABLE XII

OS (%) OBTAINED BY IESOFD WITH DIFFERENT VALUES OF K ON DIFFERENT FD TASKS OF PHM DATASET

K	P1	P2	P3	P4	P5	P6	P7	P8	Avg.
1	89.76 ± 0.54	88.47 ± 0.62	86.89 ± 0.72	84.14 ± 0.88	89.62 ± 0.31	87.84 ± 0.58	85.73 ± 0.72	82.52 ± 0.92	86.87
2	93.39 ± 0.48	91.88 ± 0.57	88.52 ± 0.69	86.01 ± 0.94	92.75 ± 0.39	90.26 ± 0.62	87.94 ± 0.80	85.41 ± 0.89	89.52
3	**97.87 ± 0.27**	**96.94 ± 0.35**	**94.23 ± 0.48**	**91.89 ± 0.74**	**97.26 ± 0.22**	**96.31 ± 0.37**	**93.39 ± 0.51**	**90.54 ± 0.82**	**94.80**
4	93.02 ± 0.38	91.47 ± 0.58	88.23 ± 0.72	85.74 ± 0.83	92.48 ± 0.40	89.88 ± 0.54	87.49 ± 0.66	85.02 ± 0.79	89.16
5	89.27 ± 0.41	87.64 ± 0.53	85.91 ± 0.72	84.03 ± 0.87	89.37 ± 0.51	87.26 ± 0.75	85.39 ± 0.82	82.31 ± 0.94	86.40



(a) (b) (c)

Fig. 18. *OS* obtained by IESOFD with different values of three weighting parameters on the FD tasks of H2, H10, P2, and P8. (a) α ($\beta = 0.3$ and $\gamma = 0.6$). (b) β ($\alpha = 0.5$ and $\gamma = 0.6$). (c) γ ($\alpha = 0.5$ and $\beta = 0.3$).

To analyze the impact of the value of the parameter K (i.e., the number of nearest neighbors in the kNN graph) on the FD performance of IESOFD, the comparison experiments are conducted on different FD tasks of HUST and PHM datasets by setting K to 1, 2, 3, 4, and 5, respectively. Table XI presents the *OS* obtained by IESOFD with different values of K on different FD tasks of the HUST dataset. When K is set to 1, 2, 3, 4, and 5, respectively, the average *OS* achieved by IESOFD is 89.94%, 88.95%, 95.42%, 88.88%, and 91.00%, respectively. Table XII gives the *OS* obtained by IESOFD with different values of K on different FD tasks of the PHM dataset. When K is set to 1, 2, 3, 4, and 5, respectively, the average *OS* achieved by IESOFD is 86.87%, 89.52%, 94.80%, 89.16%, and 86.40%, respectively. A smaller value of K can easily lead to overconfidence in a neighboring sample, whereas a larger value of K can easily lead to the selection of samples with too large distances, which will affect the FD performance of IESOFD. When K is set to 3, IESOFD achieves the best FD results, thus K is selected as 3.

The calculation of the total loss of the proposed IESOFD involves the three weighting parameters of α, β, and γ, all of which are determined using the grid search method, where α, β, and γ are all searched from {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0}. Fig. 18 shows the *OS* obtained by IESOFD with different values of α, β, and γ on four different FD tasks. As shown in Fig. 18(a), when $\beta = 0.3$ and $\gamma = 0.6$, the *OS* obtained on different FD tasks grow rapidly with α rising between 0.1 and 0.5. As α rises between 0.5 and 1.0, the *OS* obtained on different FD tasks is constantly changing, but all are lower than the *OS* obtained by $\alpha = 0.5$. As given in Fig. 18(b), when $\alpha = 0.5$ and $\gamma = 0.6$, the *OS* obtained by $\beta = 0.3$ is the highest. As seen in Fig. 18(c), when $\alpha = 0.5$ and $\beta = 0.3$, the *OS* obtained by $\gamma = 0.6$ is the highest. Therefore, the three weighting parameters of α, β, and γ are selected as 0.5, 0.3, and 0.6, respectively.

