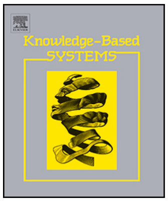
Knowledge-Based Systems 284 (2024) 111288

Contents lists available at ScienceDirect

Knowledge-Based Systems

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



Intelligent fault diagnosis via ring-based decentralized federated transfer learning

Lanjun Wan [a,*], Jiaen Ning [a], Yuanyuan Li [b], Changyun Li [a], Keqin Li [c]

[a] School of Computer Science, Hunan University of Technology, Zhuzhou 412007, China
[b] School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China
[c] Department of Computer Science, State University of New York, New Paltz, NY, 12561, USA

ARTICLE INFO

Keywords:
Communication optimization
Fault diagnosis
Federated learning
Rotating machinery
Transfer learning

ABSTRACT

Federated transfer learning (FTL) can effectively address the data silos and domain shift that exist in data-driven rotating machinery fault diagnosis (RMFD). However, in FTL used for RMFD, the huge communication overhead, idle waiting between the source clients and the target client, and negative transfer caused by model aggregation are all pressing challenges. Therefore, a ring-based decentralized federated transfer learning (RDFTL) method for intelligent fault diagnosis is proposed. Firstly, a ring-based decentralized federated transfer learning framework is designed, which can be fully integrated with the bandwidth-optimal Ring-AllReduce algorithm, thereby greatly reducing the communication overhead. Secondly, an asynchronous domain adaptation strategy is proposed, which can effectively avoid idle waiting between the source clients and the target client in the collaborative model training, thereby improving the overall training efficiency of FTL. Thirdly, a multi-perspective distribution discrepancy aggregation (MPDDA) strategy is proposed to alleviate the negative transfer caused by model aggregation. The diagnosis performance of the local model of a source client on the target domain is evaluated from the three perspectives of statistical distance, domain adversarial, and stability, and these three evaluation metrics are jointly used to determine the aggregation weights, which can effectively improve the diagnosis performance of the global model. Finally, a series of experiments are carried out to verify the effectiveness of the proposed method. The results demonstrate that the proposed method can obtain a cross-domain fault diagnosis model with excellent performance in RMFD with data privacy at a fast training speed.

1. Introduction

Rolling bearing is a critical component in many pieces of rotating machinery, and its failure often leads to the shutdown of the equipment, thereby affecting production efficiency [1]. Through the accurate fault diagnosis, the reliability of equipment and operation safety can be improved, while the maintenance costs can be reduced. In recent years, the data-driven deep learning methods [2] have been extensively applied in the field of RMFD, achieving superior diagnosis performance due to their robust feature learning and representation capabilities. The data-driven deep learning methods usually require a large amount of labeled fault data and only perform well in the specific diagnosis scenarios. Once there is a lack of large amounts of high-quality labeled training data or when switching diagnosis scenarios, the diagnosis performance will be significantly affected.

In actual industrial productions, although the long-term condition monitoring can be performed on rotating machinery, the monitoring data are typically acquired under normal conditions. Only a minuscule amount of data are acquired under fault conditions, and the manually labeled fault data are even scarcer. This results in a single client (i.e., an enterprise or factory that generates the data) typically being unable to utilize a large amount of high-quality training data to construct a fault diagnosis model. In such a scenario, a promising approach is to collect labeled fault data from multiple clients that have similar machinery equipment, thereby collaboratively constructing a superior-performance fault diagnosis model. However, due to potential conflicts of interest and privacy concerns, enterprises or factories are often reluctant to share their private data, which is known as the data silos.

To address the data silos, federated learning [3] emerged as a solution. Federated learning can utilize the private data of different clients to train a global model collaboratively, and it can guarantee the data security of each client. Specifically, each client that participates in federated learning trains a local model using its private data, and exchanges the local model in plaintext or encrypted form to obtain a global model. Federated learning reduces the risk of private data

[*] Corresponding author.
E-mail address: wanlanjun@hut.edu.cn (L. Wan).

https://doi.org/10.1016/j.knosys.2023.111288
Received 17 September 2023; Received in revised form 25 November 2023; Accepted 7 December 2023
Available online 10 December 2023

leakage while improving model performance, and has been widely studied and applied in the field of fault diagnosis. Zhang et al. [4] designed a federated learning method for RMFD, which uses self-supervised learning and a dynamic verification strategy to construct an effective global RMFD model. Ma et al. [5] implemented real-time updates of local models and asynchronous update of the global model in federated learning for RMFD, which enhances real-time diagnosis. Yu et al. [6] designed a federated learning framework for rolling bearing fault diagnosis (RBFD) suitable for cloud–edge environments. A shallow convolutional autoencoder is trained within each client, and a global fault classifier is trained on the server, which effectively alleviates the computational burden of each client. Geng et al. [7] put forward a weighted aggregation strategy based on F1-scores and accuracy differences for the problem of imbalanced fault categories in RBFD. Lin et al. [8] put forward a federated learning approach for transformer fault classification, which utilizes a hierarchical aggregation strategy to provide a personalized model that is more tailored to the local diagnosis task for each client. The above studies indicate that federated learning has been successfully applied in fault diagnosis with data privacy. However, the existing research often assumes that the data distributions of multiple clients are the same. In actual industrial productions, the data from different clients are typically collected from various equipment or under different working conditions, which means that there are usually distribution discrepancies between data from different clients. The above problem is termed domain shift, which restricts the diagnosis performance of the global model.

Transfer learning [9] can well deal with domain shift, and its core goal is to apply the knowledge learned from the source domain to the target domain. Domain adaptation is one of the most popular methods in transfer learning, and can effectively alleviate the problem of inconsistent data distributions between domains. The local maximum mean discrepancy (LMMD), margin disparity discrepancy (MDD), and weighted conditional MMD are adopted in [10,11], and [12] to measure the distribution discrepancies between domains respectively, so as to make the source domain and target domain have more similar distributions in the feature space. Wan et al. [13] introduced multiple domain discriminators to achieve adversarial training, which can extract domain-invariant features with stronger representation ability. He et al. [14] designed an adversarial domain adaptation framework combining manifold learning and similar structure discrimination, which can effectively alleviate the negative transfer and improve the target classification accuracy. He et al. [15] put forward an asymmetrical MDD approach to effectively extract the common features between domains, and adopted an outlier sample extraction algorithm to reduce the negative transfer caused by outlier samples in the source domain. Han et al. [16] utilized the gradients and weights of the model in each iteration to dynamically determine transferable and non-transferable parameters aiming at the problem of parameter transferability in domain adaptation, which achieves robust unsupervised domain adaptation. The aforementioned studies can effectively improve the performance of domain adaptation. However, similar to traditional deep learning, transfer learning requires a large amount of data from source and target domains, and it assumes that these data are publicly available. This implies that transfer learning also faces the data silos in practical applications.

Federated transfer learning [17], as an integration of federated learning and transfer learning, can well deal with domain shift while protecting data privacy. FTL has been widely applied in the field of mechanical fault diagnosis. Yang et al. [18] designed an FTL approach for RBFD, which employs a federated averaging aggregation strategy based on shared layers to enhance diagnosis performance. Zhang and Li [19] put forward an adversarial networks-based FTL approach, which effectively improves the accuracy of cross-domain RMFD in federated learning. Zhang and Li [20] utilized prior distributions in FTL to minimize domain discrepancies, which can better extract domain-invariant features for RBFD while protecting data privacy. Zhao et al. [21] combined multi-source domain adaptation with federated learning to develop an FTL framework, which performs well in cross-domain RMFD with data privacy. Liu et al. [22] proposed an FTL approach based on broad learning and active learning for addressing domain shift and incremental domain adaptation problems in cross-domain RMFD, which can effectively select high-quality target domain data and incrementally update the global model. Chen et al. [23] put forward a discrepancy-based weighted federated averaging (DWFA) method for the problem of performance differences of local models in FTL-based RMFD, which reduces the impact of low-quality local models on the global model by weighing the contributions of different local models. Zhang et al. [24] presented a decentralized FTL method based on blockchain for RMFD, which employs a committee consensus strategy for optimizing the aggregation of local models. Wang et al. [25] devised an FTL approach considering that some clients have low-quality data, which effectively mitigates the negative effect of low-quality data using a low-quality knowledge filtering strategy.

The existing research has successfully explored the application of FTL in the mechanical fault diagnosis with data privacy, but the research on the following problems is still insufficient.

The first problem is the negative transfer caused by model aggregation. In practical applications, due to the distribution discrepancies in data provided by different clients, the diagnosis performance of local models from different source clients may vary significantly on the target client. This requires accurate measurement of the diagnosis performance of local models from different source clients during the aggregation process, aiming to fully enhance the diagnosis performance of the global model.

The second problem is the idle waiting between clients in the collaborative model training. In the existing research on FTL, the high-level features of the target domain are usually utilized for domain adaptation. The two processes of local training on the source clients and feature extraction on the target client cannot be executed in parallel due to dependencies, namely there is idle waiting between the source clients and the target client, which reduces the overall training efficiency of FTL.

The third problem is the huge communication overhead of FTL. In the actual industrial environment, the clients that participate in federated learning usually have limited bandwidth and cross-regional distribution. To cope with the complex and changeable working conditions as well as potential new fault patterns in industrial productions, the client models often have complex network structures, which results in the need to transmit a large number of weight vectors in FTL. In the traditional federated learning based on client–server architecture, the communication overhead increases linearly with the number of clients, and the communication efficiency is also limited to the network and memory bandwidth of the server. The huge communication overhead would seriously affect the rapid training and updating of RMFD models, thereby reducing the real-time performance and reliability of RMFD models.

In summary, FTL can effectively address the data silos and domain shift that exist in data-driven RMFD. However, in FTL used for RMFD, the huge communication overhead, idle waiting, and negative transfer are pressing challenges. Therefore, a ring-based decentralized federated transfer learning approach for intelligent fault diagnosis is proposed.

The main contributions of this paper are as follows.

(1) A ring-based decentralized federated transfer learning framework is designed, which is characterized by weighting before transmitting and aggregating while transmitting. The proposed framework can be fully integrated with the bandwidth-optimal Ring-AllReduce algorithm, which can greatly reduce the communication overhead and avoid the problem that the model transmission overhead grows linearly with the number of clients.

(2) An asynchronous domain adaptation strategy is proposed, which can effectively avoid idle waiting between the source clients and the target client in the collaborative model training, thereby improving the overall training efficiency of FTL.

(3) A multi-perspective distribution discrepancy aggregation strategy is proposed to alleviate the negative transfer caused by model aggregation. The diagnosis performance of the local model of a source client on the target domain is evaluated from the three perspectives of statistical distance, domain adversarial, and stability, and these three evaluation metrics are jointly used to determine the aggregation weights, which can effectively improve the diagnosis performance of the global model.
(4) A series of experiments are carried out to verify the effectiveness of the proposed method. The results demonstrate that the proposed method can obtain a cross-domain fault diagnosis model with excellent performance in RMFD with data privacy at a fast training speed.

The rest of the paper is organized as follows. Section 2 introduces the basic theory. Section 3 describes the proposed method. Section 4 presents the experimental results and analysis. Section 5 gives conclusions and future work.

2. Basic theory

2.1. Federated learning

Federated learning [3] is a distributed machine learning paradigm that considers data privacy protection, and its goal is to train a global model with strong generalization from the data provided by different clients. In federated learning, model-related information can be exchanged among participants in the form of plaintext, encryption or adding noise, but the training data do not leave the local area. This exchange method does not expose the local data to other participants, reducing the risk of data leakage. The three steps of local training, model transmission, and model aggregation will be executed iteratively after each participant obtains an initialized model until the preset stopping condition is reached.

In federated learning, there are usually two or more participants to collaboratively train a shared global model, and the performance of the global model should be as close as possible to that of the ideal model. The ideal model is the model obtained through the centralized training without data privacy restrictions. Assuming that V_{sum} and V_{fed} are the performance measures (e.g., accuracy) of the centralized training model M_{sum} and federated training model M_{fed}, respectively, the performance loss of the federated model can be defined as

$$\delta = \left| V_{\text{sum}} - V_{\text{fed}} \right|, \tag{1}$$

where δ is usually a small non-negative floating-point number. In particular, when $\delta = 0$, it indicates that M_{fed} and M_{sum} have the same performance.

2.2. Transfer learning

The main goal of transfer learning [9] is to transfer knowledge learned from the source task to the target task to solve domain shift and the lack of labeled data in the target domain. The three key concepts in transfer learning are domain, task, and domain adaptation.

Domain: The domain $\mathscr{D}$ consists of the feature space $\mathscr{X}$ and the marginal probability distribution $P(X)$, and different domains usually have different marginal probability distributions, where X is the set of samples and it is defined as $X = \{\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_n\} \in \mathscr{X}$.

Task: In a specific domain $\mathscr{D} = \{\mathscr{X}, P(X)\}$, the task $\mathscr{T} = \{y, f(\cdot)\}$ consists of the label space y and the mapping function $f(\cdot)$, where $f(\cdot)$ can predict the labels of the samples.

Domain adaptation: The goal of domain adaptation is to make the distributions of source and target domain data as close as possible in the feature space by learning the domain adaptation function $G(\cdot)$ to map the source and target domain data.

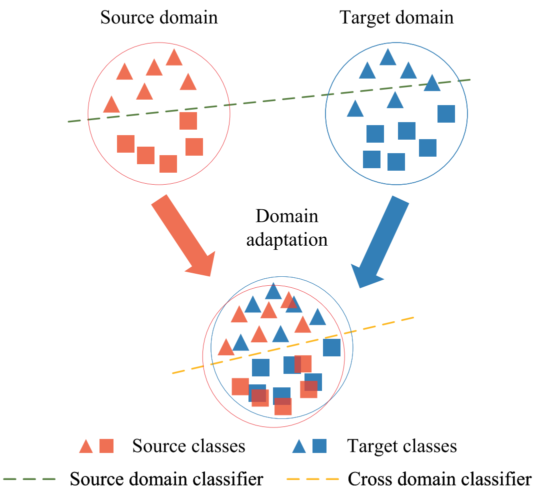


Fig. 1. Illustration of domain adaptation.

As shown in Fig. 1, if the source domain classifier is directly applied to the target domain without performing domain adaptation, some prediction errors will be generated due to domain shift. The classifier can be gradually adjusted and optimized by performing domain adaptation, thereby significantly improving the classification accuracy of the source domain model on the target domain.

In RBFD based on domain adaptation, assuming that there are N bearing vibration datasets, where $N-1$ labeled datasets D^{S_1}, D^{S_2}, …, $D^{S_{N-1}}$ are regarded as $N-1$ source domains and one unlabeled dataset is regard as the target domain. The ith source domain dataset can be represented as $D^{S_i} = \{X^{S_i}, Y^{S_i}\}$, where $X^{S_i} = \left\{\mathrm{x}_j^{S_i}\right\}_{j=1}^{n_{S_i}}$ and $Y^{S_i} = \left\{\mathrm{y}_j^{S_i}\right\}_{j=1}^{n_{S_i}}$ denote the n_{S_i} samples and labels of the ith source domain respectively, and $Y^{S_i} \in \{1, 2, \ldots, C\}$ represents the C different bearing health conditions. The unlabeled target domain dataset can be denoted as $D^T = \{X^T\}$, where $X^T = \left\{\mathrm{x}_j^T\right\}_{j=1}^{n_T}$ denotes the n_T samples of the target domain. $P\left(X^{S_i}\right)$ denotes the marginal distribution of the ith source domain, $Q\left(Y^{S_i} \middle| X^{S_i}\right)$ denotes the conditional distribution of the ith source domain, $P\left(X^T\right)$ denotes the marginal distribution of the target domain, and $Q\left(Y^T \middle| X^T\right)$ denotes the conditional distribution of the target domain. In actual RBFD scenarios, D^{S_i} and D^T usually come from different types of bearings or different working conditions, implying that $D^{S_i} \neq D^T$, $D^{S_i} \neq D^{S_j}$, $P\left(X^{S_i}\right) \neq P\left(X^T\right)$, $P\left(X^{S_i}\right) \neq P\left(X^{S_j}\right)$, $Q\left(Y^{S_i} \middle| X^{S_i}\right) \neq Q\left(Y^T \middle| X^T\right)$, and $Q\left(Y^{S_i} \middle| X^{S_i}\right) \neq Q\left(Y^{S_j} \middle| X^{S_j}\right)$, where $1 \leq i, \; j \leq N-1$. The optimization objective of domain adaptation can be represented as

$$\min\left\{\mathrm{Dist}\left(P\left(X^{S_i}\right), P\left(X^T\right)\right) + \mathrm{Dist}\left(Q\left(Y^{S_i} \mid X^{S_i}\right), Q\left(Y^T \mid X^T\right)\right)\right\}, \tag{2}$$

where $\mathrm{Dist}(\cdot)$ is a function used to measure the inter-domain distribution discrepancy. By minimizing the discrepancy between $Q\left(Y^{S_i} \middle| X^{S_i}\right)$ and $Q\left(Y^T \middle| X^T\right)$ and the discrepancy between $P\left(X^{S_i}\right)$ and $P\left(X^T\right)$, it ensures that the knowledge learned from the source domain performs well on the target domain.

3. Proposed method

3.1. Design of RDFTL framework

Under the premise of ensuring the accuracy of collaborative fault diagnosis based on FTL for rotating machinery, a ring-based decentralized federated transfer learning framework is designed to reduce the

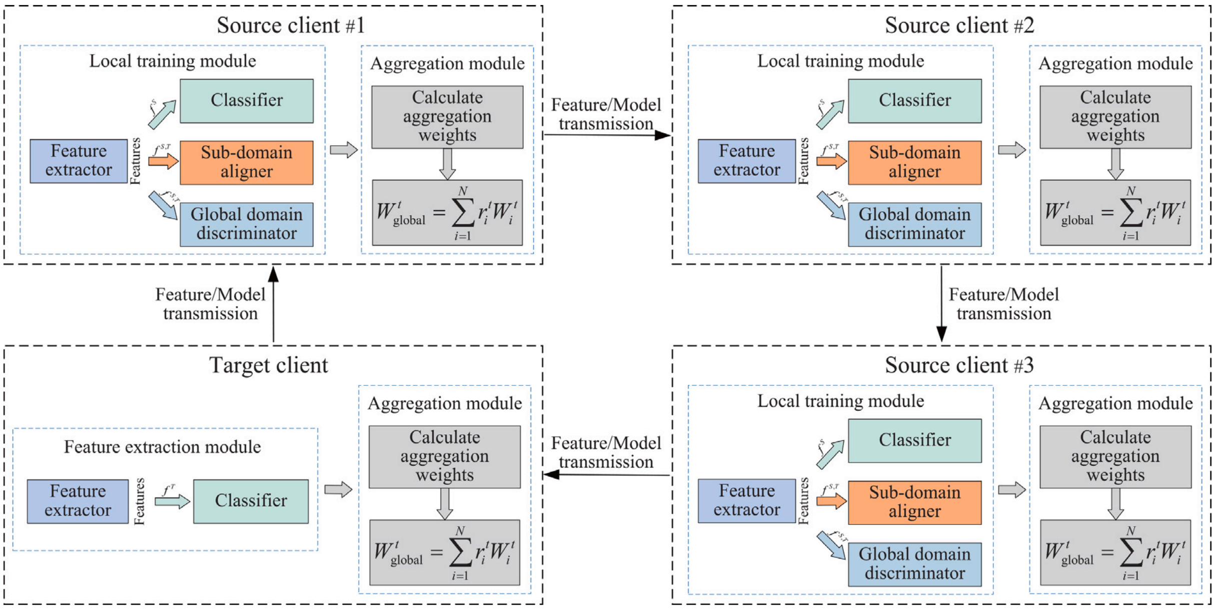

Source client #1
Local training module
Classifier
Feature extractor
Features
f^S
$f^{S,T}$
Sub-domain aligner
Global domain discriminator
Aggregation module
Calculate aggregation weights
$W_{\text{global}}^t = \sum_{i=1}^{N} r_i^t W_i^t$
Feature/Model transmission
Source client #2
Local training module
Classifier
Feature extractor
Features
Sub-domain aligner
Global domain discriminator
Aggregation module
Calculate aggregation weights
$W_{\text{global}}^t = \sum_{i=1}^{N} r_i^t W_i^t$
Feature/Model transmission
Feature/Model transmission
Target client
Feature extraction module
Feature extractor
Features
f^T
Classifier
Aggregation module
Calculate aggregation weights
$W_{\text{global}}^t = \sum_{i=1}^{N} r_i^t W_i^t$
Feature/Model transmission
Source client #3
Local training module
Classifier
Feature extractor
Features
Sub-domain aligner
Global domain discriminator
Aggregation module
Calculate aggregation weights
$W_{\text{global}}^t = \sum_{i=1}^{N} r_i^t W_i^t$

Fig. 2. Illustration of the proposed RDFTL framework.

communication overhead of FTL, as shown in Fig. 2. In the proposed framework, each source client consists of a local training module and an aggregation module, and the target client consists of a feature extraction module and an aggregation module. The local training module is responsible for training and optimizing the local model. The aggregation module is responsible for aggregating the local models of all clients to obtain a global model. The local training module of each source client consists of a feature extractor, a global domain discriminator, a sub-domain aligner, and a classifier, where both the global domain discriminator and the sub-domain aligner require to utilize the features from source and target domains $f^{S,T}$ to reduce inter-domain distribution discrepancies, and the classifier utilizes the source domain features f^S for predicting faults. The feature extraction module of the target client is composed of a feature extractor and a classifier, where the target domain features f^T are input into the classifier. The bandwidth-optimal Ring-AllReduce algorithm [26] is adopted in RDFTL framework, where each client only needs to perform model transmission with two adjacent clients, which can effectively avoid the problem that the model transmission overhead grows linearly with the number of clients.