TABLE XIII

OS (%) OBTAINED BY IESOFD ON DIFFERENT FD TASKS OF DIFFERENT DATASETS UNDER DIFFERENT NOISE LEVELS

Task	0% noise	25% noise	50% noise	75% noise	100% noise
H1	99.89 ± 0.13	99.75 ± 0.12	99.56 ± 0.13	99.31 ± 0.15	99.04 ± 0.16
H2	98.84 ± 0.25	98.69 ± 0.25	98.42 ± 0.27	98.17 ± 0.28	97.96 ± 0.27
H3	95.34 ± 0.41	95.22 ± 0.39	95.01 ± 0.44	94.76 ± 0.47	94.52 ± 0.50
H4	92.45 ± 0.53	92.21 ± 0.58	91.95 ± 0.60	91.66 ± 0.62	91.43 ± 0.64
H5	89.12 ± 0.86	88.96 ± 0.90	88.73 ± 0.88	88.51 ± 0.92	88.27 ± 0.95
H6	99.78 ± 0.11	99.51 ± 0.15	99.26 ± 0.14	99.02 ± 0.17	98.75 ± 0.20
H7	99.76 ± 0.18	99.48 ± 0.18	99.24 ± 0.19	98.98 ± 0.22	98.67 ± 0.23
H8	94.86 ± 0.64	94.65 ± 0.67	94.41 ± 0.69	94.16 ± 0.72	93.97 ± 0.74
H9	93.68 ± 0.64	93.42 ± 0.69	93.18 ± 0.71	92.97 ± 0.75	92.63 ± 0.77
H10	90.44 ± 1.12	90.29 ± 1.18	90.03 ± 1.21	89.86 ± 1.17	89.61 ± 1.20
Avg.	95.42	95.22	94.98	94.74	94.48
P1	98.87 ± 0.27	98.52 ± 0.30	98.27 ± 0.32	98.07 ± 0.33	97.93 ± 0.34
P2	96.94 ± 0.35	96.73 ± 0.32	96.49 ± 0.36	96.21 ± 0.37	95.96 ± 0.40
P3	94.23 ± 0.48	94.02 ± 0.51	93.75 ± 0.49	93.53 ± 0.53	93.25 ± 0.55
P4	91.89 ± 0.74	91.63 ± 0.71	91.39 ± 0.76	91.13 ± 0.78	90.84 ± 0.82
P5	97.26 ± 0.22	97.01 ± 0.23	96.82 ± 0.26	96.65 ± 0.28	96.36 ± 0.30
P6	96.31 ± 0.37	96.07 ± 0.34	95.74 ± 0.39	95.42 ± 0.42	95.14 ± 0.45
P7	93.39 ± 0.51	93.11 ± 0.56	92.78 ± 0.58	92.57 ± 0.62	92.22 ± 0.64
P8	90.54 ± 0.82	90.32 ± 0.81	90.13 ± 0.86	89.85 ± 0.84	89.57 ± 0.87
Avg.	94.80	94.68	94.42	94.18	93.90

H. Performance Analysis Under Noise Environment

To verify the noise robustness of the proposed IESOFD, a series of experiments is conducted by adding the white Gaussian noise with intensities of 25%, 50%, 75%, and 100% to the vibration signals. Table XIII presents the *OS* obtained by IESOFD on different FD tasks of different datasets under different noise levels. As shown in Table XIII, IESOFD exhibits strong robustness under different noise levels. Although the *OS* achieved by IESOFD decreases with increasing noise

intensity on different FD tasks, IESOFD still achieves good FD accuracies on different FD tasks of different datasets. On HUST dataset, IESOFD achieves the average accuracies of 95.42%, 95.22%, 94.98%, 94.74%, and 94.48% under 0%, 25%, 50%, 75%, and 100% noises, respectively. On PHM dataset, IESOFD obtains the average accuracies of 94.80%, 94.68%, 94.42%, 94.18%, and 93.90% under 0%, 25%, 50%, 75%, and 100% noises, respectively. On HUST and PHM datasets, the average FD accuracies obtained under 100% noise than those obtained under 0% noise are reduced by 0.94% and 0.90%, respectively. This shows that IESOFD can still achieve high FD accuracy in the presence of severe noise interference, verifying its strong noise robustness.