The traditional aggregation process in FTL is as follows.

Step 1: Transmit local models. In the traditional client–server architecture [23], the local models are typically transmitted to the server.

Step 2: Calculate aggregation weights. The aggregation weight of each local model is calculated on the server.

Step 3: Model aggregation. Each local model is weighted and aggregated on the server to obtain a global model.

In the traditional aggregation process, all local models need to be transmitted to the server, so the communication efficiency is limited to the network and memory bandwidth of the server, and the communication overhead increases linearly with the number of clients. The traditional aggregation process is changed in the proposed RDFTL framework, which is characterized by weighting before transmitting and aggregating while transmitting. Specifically, first, the aggregation weights are calculated, the local models are weighted, and then the model aggregation is completed in the process of transmitting the local models. The proposed framework can be fully integrated with the bandwidth-optimal Ring-AllReduce algorithm, which can greatly reduce the communication overhead and avoid the problem that the model transmission overhead increases linearly with the number of clients.

The model aggregation process of the proposed RDFTL framework is as follows.

Step 1: Calculate aggregation weights. The aggregation weights of the local models are calculated on each client using the MPDDA strategy proposed in Section 3.4.

Step 2: Weight local models. The local model of each client is weighted by

$$W_i^{\text{weight}} = r_i W_i^{\text{local}}, \tag{3}$$

where W_i^{local} denotes the local model of the ith source client, r_i is the corresponding aggregation weight, and W_i^{weight} represents the weighted local model.

Step 3: Aggregating while transmitting. The process of aggregating local models while transmitting local models includes two stages: Scatter-Reduce and Allgather.

Step 3.1: Scatter-Reduce stage. The model parameters of each client are equally divided into N blocks. During the jth communication, the ith client sends its $((i-j)\%N+1)$th block to its right neighbor, receives the $((i-j-1)\%N+1)$th block from its left neighbor, and accumulates the received block to its own $((i-j)\%N+1)$th block, where $1 \le i \le N$ and $1 \le j \le N-1$. After $N-1$ Scatter-Reduce operations, the ith client merges the $((i-N)\%N+1)$th block from each other client to obtain a complete block. Note that during each Scatter-Reduce operation, each client will accumulate the received block to the corresponding block to obtain a new block, meaning that each client only needs to transmit the data of one block size each time.

Step 3.2: Allgather stage. During each Allgather operation, the clients exchange complete blocks with each other. During the jth communication, the ith client sends its $((i-j)\%N+1)$th block to its right neighbor, receives the $((i-j-1)\%N+1)$th block from its left neighbor, and replaces its own $((i-j-1)\%N+1)$th block with the received block, where $1 \le i \le N$ and $N \le j \le 2(N-1)$. After $N-1$ Allgather operations, all clients will have N complete blocks, forming a complete global model.

As illustrated in Fig. 3, there are one target client and two source clients, and the local model parameters of each client are equally divided into three blocks. During the Scatter-Reduce stage, the clients gradually exchange and merge the blocks of each other. After two Scatter-Reduce operations, the first client collects the second block from each other client, the second client collects the third block from each other client, and the third client collects the first block from each

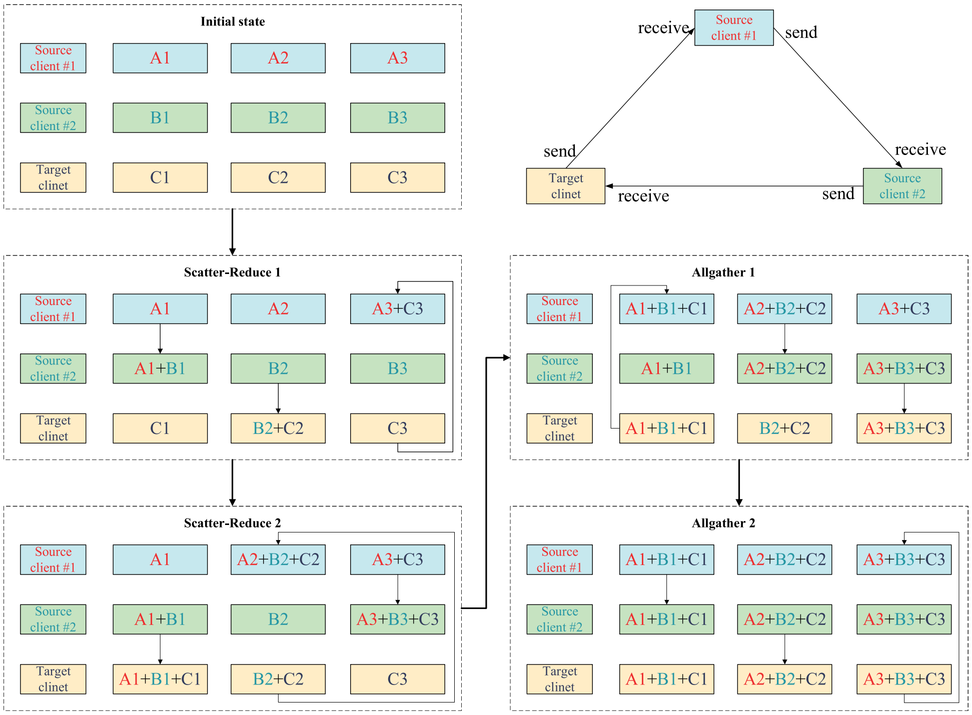


Fig. 3. Illustration of the model aggregation based on the Ring-AllReduce algorithm.

other client. During the Allgather stage, the clients exchange their own complete blocks. After two Allgather operations, each client has all complete blocks.

Suppose that there are N clients, the size of the local model is D, the network bandwidth is B, and the network latency can be ignored. Under the client–server architecture, the size of data transmitted from all the clients to the server is $D\times N$, and the size of the data transmitted from the server to all the clients is $D\times N$. Therefore, the total communication overhead is $(2\times D\times N)/B$. In the proposed RDFTL framework, the Scatter-Reduce stage and the Allgather stage require $N-1$ data transmissions respectively, and the size of data transmitted each time is D/N. Hence, the total communication overhead is $2\frac{D}{N}(N-1)/B$. It is evident that the communication overhead of RDFTL framework is much smaller than that of the client–server architecture. More significantly, when N is large, the communication overhead of RDFTL framework can be approximately $2D/B$, which means that the communication overhead does not increase linearly with the number of clients.

3.2. Local training

The local training based on RDFTL framework is shown in Fig. 4. The bearing vibration signals are first input into the feature extractor G_f to obtain the high-level features, and then the high-level features are sent to the classifier G_c for fault classification. To ensure that the model has excellent diagnosis performance on different data distributions, the sub-domain aligner and the global domain discriminator G_d are used for domain adaptation.

Taking the ith source client as an example, the feature $G_f(\mathrm{x}_j^{S_i})$ and the fault classification result $G_c(G_f(\mathrm{x}_j^{S_i}))$ can be obtained by forward-propagation of the training data. According to the fault classification results and the true labels, the classification loss L_c can be calculated by

$$L_c=-\frac{1}{n_{S_i}}\sum_{j=1}^{n_{S_i}}\mathrm{y}_j^{S_i}\log\left(G_c\left(G_f\left(\mathrm{x}_j^{S_i}\right)\right)\right), \tag{4}$$

where $\mathrm{x}_j^{S_i}$ denotes the jth sample of the ith source domain, $\mathrm{y}_j^{S_i}$ is the corresponding true label, and n_{S_i} denotes the number of samples of the ith source domain.

Due to the distribution discrepancies of vibration signals in different domains, the diagnosis performance of the source domain model on the target domain is reduced. To improve the cross-domain fault diagnosis performance of the model, a global domain discriminator is set between each pair of source and target domains. As shown in Fig. 4, a gradient reversal layer (GRL) is inserted between the feature extractor and the global domain discriminator. GRL will not affect the forward-propagation in training, but will reverse the symbols of the gradients received from the global domain discriminator. The original intention of the global domain discriminator is to back-propagate the gradients to the feature extractor to further improve the domain discrimination ability. However, due to the effect of the GRL, these gradients will be reversed, which makes the feature extractor generate the inter-domain invariant features that are more difficult to be distinguished, greatly increasing the difficulty of domain discrimination. The aforementioned process is the domain adversarial training process, which is actually a kind of minimax game. The forward-propagation and back-propagation of the GRL can be defined as

$$\begin{aligned}&R(x)=x,\\&\frac{dR(x)}{dx}=-\lambda I,\end{aligned} \tag{5}$$

where $R(x)$ represents the GRL, I indicates the identity matrix, and λ is the scale parameter.

Through the adversarial training between the feature extractor and the global domain discriminator, the feature extractor can extract domain-invariant features with excellent performance on both the source and target domains. The global domain discrimination loss L_d^g can be calculated by

$$L_d^g=-\frac{1}{n_{S_i}+n_T}\sum_{j=1}^{n_{S_i}+n_T}\mathrm{y}_j^d\log\left(G_d\left(feature_j^{S_i,T}\right)\right), \tag{6}$$

where $feature_j^{S_i,T}$ denotes the high-level feature of the jth sample from the mixture of the target domain and the ith source domain, y_j^d denotes the true domain label of the jth sample, and $\mathrm{y}_j^d\in\{0,1\}$.

Only performing the marginal distribution alignment (i.e., global domain alignment) may lead to the problem of sub-domain misalignment in complex transfer diagnosis tasks. Therefore, it is necessary

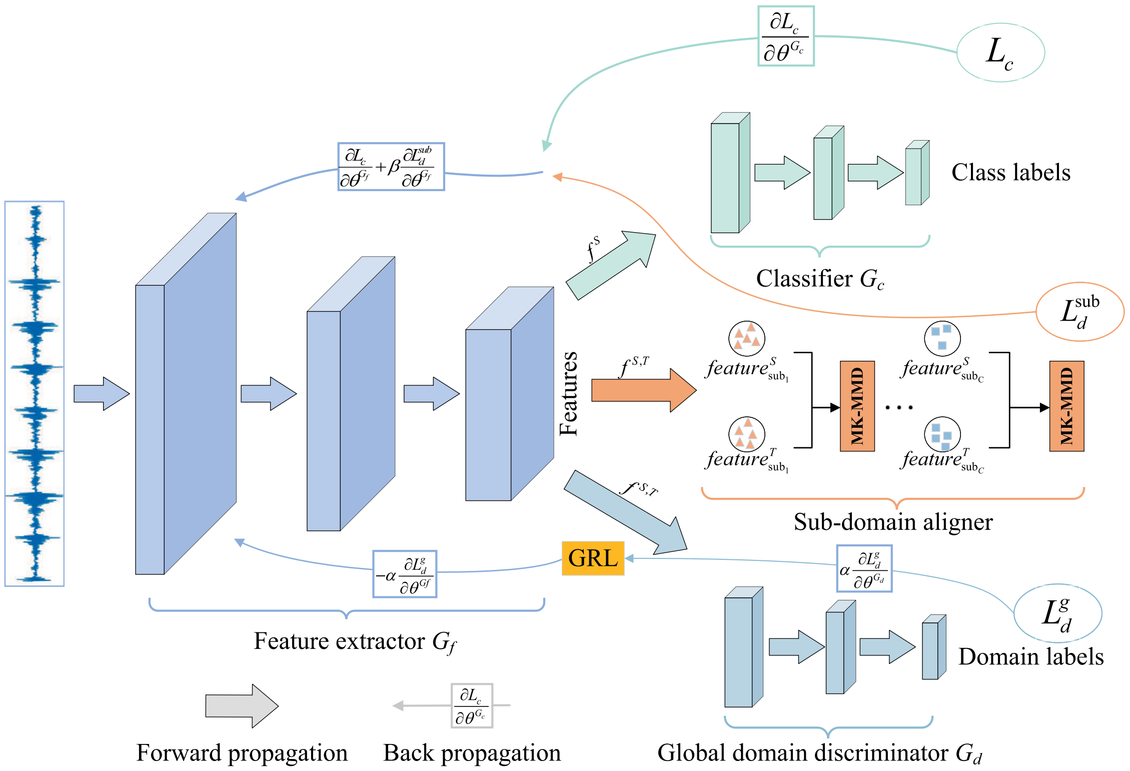


Fig. 4. Illustration of the local training based on RDFTL framework.

to further reduce the conditional distribution discrepancy. Unlike the global domain alignment, in the conditional distribution alignment, the multi-kernel MMD (MK-MMD) distance is adopted to minimize the distribution discrepancy of each fault category (i.e., sub-domain) of different domains. Since the data of the target domain are not labeled, the global model is used to set the pseudo-labels $\hat{Y}^T$ for all samples of the target domain:

$$\hat{Y}^T = \underset{k\in\{1,\ 2,\ \ldots,\ C\}}{\operatorname{argmax}} \left(\frac{e^{z_k}}{\sum_{c=1}^{C} e^{z_c}} \right), \tag{7}$$

where z_k denotes the output result of the kth neuron in the last layer of the model, and C denotes the total number of fault categories. The sub-domain alignment loss L_d^{sub} can be calculated by

$$L_d^{\mathrm{sub}} = \frac{1}{C}\sum_{c=1}^{C} \left\| E_{P\left(X_c^{S_i}\right)}\left[\phi\left(G_f\left(X_c^{S_i}\right)\right)\right] - E_{P\left(X_c^T\right)}\left[\phi\left(G_f\left(X_c^T\right)\right)\right] \right\|_{\mathscr{H}_k}^2, \tag{8}$$

where $\phi(\cdot)$ represents the kernel function that can map data to a high-level feature space, $\mathscr{H}_k$ is the reproducing kernel Hilbert space, X_c^T and $X_c^{S_i}$ represent the sample set belonging to the cth fault category on the target domain and the ith source domain respectively, and $P\left(X_c^T\right)$ and $P\left(X_c^{S_i}\right)$ represent the probability distribution of the cth fault category on the target domain and the ith source domain respectively. The pseudo-labels can be set for X_c^T by Eq. (7).

The total loss function of the local model after using the GRL is calculated by

$$L = L_c - \alpha L_d^g + \beta L_d^{\mathrm{sub}}, \tag{9}$$

where α and β are the adjustable weights, $0 \le \alpha,\ \ \beta \le 1$, and $\alpha + \beta = 1$.

The purpose of the classifier is to accurately identify fault categories, therefore its optimization objective is to minimize the classification loss, which is defined as

$$\hat{\theta}^{G_c} = \arg\min_{\theta^{G_c}} L_c. \tag{10}$$

The purpose of the global domain discriminator is to accurately identify which domain the features come from, therefore its optimization objective is to minimize the global domain discrimination loss, which is defined as

$$\hat{\theta}^{G_d} = \arg\min_{\theta^{G_d}} L_d^g, \tag{11}$$

As the training goes on, it becomes increasingly difficult for the global domain discriminator to identify the features, because the distributions of the features extracted from the source domains and those of the features extracted from the target domain become increasingly similar. Due to the gradients of domain discrimination received by the feature extractor are reversed, one of the purposes of the feature extractor is to maximize L_d^g. Therefore, the optimization objective of the feature extractor is defined as

$$\hat{\theta}^{G_f} = \arg\{\min_{\theta^{G_f}} L_c + L_d^{\mathrm{sub}},\ \max_{\theta^{G_f}} L_d^g\}, \tag{12}$$

where θ^{G_c} denotes the classifier parameter, θ^{G_d} denotes the global domain discriminator parameter, and θ^{G_f} denotes the feature extractor parameter. The parameter update of the classifier is defined as

$$\theta^{G_c} = \theta^{G_c} - \gamma\left(\frac{\partial L_c}{\partial \theta^{G_c}}\right), \tag{13}$$

the parameter update of the global domain discriminator is defined as

$$\theta^{G_d} = \theta^{G_d} - \gamma\left(\alpha\frac{\partial L_d^g}{\partial \theta^{G_d}}\right), \tag{14}$$

and the parameter update of the feature extractor is defined as

$$\theta^{G_f} = \theta^{G_f} - \gamma\left(\frac{\partial L_c}{\partial \theta^{G_f}} - \alpha\frac{\partial L_d^g}{\partial \theta^{G_f}} + \beta\frac{\partial L_d^{\mathrm{sub}}}{\partial \theta^{G_f}}\right), \tag{15}$$

where γ represents the learning rate.

3.3. Asynchronous domain adaptation strategy

The model training process of the traditional FTL is shown in Fig. 5. Firstly, the local models of the source clients are forwarded to the target

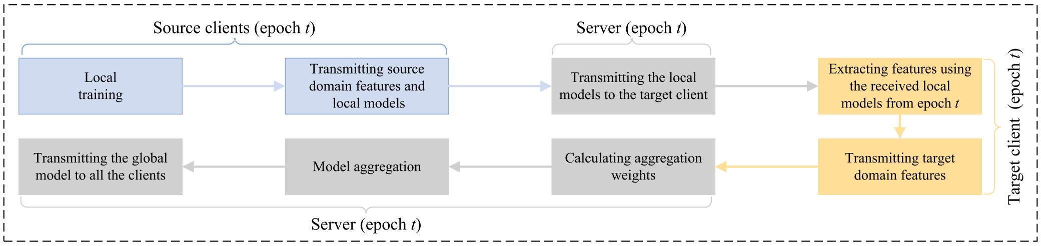

Source clients (epoch t)
Server (epoch t)
Local training
Transmitting source domain features and local models
Transmitting the local models to the target client
Extracting features using the received local models from epoch t
Target client (epoch t)
Transmitting target domain features
Calculating aggregation weights
Model aggregation
Transmitting the global model to all the clients
Server (epoch t)

Fig. 5. Model training process of the traditional FTL.

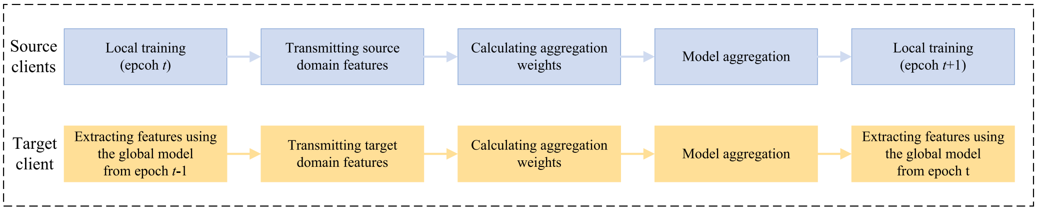

Source clients
Local training (epcoh t)
Transmitting source domain features
Calculating aggregation weights
Model aggregation
Local training (epcoh t+1)
Target client
Extracting features using the global model from epoch t-1
Transmitting target domain features
Calculating aggregation weights
Model aggregation
Extracting features using the global model from epoch t

Fig. 6. Model training process based on asynchronous domain adaptation strategy in the proposed RDFTL framework.

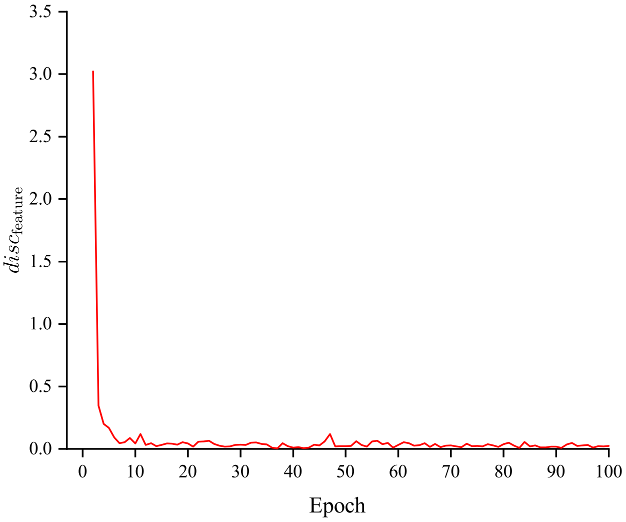


Fig. 7. Feature discrepancy of the target domain between two consecutive global updates.

client via the server to extract the target domain features. Secondly, the target client transmits the extracted target domain features to the server. Finally, the server performs model aggregation and transmits the global model to all clients. This will lead to idle waiting among the source clients, the target client, and the server, and will produce huge communication overhead. In the proposed RDFTL framework, there is no idle waiting between the clients and the server because there is no central server. To avoid idle waiting between the source clients and the target client, an asynchronous domain adaptation strategy is proposed. The basic idea of this strategy is to use the target domain features extracted by the global model obtained from the previous epoch as the target domain features that need to be used on the source clients at the current epoch.

The model training process based on the asynchronous domain adaptation strategy in the proposed RDFTL framework is shown in Fig. 6. At the tth epoch, all source clients perform local training. At the same time, the target client uses the global model obtained from the $(t-1)$th epoch to extract the target domain features and use them as the target domain features that need to be used on the source clients at the tth epoch. This allows the local training of the source clients and the feature extraction of the target client to be executed in parallel, which can effectively avoid idle waiting between clients in collaborative model training. In addition, there is no need to transmit the local models of the source clients to the target client, and it only needs to transmit the features extracted by one global model instead of the features extracted by all the local models. This reduces the additional communication overhead and improves the overall training efficiency of FTL. In the proposed asynchronous domain adaptation strategy, the source clients do not perform domain adaptation at the first two epochs of local training, and only local classification loss is considered. Starting from the third epoch, the source clients perform domain adaptation using the target domain features obtained from the previous epoch.