V. Conclusion

In this article, a novel IESOFD approach is proposed, which can accurately identify the known and unknown fault classes in the TD without directly accessing the SD data. The proposed approach provides a new paradigm of FD under privacy protection for industrial measurement systems, enhancing the ability to identify unknown fault classes and adapt to complex working conditions. By transferring the pretrained GCN-based source model to the TD, the target model is obtained and fine-tuned, effectively realizing the knowledge transfer from SD to TD. The proposed MGPPL strategy effectively achieves the multigroup partitioning of known and unknown classes according to the pseudo-labels of the TD samples. In the proposed MGPPL strategy, the information entropy of the known and unknown class sample sets is minimized and maximized, respectively. This effectively separates the unknown fault classes in the TD and enhances the reliability of the pseudo-labels. This strategy improves the measurement accuracy and robustness of fault identification in the open-set scenario. The designed DCL strategy achieves more accurate identifications of the known and unknown fault classes, and enhances the measurement and recognition ability for difficult samples, effectively improving the classification robustness of the model in complex measurement environments. The effectiveness of the proposed approach is verified by extensive experiments. The proposed approach achieves average FD accuracies of 95.42% and 94.80% on the cross-condition FD tasks with different openness of HUST bearing and PHM gearbox datasets, respectively, outperforming the other comparison approaches.

Although the proposed IESOFD approach is capable of dealing with the scenario where both the SD data cannot be directly accessed and the unknown fault classes appear on the TD, the following limitations remain.

1) IESOFD can effectively address the problem that the unknown fault classes appear on the TD. However, the private classes may appear on the SD, making it more difficult to accurately identify known and unknown fault classes.
2) In most applications, the number of samples for different fault classes is relatively balanced. However, in actual industrial productions, the problem of imbalanced samples may arise. This causes the model to be biased toward the majority classes during training, thereby making it difficult to construct reliable class decision boundaries.
3) The hyperparameters adopted in the proposed MGPPL and DCL strategies are determined through the grid search.

Considering these limitations, future work will primarily focus on the following aspects.

1) A dual-classification pseudo-label strategy will be introduced, which can use the different class prototypes and the classification confidence to distinguish private classes in the SD and TD, aiming to achieve accurate fault classification on the TD.
2) An adaptive weight and decision boundary optimization mechanism will be designed. It can dynamically adjust the class weightings during training to make the model focus more on the minority classes, and construct clearer and more reliable decision boundaries. It can realize robust recognition of different fault classes under the scenario where the class distribution is extremely imbalanced.
3) The relationship between the physical mechanism of RMFD and some key hyperparameters will be deeply studied, and a parameter adaptive optimization strategy based on physical information will be established.

References

[1] D. Neupane, M. R. Bouadjenek, R. Dazeley, and S. Aryal, "Data-driven machinery fault diagnosis: A comprehensive review," *Neurocomputing*, vol. 627, Apr. 2025, Art. no. 129588.

[2] Y. Qin, Q. Qian, J. Luo, and H. Pu, "Deep joint distribution alignment: A novel enhanced-domain adaptation mechanism for fault transfer diagnosis," *IEEE Trans. Cybern.*, vol. 53, no. 5, pp. 3128–3138, May 2023.

[3] Z. Wang, J. Xuan, and T. Shi, "Domain reinforcement feature adaptation methodology with correlation alignment for compound fault diagnosis of rolling bearing," *Expert Syst. Appl.*, vol. 262, Mar. 2025, Art. no. 125594.

[4] J. Li, C. Shen, J. Shi, C. Li, D. Wang, and Z. Zhu, "Bi-generator cooperative domain adversarial neural network for bearing fault diagnosis," *IEEE Sensors J.*, vol. 24, no. 7, pp. 10584–10593, Apr. 2024.

[5] M. Ghorvei, M. Kavianpour, M. T. H. Beheshti, and A. Ramezani, "Spatial graph convolutional neural network via structured subdomain adaptation and domain adversarial learning for bearing fault diagnosis," *Neurocomputing*, vol. 517, pp. 44–61, Jan. 2023.

[6] Y. Sun, H. Tao, and V. Stojanovic, "Pseudo-label guided dual classifier domain adversarial network for unsupervised cross-domain fault diagnosis with small samples," *Adv. Eng. Informat.*, vol. 64, Mar. 2025, Art. no. 102986.

[7] Y. Yu, H. R. Karimi, P. Shi, R. Peng, and S. Zhao, "A new multi-source information domain adaption network based on domain attributes and features transfer for cross-domain fault diagnosis," *Mech. Syst. Signal Process.*, vol. 211, Apr. 2024, Art. no. 111194.

[8] Y. Xie, F. Wan, Y. Hua, M. Yang, and X. Qing, "Prototype-oriented domain adaptation for rotating machinery fault diagnosis," *Measurement*, vol. 253, Sep. 2025, Art. no. 117544.

[9] X. Chen, H. Shao, Y. Xiao, S. Yan, B. Cai, and B. Liu, "Collaborative fault diagnosis of rotating machinery via dual adversarial guided unsupervised multi-domain adaptation network," *Mech. Syst. Signal Process.*, vol. 198, Sep. 2023, Art. no. 110427.

[10] L. Wan, J. Ning, Y. Li, and C. Li, "Efficient cross-domain fault diagnosis via distributed multi-source domain deep transfer learning," *Meas. Sci. Technol.*, vol. 36, no. 1, Jan. 2025, Art. no. 016165.

[11] B. Liu, G. Li, J. He, S. Liu, T. Wang, and R. Shi, "Dual branch feature matching guided multi-source domain adaptive ensemble network for rotating machinery fault diagnosis," *Adv. Eng. Inf.*, vol. 64, Mar. 2025, Art. no. 103061.

[12] J. Jiao, H. Li, T. Zhang, and J. Lin, "Source-free adaptation diagnosis for rotating machinery," *IEEE Trans. Ind. Informat.*, vol. 19, no. 9, pp. 9586–9595, Sep. 2023.

[13] J. Tian, J. Zhang, Y. Jiang, S. Wu, H. Luo, and S. Yin, "A novel generalized source-free domain adaptation approach for cross-domain industrial fault diagnosis," *Rel. Eng. Syst. Saf.*, vol. 243, Mar. 2024, Art. no. 109891.

[14] W. Cao, Z. Meng, Z. Pan, H. Chen, and F. Fan, "A two-stage source-free domain adaptation method for fault diagnosis of wind turbine gearboxes," *Meas. Sci. Technol.*, vol. 36, no. 9, Sep. 2025, Art. no. 096122.

[15] J. Wang, H. Shao, Y. Xiao, and B. Liu, "SFDA-T: A novel source-free domain adaptation method with strong generalization ability for fault diagnosis," *Adv. Eng. Informat.*, vol. 62, Oct. 2024, Art. no. 102903.

[16] Q. Li, B. Tang, L. Deng, and P. Zhu, "Source-free domain adaptation framework for fault diagnosis of rotation machinery under data privacy," *Rel. Eng. Syst. Saf.*, vol. 238, Oct. 2023, Art. no. 109468.

[17] R. Liu, W. Ma, F. Kuang, J. Guo, and N. Zhao, "A source free robust domain adaptation approach with pseudo-labels uncertainty estimation for rolling bearing fault diagnosis under limited sample conditions," *Knowl.-Based Syst.*, vol. 304, Nov. 2024, Art. no. 112443.

[18] Z. Liu, X. Zhong, H. Shao, and B. Liu, "Graph structure learning guided multi-source-free domain adaptation for mechanical fault diagnosis," *Adv. Eng. Informat.*, vol. 68, Nov. 2025, Art. no. 103769.

[19] A. Yu, B. Cai, Q. Wu, M. M. García, J. Li, and X. Chen, "Source-free domain adaptation method for fault diagnosis of rotation machinery under partial information," *Rel. Eng. Syst. Saf.*, vol. 248, Aug. 2024, Art. no. 110181.

[20] Q. Gao, T. Huang, K. Zhao, H. Shao, and B. Jin, "Multi-source weighted source-free domain transfer method for rotating machinery fault diagnosis," *Expert Syst. Appl.*, vol. 237, Mar. 2024, Art. no. 121585.

[21] Y. Lin, Y. Wang, M. Zhang, and M. Zhao, "A robust source-free unsupervised domain adaptation method based on uncertainty measure and adaptive calibration for rotating machinery fault diagnosis," *Rel. Eng. Syst. Saf.*, vol. 253, Jan. 2025, Art. no. 110516.

[22] C. Ma, X. Tu, G. Zhou, Y. Huang, and X. Ding, "Source-free cross-domain fault diagnosis of rotating machinery using the Siamese framework," *Knowl.-Based Syst.*, vol. 300, Sep. 2024, Art. no. 112179.

[23] H. Li, Z. Liu, J. Lin, J. Jiao, T. Zhang, and H. Pan, "Calibrated source-free adaptation for intelligent diagnosis," *Mech. Syst. Signal Process.*, vol. 229, Apr. 2025, Art. no. 112582.