In the proposed asynchronous domain adaptation strategy, the target domain features obtained from the previous epoch are used at the current epoch, which does not affect the domain adaptation. This is because the discrepancy of target domain features between two consecutive global updates in FTL is usually tiny. Experiments are conducted on the task T5 listed in Table 5 to obtain the discrepancy of target domain features between two consecutive global updates. The MK-MMD distance-based feature discrepancy $disc_{\text{feature}}$ between the target domain features $feature_T^{t-1}$ obtained from the $(t-1)$th epoch and the target domain features $feature_T^{t}$ obtained from the tth epoch can be calculated by

$$disc_{\text{feature}} = \| E[\phi(feature_T^{t-1})] - E[\phi(feature_T^{t})] \|^2_{\mathcal{H}_k}. \tag{16}$$

Fig. 7 shows the feature discrepancy of the target domain between two consecutive global updates during 100 epochs of training. Except for the discrepancies in the first five epochs are significant, the average discrepancy for all other epochs is 0.03, and the maximum discrepancy is only 0.12. The results indicate that the discrepancy of target domain features between two consecutive global updates is tiny. Therefore, the asynchronous domain adaptation strategy does not affect domain adaptation.

3.4. Multi-perspective distribution discrepancy aggregation strategy

The model aggregation strategy of FTL has a significant impact on the diagnosis performance of the global model, and an inappropriate model aggregation strategy usually leads to serious negative transfer. The most classical model aggregation strategy is the federated averaging (FedAvg) strategy [27]. This strategy weights the local models of N

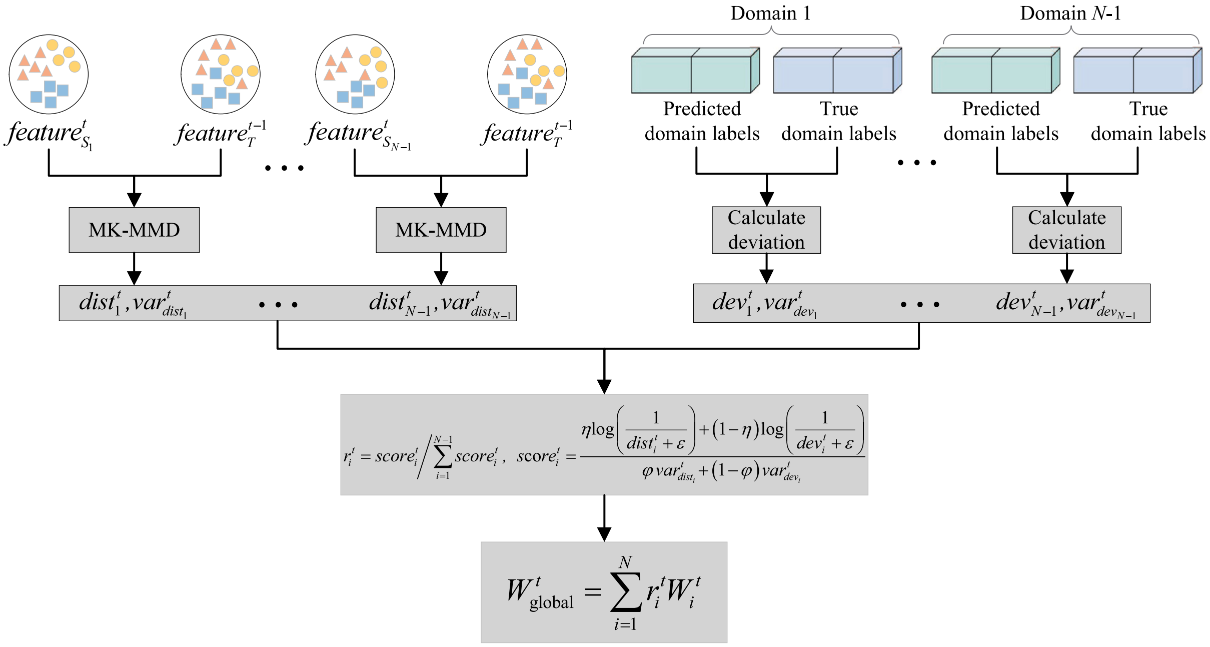

Domain 1
Domain N-1
$feature^t_{S_1}$
$feature^{t-1}_T$
$feature^t_{S_{N-1}}$
$feature^{t-1}_T$
Predicted domain labels
True domain labels
Predicted domain labels
True domain labels
MK-MMD
MK-MMD
Calculate deviation
Calculate deviation
$dist^t_1, var^t_{dist_1}$
$dist^t_{N-1}, var^t_{dist_{N-1}}$
$dev^t_1, var^t_{dev_1}$
$dev^t_{N-1}, var^t_{dev_{N-1}}$
$r^t_i = score^t_i \Big/ \sum_{i=1}^{N-1} score^t_i$, $\mathrm{score}^t_i = \frac{\eta \log\left(\frac{1}{dist^t_i + \varepsilon}\right) + (1-\eta)\log\left(\frac{1}{dev^t_i + \varepsilon}\right)}{\varphi\, var^t_{dist_i} + (1-\varphi)\, var^t_{dev_i}}$
$W^t_{\mathrm{global}} = \sum_{i=1}^{N} r^t_i W^t_i$

Fig. 8. The proposed multi-perspective distribution discrepancy aggregation strategy.

clients on average and aggregates them into a global model according to

$$W^t_{\mathrm{global}} = \sum_{i=1}^{N} \frac{1}{N} W^t_i, \tag{17}$$

where W^t_i denotes the local model of the ith client obtained from the tth epoch, and W^t_{global} denotes the global model obtained from the tth epoch.

The FedAvg strategy has the advantages of being simple in design and easy to use. However, it has an obvious drawback of assuming that all clients make the same contribution during federation training. In the FTL-based RMFD, the diagnosis performance of the local models from different source clients on the target domain may be significantly different due to the discrepancy in the distribution of data provided by different clients. To avoid negative transfer caused by model aggregation, it is essential to accurately evaluate the diagnosis performance of the local models from different source clients during the model aggregation process. Therefore, a multi-perspective distribution discrepancy aggregation strategy is proposed. As shown in Fig. 8, the proposed MPDDA strategy evaluates the diagnosis performance of the local models from different source clients on the target domain from three perspectives: statistical distance, domain adversarial, and stability, and jointly determines the aggregation weights according to these three evaluation metrics.

Statistical distance: The inter-domain distribution discrepancy is an important reason that affects the diagnosis performance of the local models from different source clients on the target domain. In general, the smaller the inter-domain distribution discrepancy, the better the diagnosis performance of the local models from different source clients on the target domain. The MK-MMD distance can reflect the inter-domain distribution discrepancy to some extent. Generally, the smaller the MK-MMD distance, the smaller the inter-domain distribution discrepancy. Therefore, the MK-MMD distance is adopted as an evaluation metric to determine the aggregation weights. The MK-MMD distance $dist_i$ of the high-level features between the ith source domain and the target domain can be calculated by

$$dist_i = \left\| E_{P(X^{S_i})}\left[\phi(G_f(X^{S_i}))\right] - E_{P(X^T)}\left[\phi(G_f(X^T))\right] \right\|^2_{\mathscr{H}_k}. \tag{18}$$

The smaller $dist_i$, the smaller the distribution discrepancy between the ith source domain and the target domain, which means that the aggregation weight assigned to the local model of the ith source client should be increased.

Domain adversarial: The adversarial training is introduced in [28] to reduce the inter-domain distribution discrepancy. The domain adversarial involves two components of RDFTL framework: the global domain discriminator and the feature extractor. The global domain discriminator is employed to discriminate whether a feature comes from the target or source domain. The feature extractor is used to extract features that are difficult to be discriminated by the global domain discriminator. Inspired by this idea, the domain adversarial is introduced into the model aggregation strategy and used as another evaluation metric to determine aggregation weights. Typically, the more difficult it is for the domain discriminator to discriminate whether features come from the target or source domain, the more similar the feature representations of the source and target domains are, that is, the better the performance of domain adaptation is. Therefore, the domain discriminator accuracy can be used to evaluate the performance of domain adaptation. The domain discrimination accuracy of the ith source client acc_i can be calculated by

$$acc_i = \frac{TP_i + TN_i}{TP_i + FP_i + TN_i + FN_i}, \tag{19}$$

where TP_i, TN_i, FP_i, and FN_i denote the number of true positives, true negatives, false positives, and false negatives, respectively. The domain discrimination deviation dev_i between acc_i and the ideal domain discrimination accuracy of 0.5 can be calculated by

$$dev_i = \left| acc_i - 0.5 \right|. \tag{20}$$

The smaller the domain discrimination deviation, the better the performance of domain adaptation, which implies that the aggregation weight should be increased.

Stability: The stability of the domain adaptation process can also reflect the diagnosis performance of the local models from different source clients on the target domain. The source domains with smaller distribution discrepancies from the target domain can usually align with the target domain faster and more stably during the domain adaptation process. Furthermore, since the $dist_i$ and dev_i obtained from each epoch have a certain fluctuation, the local models with poor performance may be assigned larger aggregation weights. Therefore, the variances of $dist_i$ and dev_i are introduced into the model aggregation strategy, which enables the diagnosis performance of the local models from the source clients to be evaluated from a periodic perspective, thus avoiding the negative transfer caused by a single fluctuation. After completing the tth epoch of training, the variance of $dist_i$ can be calculated by

$$var^t_{dist_i} = \frac{1}{\ell - 1} \sum_{k=t-\ell}^{t} \left[dist_i^k - \left(\sum_{j=t-\ell}^{t} dist_i^j / \ell \right) \right]^2, \tag{21}$$

and the variance of dev_i can be calculated by

$$var^t_{dev_i} = \frac{1}{\ell - 1} \sum_{k=t-\ell}^{t} \left[dev_i^k - \left(\sum_{j=t-\ell}^{t} dev_i^j / \ell \right) \right]^2, \tag{22}$$

where ℓ is the length of the sliding window used to calculate the variance.

The proposed MPDDA strategy is described in Algorithm 1. The model aggregation during the tth epoch of training is taken as an example, and the aggregation process is as follows, where $t \geq \ell$.