[24] J. Jiao, T. Zhang, H. Li, H. Liu, and J. Lin, "Source-free black-box adaptation for machine fault diagnosis," *IEEE Trans. Ind. Informat.*, vol. 21, no. 4, pp. 3366–3375, Jan. 2025.

[25] M. Xia, H. Shao, D. Williams, S. Lu, L. Shu, and C. W. de Silva, "Intelligent fault diagnosis of machinery using digital twin-assisted deep transfer learning," *Rel. Eng. Syst. Saf.*, vol. 215, Nov. 2021, Art. no. 107938.

[26] A. Kumar, R. Kumar, J. Xiang, Z. Qiao, Y. Zhou, and H. Shao, "Digital twin-assisted AI framework based on domain adaptation for bearing defect diagnosis in the centrifugal pump," *Measurement*, vol. 235, Aug. 2024, Art. no. 115013.

[27] C. Fang, Q. Chang, X. Hu, W. Zhou, and X. Meng, "A digital twin-enabled domain adaptation network for cross-space fault diagnosis of roller bearings," *Mech. Syst. Sig. Process.*, vol. 236, Aug. 2025, Art. no. 113053.

[28] P. Liang, J. Tian, S. Wang, and X. Yuan, "Multi-source information joint transfer diagnosis for rolling bearing with unknown faults via wavelet transform and an improved domain adaptation network," *Rel. Eng. Syst. Saf.*, vol. 242, Feb. 2024, Art. no. 109788.

[29] J. Mei, M. Zhu, W. Liu, M. Fu, and Q. Tang, "Conditional variational encoder classifier for open set fault classification of rotating machinery vibration signals," *IEEE Trans. Ind. Informat.*, vol. 20, no. 3, pp. 3038–3049, Mar. 2024.

[30] J. Wang and R. Zhang, "Open-set fault diagnosis based on 1D-ResNet with fusion of cross-class and extreme information for out-of-distribution detection," *IEEE Trans. Instrum. Meas.*, vol. 74, pp. 1–9, 2025.

[31] Z. Zhu, G. Chen, and G. Tang, "Domain adaptation with multi-adversarial learning for open-set cross-domain intelligent bearing fault diagnosis," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–11, 2023.

[32] C. Weng, B. Lu, L. Chen, X. Zhao, and W. Huang, "A novel attention-free FCformer network with dynamic subdomain adaptation for open-set fault diagnosis of motor bearing," *Comput. Ind.*, vol. 173, Dec. 2025, Art. no. 104357.

[33] Y. Jin et al., "A novel open-set cross-domain fault diagnosis method based on improved PACDAN," *IEEE Trans. Instrum. Meas.*, vol. 74, pp. 1–15, 2025.

[34] C. Pan, Z. Shang, L. Tang, H. Cheng, and W. Li, "Open-set domain adaptive fault diagnosis based on supervised contrastive learning and a complementary weighted dual adversarial network," *Mech. Syst. Signal Process.*, vol. 222, Jan. 2025, Art. no. 111780.

[35] J. Zheng et al., "Open-set domain adaptation for scene classification using multi-adversarial learning," *ISPRS J. Photogramm. Remote Sens.*, vol. 208, pp. 245–260, Feb. 2024.

[36] C. Weng, B. Lu, L. Chen, X. Zhao, and W. Huang, "A novel progressive domain separation network with multi-metric ensemble quantification for open set fault diagnosis of motor bearings," *Adv. Eng. Informat.*, vol. 64, Mar. 2025, Art. no. 103060.

[37] Y. Li, X. Xu, L. Hu, K. Sun, and M. Han, "A centroid contrastive multi-source domain adaptation method for fault diagnosis with category shift," *Measurement*, vol. 248, May 2025, Art. no. 116801.

[38] L. Wang, Y. Gao, X. Li, and L. Gao, "Self-supervised-enabled open-set cross-domain fault diagnosis method for rotating machinery," *IEEE Trans. Ind. Informat.*, vol. 20, no. 8, pp. 10314–10324, Aug. 2024.

[39] C. Zhao, E. Zio, and W. Shen, "Domain generalization for cross-domain fault diagnosis: An application-oriented perspective and a benchmark study," *Rel. Eng. Syst. Saf.*, vol. 245, May 2024, Art. no. 109964.