Algorithm 1 The proposed MPDDA strategy

```
Input: The source domain datasets $\{D^{S_i}\}_{i=1}^{N-1}$, the target domain
    dataset $D^T$, the maximum number of epochs $maxEpochs$, and
    the length of sliding window for calculating the variance $\ell$.
Output: A global model $W_{\text{global}}^{maxEpochs}$.
The target client initializes the global model $W_{\text{global}}^{0}$ and send
    it to all source clients;
for $t = 1$ to $maxEpochs$ do
    do in parallel
        Perform the source client operation($\{D^{S_i}\}_{i=1}^{N-1}$);
        Perform the target client operation($D^T$);
    end in parallel
end for
Perform the source client operation($\{D^{S_i}\}_{i=1}^{N-1}$):
    for source client $i$, $1 \leq i \leq N-1$, in parallel do
        Use the global model $W_{\text{global}}^{t-1}$ as the local model $W_i^t$;
        Train the local model $W_i^t$ using the dataset $D^{S_i}$ and
        calculate the domain discrimination accuracy $acc_i^t$ by
        Eq. (19);
        Extract the source domain features $feature_{S_i}^t$ by $W_i^t$;
        Transmit $acc_i^t$ and $feature_{S_i}^t$;
        if $t < \ell$ then
            Set the aggregation weight $r_i^t$ to $1/(N-1)$;
        else
            Calculate $dist_i^t$ and $dev_i^t$ by Eqs. (18) and (20),
            respectively;
            Calculate $var^t_{dist_i}$ and $var^t_{dev_i}$ by Eqs. (21) and (22),
            respectively;
            Calculate $score_i^t$ and $r_i^t$ by Eqs. (23) and (24),
            respectively;
        end if
        Perform the model aggregation to obtain the global model
        $W_{\text{global}}^t$ by Eq. (25);
    end for
Perform the target client operation($D^T$):
    Use the global model $W_{\text{global}}^{t-1}$ as the local model $W_N^t$;
    Extract the target domain features $feature_T^{t-1}$ by $W_N^t$;
    Transmit $feature_T^{t-1}$;
    Set the aggregation weight $r_N^t$ of the target client to zero;
    Perform the model aggregation to obtain the global model
    $W_{\text{global}}^t$ by Eq. (25);
```

Step 1: Use the ith source client to extract the source domain features $feature_{S_i}^t$, at the same time, the target client is used to extract the target domain features $feature_T^{t-1}$, where $i \in \{1, 2, \ldots, N-1\}$.

Step 2: Calculate the MK-MMD distance $dist_i^t$ between $feature_{S_i}^t$ and $feature_T^{t-1}$ by Eq. (18).

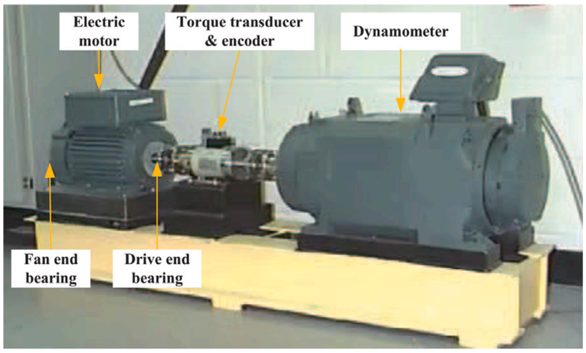


Fig. 9. CWRU bearing test rig [29].

Step 3: Calculate the domain discrimination accuracy acc_i^t and the domain discrimination deviation dev_i^t by Eqs. (19) and (20), respectively.

Step 4: Calculate the variances $var^t_{dist_i}$ and $var^t_{dev_i}$ by Eqs. (21) and (22), respectively.

Step 5: Calculate the score $score_i^t$ used to evaluate the diagnosis performance of the local model from the ith source client by

$$score_i^t = \frac{\eta \log\left(\frac{1}{dist_i^t + \varepsilon}\right) + (1-\eta)\log\left(\frac{1}{dev_i^t + \varepsilon}\right)}{\varphi var^t_{dist_i} + (1-\varphi) var^t_{dev_i}}, \tag{23}$$

where η and φ are the adjustable parameters and ε is a smoothing factor. η and φ are used to weight different evaluation metrics to ensure the optimal performance of the MPDDA strategy. η is used to balance the relative importance of MK-MMD distance and the domain discrimination deviation. φ is used to balance the relative importance of the variance of MK-MMD distance and that of the domain discrimination deviation. The aggregation weight r_i^t of the local model from the ith source client can be calculated by

$$r_i^t = score_i^t / \sum_{i=1}^{N-1} score_i^t. \tag{24}$$

Step 6: Weight and aggregate the local models of all clients to obtain the global model W_{global}^t by

$$W_{\text{global}}^t = \sum_{i=1}^{N} r_i^t W_i^t, \tag{25}$$

where the aggregation weight r_N^t of the target client is set to zero. W_{global}^t is taken as the local model of each client that will be used at the $(t+1)$th epoch.

4. Experiments

4.1. Experimental setup

The datasets used in the experiment include the Case Western Reserve University (CWRU) bearing dataset [29] and the Paderborn University (PU) bearing dataset [30].

The CWRU bearing test rig is shown in Fig. 9. The dataset used in this experiment consists of vibration signals collected at the drive end with a sampling frequency of 12 kHz. The vibration data under different health conditions listed in Table 1 are collected under four different working conditions, namely, four different rotating speeds including 1797, 1772, 1750, and 1730 rpm. The datasets under the four different working conditions are named A1, A2, A3, and A4, respectively. Each dataset contains 2000 samples, and each sample contains 1024 consecutive sample points, where a sample point is a vibration signal. There are 200 samples under each health condition. The ratio of the training set and test set is 8:2.

Table 1
Details of the CWRU bearing dataset.

Health condition	Fault diameter (in.)	Label
N	–	0
IF1	0.007	1
IF2	0.014	2
IF3	0.021	3
BF1	0.007	4
BF2	0.014	5
BF3	0.021	6
OF1	0.007	7
OF2	0.014	8
OF3	0.021	9

Table 2
Details of the PU bearing dataset.

Health condition	Damage mode	Damage degree	Bearing code	Label
N	–	–	K004	0
IF1	Pitting	1	KI21	1
IF2	Pitting	2	KI18	2
IF3	Pitting	3	KI16	3
OF1	Pitting	1	KA04	4
OF2	Indentations	1	KA15	5
OF3	Pitting	2	KA16	6
IF + OF1	Indentations	1	KB27	7
IF + OF2	Pitting	2	KB23	8
IF + OF3	Pitting	3	KB24	9

Table 3
Network structure of the model.

Module	Layer type	Kernel/Channels/Stride	Output
Feature extractor	Convolution	$7 \times 1/64/2$	64, 512
	Max-pooling	$3 \times 1/64/2$	64, 256
	Residual-block1	$\begin{bmatrix} 3 \times 1/64/1 \\ 3 \times 1/64/1 \end{bmatrix} \times 2$	64, 256
	Residual-block2	$\begin{bmatrix} 3 \times 1/128/2 \\ 3 \times 1/128/1 \end{bmatrix} \times 2$	128, 128
	Residual-block3	$\begin{bmatrix} 3 \times 1/256/2 \\ 3 \times 1/256/1 \end{bmatrix} \times 2$	256, 64
	Residual-block4	$\begin{bmatrix} 3 \times 1/512/2 \\ 3 \times 1/512/1 \end{bmatrix} \times 2$	512, 32
	Avg-pooling	$32 \times 1/512/32$	512, 1
	Flatten	–	512
Classifier	FC1	–	256
	FC2	–	10
Global domain discriminator	FC1	–	256
	FC2	–	2

Table 4
Setting of hyper-parameters used in model training.

Parameter name	Description	Value
momentum	Momentum used in SGD	0.9
batchSize	Number of samples used in each iteration	64
maxEpochs	Maximum number of epochs	100
γ	Learning rate	0.001
α	Weight of global domain alignment loss	0.65
β	Weight of sub-domain alignment loss	0.35
ℓ	Length of the sliding window for calculating variances	10
η	Weight of the statistical distance	0.7
φ	Weight of the variance of the statistical distance	0.3
ε	Smoothing factor of MPDDA strategy	0.001

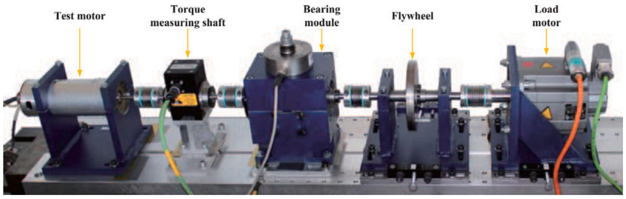


Fig. 10. PU bearing test rig [30].

The PU bearing test rig is shown in Fig. 10. The data used in this experiment include the healthy and real damaged bearing data collected under four different working conditions with a sampling frequency of 64 kHz. According to different radial forces, load torques, and rotating speeds, the four different working conditions are as follows: 1000 N/0.7 N m/1500 rpm, 1000 N/0.7 N m/900 rpm, 1000 N/0.1 N m/1500 rpm, and 400 N/0.7 N m/1500 rpm. The vibration data under different health conditions listed in Table 2 are collected under the four different working conditions. The datasets under the four different working conditions are named B1, B2, B3, and B4, respectively. Each dataset contains 4000 samples, and each sample contains 2048 consecutive sample points. There are 400 samples under each health condition. The ratio of the training set and test set is 8:2.

In the RMFD based on RDFTL, the network structure of each client model is the same, as shown in Table 3. The model uses a deep residual network as the feature extractor, which can better extract inter-domain invariant features from the original signals. The ELU activation function is used after each convolutional layer and each fully connected layer in the model. The stochastic gradient descent (SGD) algorithm is used in model training. The setting of hyper-parameters used in model training is shown in Table 4. Note that the parameters α, β, η, and φ adopted in this paper are chosen using the grid-search method. The influences of changes in the values of α, β, η, and φ on the diagnosis performance are observed through a series of experiments, and the optimal combination pairs are selected as the experimental parameters.

The experimental platform includes three source clients and one target client. The hardware configurations of each client mainly include 64 GB of RAM, an Intel i7-9700K CPU with eight cores @ 3.6 GHz, 8 GB of GPU memory, and an NVIDIA RTX 2070 SUPER GPU with 2560 CUDA cores. The software configurations of each client mainly include Horovod 0.27.0, PyTorch 1.12.1, and CentOS 8.1.

4.2. Validation of the effectiveness of the proposed RDFTL framework and asynchronous domain adaptation strategy

The computational and communication overheads are the two main factors that affect the efficiency of model training in FTL. To verify the effectiveness of the proposed RDFTL framework and asynchronous domain adaptation strategy in improving training efficiency, the three methods of CSFTL, CSFTL-ASYNC, and RDFTL are compared in this experiment. The CSFTL method employs the client–server architecture without incorporating the proposed asynchronous domain adaptation strategy. The CSFTL-ASYNC method adopts the client–server architecture that incorporates the proposed asynchronous domain adaptation strategy. The RDFTL method uses the proposed decentralized architecture based on Ring-AllReduce and the proposed asynchronous domain adaptation strategy. Four cross-working condition RMFD tasks are designed on the CWRU and PU bearing datasets respectively, as shown in Table 5. Note that the data of the three source clients are labeled, whereas the data of the target client are set to be unlabeled.

Fig. 11 presents the average computation time and communication time of the three different FTL methods on the eight cross-working condition RMFD tasks listed in Table 5. The computation time includes the local training time of the source clients and the feature extraction time of the target client, and the communication time includes the feature transmission time and the model transmission time. As shown in Fig. 11, RDFTL has the least total time. The total time of RDFTL is reduced by 44.09% and 11.60% than that of CSFTL and CSFTL-ASYNC, respectively. The computation time and communication time of RDFTL are reduced by 30.92% and 58.77% than those of CSFTL, respectively.

Table 5
Description of the cross-working condition RMFD tasks.

Task name	Source client #1	Source client #2	Source client #3	Target client
T1	A2	A3	A4	A1
T2	A1	A3	A4	A2
T3	A1	A2	A4	A3
T4	A1	A2	A3	A4
T5	B2	B3	B4	B1
T6	B1	B3	B4	B2
T7	B1	B2	B4	B3
T8	B1	B2	B3	B4

Table 6
Comparison of the local training time, feature extraction time, and total computation time of different FTL methods used for RMFD.

FTL method	Local training time (s)	Feature extraction time (s)	Total computation time (s)
CSFTL	327.54	148.94	476.48
CSFTL-ASYNC	332.77	53.21	332.77
RDFTL	329.15	50.47	329.15

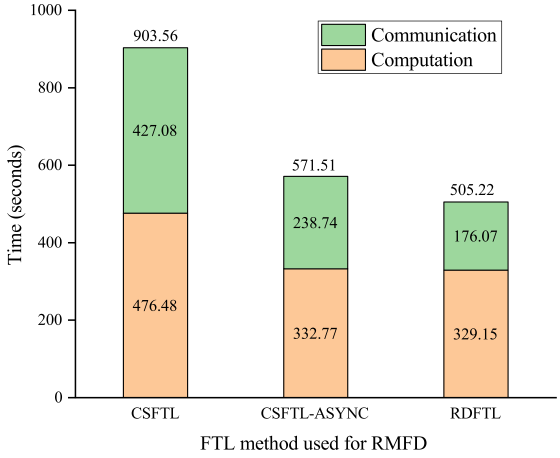


Fig. 11. Comparisons of the average computation time and communication time of different FTL methods used for RMFD.

The computation time of RDFTL is basically the same as that of CSFTL-ASYNC, but the communication time of RDFTL is reduced by 26.25% than that of CSFTL-ASYNC.

Fig. 12 illustrates the feature transmission time and model transmission time of the three different FTL methods. As shown in Fig. 12, the model transmission time of RDFTL is reduced by 80.41% and 70.55% than that of CSFTL and CSFTL-ASYNC, respectively. This is because RDFTL adopts the proposed decentralized architecture based on Ring-AllReduce, which can greatly reduce the model transmission time. The model transmission time of CSFTL-ASYNC is reduced by 33.48% than that of CSFTL, which is because there is no need to transmit the local models from the server to the target client when adopting the asynchronous domain adaptation strategy in the client–server architecture. Compared with CSFTL, the feature transmission time of RDFTL and that of CSFTL-ASYNC are reduced by 49.46% and 48.67%, respectively. This is because only one global model is needed instead of multiple local models to extract the target domain features when the asynchronous domain adaptation strategy is adopted in RDFTL and CSFTL-ASYNC, which helps to reduce the additional feature transmission time.

Table 6 gives the comparison of the local training time, feature extraction time, and total computation time of different FTL methods used for RMFD. The total computation time of CSFTL is the sum of the maximum local training time among all source clients and the feature extraction time of the target client, and the total computation time of CSFTL-ASYNC and that of RDFTL depend on the larger of the maximum local training time and the feature extraction time. As seen in Table 6, the differences in the local training time of the three methods are slight, which is because the local training strategies adopted by the three methods are basically the same. The feature extraction time of CSFTL accounts for 31.26% of the total computation time, whereas the feature extraction time of RDFTL and that of CSFTL-ASYNC account for 0% of the total computation time, which means that the feature extraction time of RDFTL and that of CSFTL-ASYNC are completely hidden. This is because there is idle waiting between the target client and the source clients in CSFTL, whereas RDFTL and CSFTL-ASYNC adopt the proposed asynchronous domain adaptation strategy, which can enable the local training of the source clients and the feature extraction of the target client to be executed in parallel.

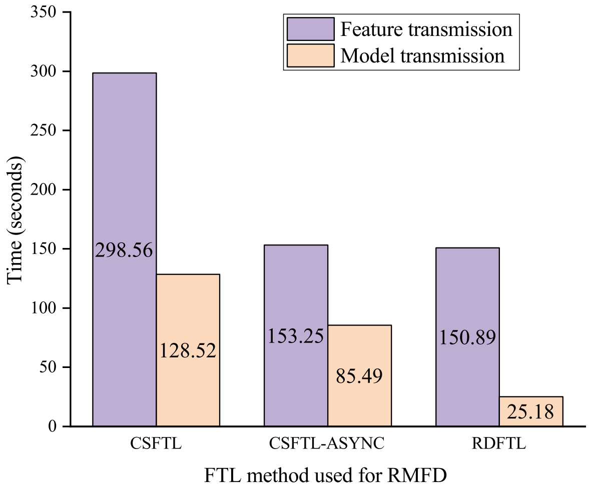


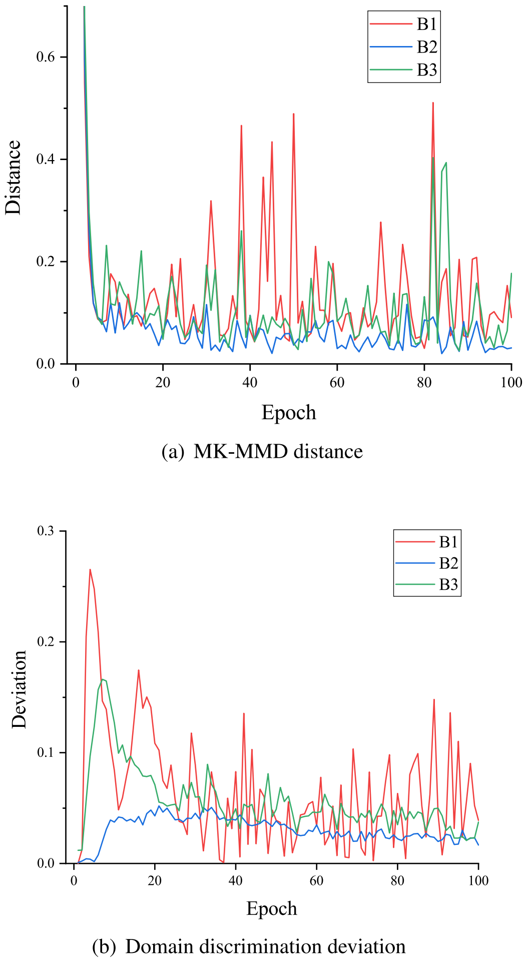
Fig. 12. Comparisons of the feature transmission time and model transmission time of different FTL methods used for RMFD.

From the above analysis, the conclusions are as follows. Firstly, the proposed RDFTL framework can significantly reduce the communication overhead of FTL. Secondly, the proposed asynchronous domain adaptation strategy can avoid idle waiting and reduce the additional communication overhead, thereby improving the overall training efficiency of FTL.

4.3. Validation of the effectiveness of the proposed MPDDA strategy

To validate the effectiveness of the three evaluation metrics in the proposed MPDDA strategy, some experiments are conducted on the task T8 listed in Table 5. The three PU bearing datasets B1, B2, and B3 are used as the three source domains and B4 is used as the target domain. The local models trained on B1, B2, and B3 achieve diagnosis accuracies of 65.78%, 96.73%, and 87.45% on the target domain B4, respectively. In terms of the diagnosis accuracies achieved by the local models of the three source clients, B2 is the best, followed by B3, and B1 is the worst.

Fig. 13(a) shows the variation curves of the MK-MMD distances between different source domains and the target domain during the training process. The smaller the MK-MMD distance, the smaller the inter-domain distribution discrepancy. Fig. 13(b) presents the variation curves of the domain discrimination deviations of different source clients during the training process. The smaller the domain discrimination deviation, the smaller the inter-domain distribution discrepancy. From the overall performance of the local models from the three source clients on both the MK-MMD distance and domain discrimination deviation, B2 is the best, followed by B3, and B1 is the worst, which indicates that both the MK-MMD distance and domain discrimination deviation are effective evaluation metrics. As shown in Figs. 13(a) and 13(b), from the overall fluctuation amplitude of the three MK-MMD distance curves and the three domain discrimination deviation curves,



(a) MK-MMD distance

(b) Domain discrimination deviation

Fig. 13. Variation curves of different evaluation metrics under the task T8.

B2 is the smallest, followed by B3, and B1 is the largest, which indicates that the stability is also an effective evaluation metric.

Fig. 14 presents the aggregation weights of different source clients obtained by MPDDA strategy under the task T8. As shown in Fig. 14, in terms of the aggregation weights of the three source clients, B2 is the largest, followed by B3, and B1 is the smallest. The aggregation weights of the three source clients are consistent with their diagnosis performance on the target domain, and the aggregation weights of the high-quality local models are significantly different from those of the low-quality local models. This can fully utilize the role of high-quality local models in aggregation and reduce the negative transfer caused by low-quality local models, thereby effectively improving the diagnosis performance of the global model.

To verify the effectiveness of the proposed MPDDA strategy in improving the diagnosis performance of the global model, the comparative experiments are carried out using four different model aggregation strategies on the eight tasks listed in Table 5. These strategies are FedAvg [27], Fed-DA, Fed-MK-MMD, and the proposed MPDDA. FedAvg weights the local models participating in the aggregation on average. Fed-DA uses the domain discrimination deviation and the corresponding variance to calculate the aggregation weights. Fed-MK-MMD uses the MK-MMD distance and the corresponding variance to calculate the aggregation weights.