[40] B. Zhang, W. Li, X.-L. Li, and S.-K. Ng, "Intelligent fault diagnosis under varying working conditions based on domain adaptive convolutional neural networks," *IEEE Access*, vol. 6, pp. 66367–66384, 2018.

[41] Y. Ganin et al., "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 59, pp. 1–35, 2016.

[42] W. Zhang, X. Li, H. Ma, Z. Luo, and X. Li, "Open-set domain adaptation in machinery fault diagnostics (don't short) using instance-level weighted adversarial learning," *IEEE Trans. Ind. Informat.*, vol. 17, no. 11, pp. 7445–7455, Nov. 2021.

[43] J. Liang, D. Hu, and J. Feng, "Do we really need to access the source data? Source hypothesis transfer for unsupervised domain adaptation," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 6028–6039.

Lanjun Wan received the B.S. and M.S. degrees in computer science and technology from Hunan University of Technology, Zhuzhou, China, in 2005 and 2009, respectively, and the Ph.D. degree in circuits and systems from Hunan University, Changsha, China, in 2016.

He is currently an Associate Professor with the School of Computer Science and Artificial Intelligence, Hunan University of Technology. His research interests include industrial big data analysis, industrial equipment fault diagnosis, high-performance computing, and parallel computing.

Hui Chen received the B.S. degree in electronic information engineering from Changsha Normal University, Changsha, China, in 2023. She is currently pursuing the M.S. degree in computer technology with Hunan University of Technology, Zhuzhou, China.

Her research interests include industrial big data analysis and industrial equipment fault diagnosis.

Hongwei Tan received the B.S. degree in software engineering from Hunan University of Information Technology, Changsha, China, in 2023. He is currently pursuing the M.S. degree in computer science and technology with Hunan University of Technology, Zhuzhou, China.

His research interests include industrial big data analysis and industrial equipment fault diagnosis.

Jiaen Ning received the B.S. degree in computer science and technology and the M.S. degree in computer technology from Hunan University of Technology, Zhuzhou, China, in 2022 and 2025, respectively. He is currently pursuing the Ph.D. degree in information and communication engineering with Central South University, Changsha, China.

His research interests include industrial big data analysis and industrial equipment fault diagnosis.

Wei Ni received the B.S. and Ph.D. degrees in communication engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2003 and 2012, respectively.

He is currently an Assistant Professor with the School of Computer Science and Artificial Intelligence, Hunan University of Technology, Zhuzhou, China. His research interests include industrial artificial intelligence and industrial equipment fault diagnosis.

Keqin Li (Fellow, IEEE) received the B.S. degree in computer science from Tsinghua University, Beijing, China, in 1985, and the Ph.D. degree in computer science from the University of Houston, Houston, TX, USA, in 1990.

He is a SUNY Distinguished Professor with The State University of New York, New Paltz, NY, USA, and a National Distinguished Professor with Hunan University, Changsha, China. He has authored or co-authored more than 1230 journal articles, book chapters, and refereed conference papers. He holds nearly 80 patents announced or authorized by Chinese National Intellectual Property Administration. He is among the world's top few most influential scientists in parallel and distributed computing, regarding single-year impact (ranked #2) and career-long impact (ranked #3) based on a composite indicator of the Scopus citation database. He is listed in Scilit Top-Cited Scholars (2023–2025) and is among the top 0.02% out of over 20 million scholars worldwide based on top-cited publications in the last ten years. He is listed in ScholarGPS Highly Ranked Scholars (2022–2025) and is among the top 0.002% out of over 30 million scholars worldwide based on a composite score of three ranking metrics for research productivity, impact, and quality in the recent five years.

Dr. Li is a member of the SUNY Distinguished Academy. He is an AAAS Fellow, an AAIA Fellow, an ACIS Fellow, and an AIIA Fellow. He is a member of the European Academy of Sciences and Arts. He is a member of Academia Europaea (Academician of the Academy of Europe). He received the IEEE TCCLD Research Impact Award from the IEEE CS Technical Committee on Cloud Computing in 2022 and the IEEE TCSVC Research Innovation Award from the IEEE CS Technical Community on Services Computing in 2023. He won the IEEE Region 1 Technological Innovation Award (Academic) in 2023. He was a recipient of the 2022–2023 International Science and Technology Cooperation Award and the 2023 Xiaoxiang Friendship Award of Hunan Province, China.