Table 7 shows the diagnosis accuracies achieved with different aggregation strategies under the cross-working condition RMFD tasks. As shown in Table 7, under the four transfer tasks of CWRU, the average diagnosis accuracies obtained by FedAvg, Fed-DA, Fed-MK-MMD, and MPDDA are 97.67%, 99.12%, 99.00%, and 99.81%, respectively. Under the four transfer tasks of PU, the average diagnosis accuracies obtained by FedAvg, Fed-DA, Fed-MK-MMD, and MPDDA are 82.34%, 92.33%, 91.48%, and 98.87%, respectively. The results indicate that these four aggregation strategies have superior diagnosis performance on the CWRU bearing dataset, whereas MPDDA significantly outperforms the other three aggregation strategies on the PU bearing dataset. The diagnosis performance of FedAvg, Fed-DA, and Fed-MK-MMD on the CWRU bearing dataset is better than that on the PU bearing dataset, which is because the distribution discrepancy between different domains in the CWRU bearing dataset is smaller than that in the PU bearing dataset. Therefore, the local model from each source client performs well on the target domain for the CWRU bearing dataset, which makes the impact of aggregation weights on the diagnosis performance of the global model insignificant. However, when facing the more complex PU bearing dataset, the diagnosis performance of the local models from different source clients differs significantly. The accurate evaluation of the aggregation weights of different source clients becomes essential to fully ensure the diagnosis performance of the global model. The inappropriate aggregation weights can lead to serious negative transfer.

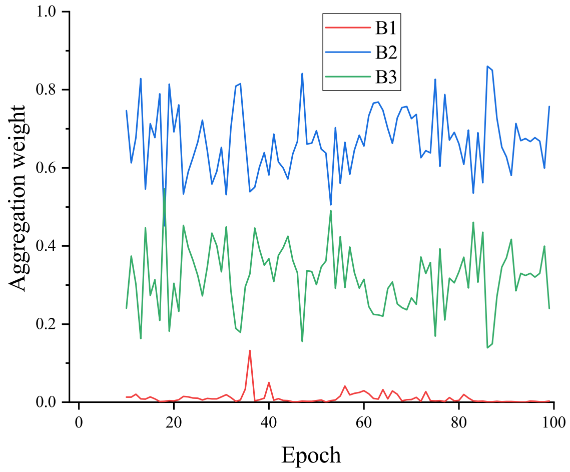


Fig. 14. Aggregation weights of the local models from different source clients obtained by MPDDA strategy under the task T8.

Table 7
Diagnosis accuracies achieved with different aggregation strategies under the cross-working condition RMFD tasks.

Task	FedAvg	Fed-DA	Fed-MK-MMD	MPDDA
T1	93.75 ± 0.45	96.75 ± 0.55	96.50 ± 0.25	99.25 ± 0.12
T2	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
T3	97.25 ± 0.31	100.00 ± 0.00	99.50 ± 0.09	100.00 ± 0.00
T4	99.68 ± 0.11	99.75 ± 0.06	100.00 ± 0.00	100.00 ± 0.00
Avg.	97.67	99.12	99.00	99.81
T5	81.09 ± 0.85	91.25 ± 1.25	93.20 ± 1.47	98.82 ± 0.38
T6	84.50 ± 1.15	93.67 ± 1.51	89.21 ± 1.57	99.21 ± 0.15
T7	82.75 ± 0.95	92.02 ± 1.17	90.77 ± 1.39	98.39 ± 0.22
T8	81.05 ± 1.12	92.39 ± 1.26	92.73 ± 1.75	99.07 ± 0.19
Avg.	82.34	92.33	91.48	98.87
Total avg.	90.01	95.72	95.23	99.34

The tasks T4 and T8 are taken as examples, the variation of the diagnosis accuracies achieved with the four different aggregation strategies during the training process is further analyzed. Figs. 15 and 16 present the diagnosis accuracies achieved with different aggregation strategies under the tasks T4 and T8, respectively. As shown in Fig. 15, the final diagnosis accuracies achieved by MPDDA and Fed-MK-MMD reach 100% under the task T4. MPDDA has the fastest convergence speed and almost no oscillation. The diagnosis accuracy of Fed-DA is slightly lower than that of MPDDA, and its convergence speed is also slightly slower than that of MPDDA. FedAvg has the lowest diagnosis accuracy,

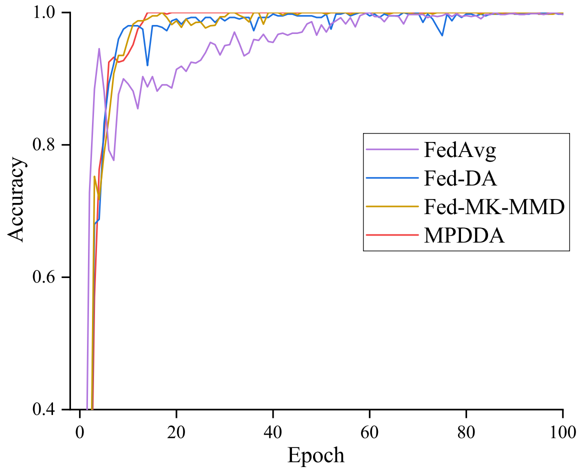


Fig. 15. Diagnosis accuracies achieved with different aggregation strategies under the task T4.

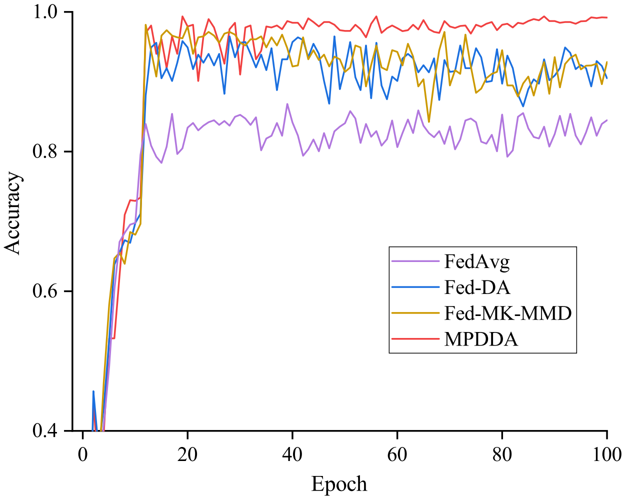


Fig. 16. Diagnosis accuracies achieved with different aggregation strategies under the task T8.

the slowest convergence speed, and obvious oscillation. As shown in Fig. 16, the final diagnosis accuracy obtained by MPDDA is the highest and its oscillation is slight under the task T8. The diagnosis accuracies of Fed-DA and Fed-MK-MMD are significantly lower than that of MPDDA, and the oscillations of the diagnosis accuracies of Fed-DA and Fed-MK-MMD are larger than that of MPDDA. FedAvg has the lowest diagnosis accuracy. As can be seen from Figs. 15 and 16, when facing the complex cross-working condition RMFD tasks, MPDDA can significantly alleviate the negative transfer caused by model aggregation, thereby improving the diagnosis performance of the global model.

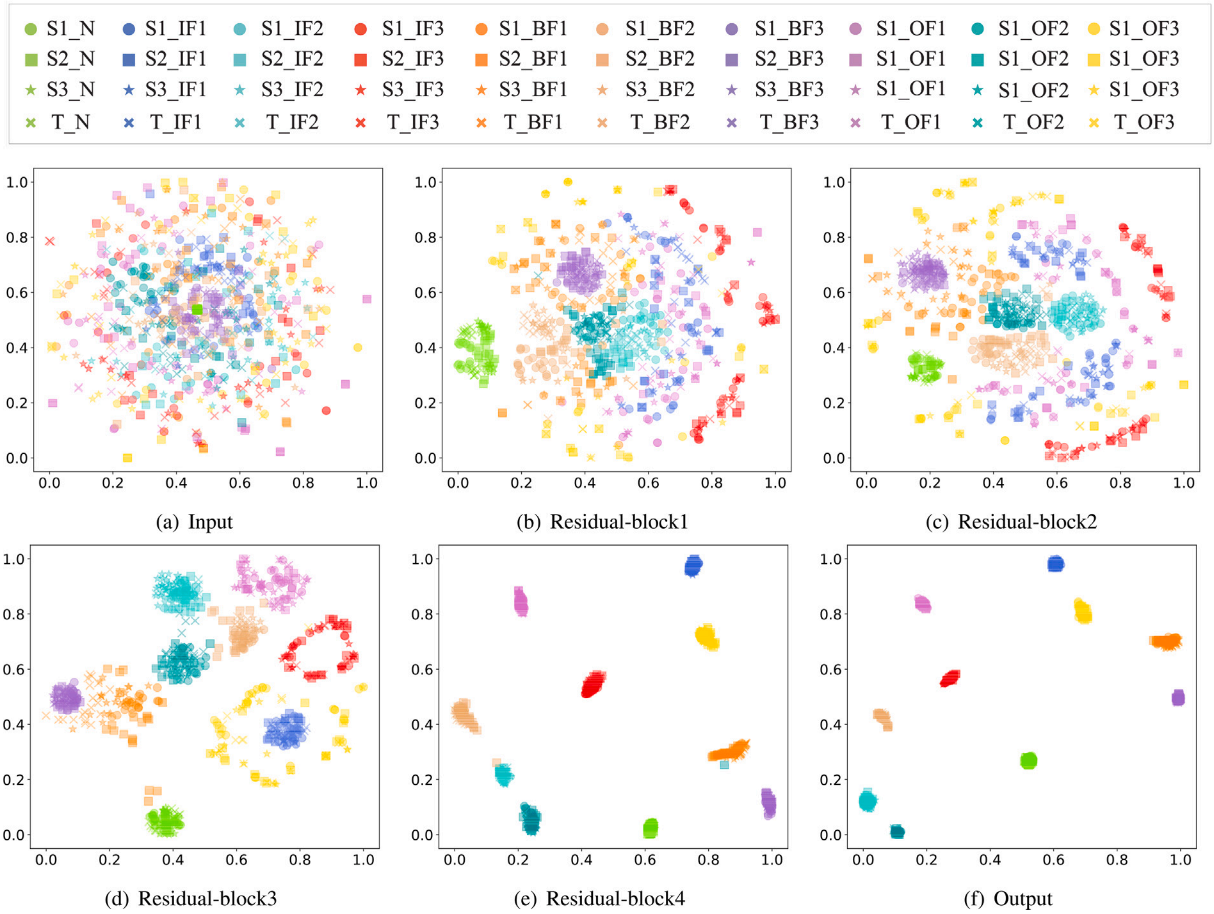
4.4. Comparison with other fault diagnosis methods

To further validate the effectiveness of the proposed RDFTL method, RDFTL is compared with six other different fault diagnosis methods, including FTLDAN [19] (2022), DWFA [23] (2022), MSSA [10] (2022), FMDAAN-V [21] (2023), LQKF [25] (2023), and AFTBL [22] (2023), where FTLDAN, DWFA, FMDAAN-V, LQKF, and AFTBL are the advanced federated transfer learning methods for RMFD, and MSSA is the advanced transfer learning approach for RMFD. FTLDAN [19] uses deep adversarial networks to reduce inter-domain distribution discrepancy, and adopts a source domain multi-classifier consistency scheme to improve the prediction accuracy. DWFA [23] adopts a domain adaptation strategy based on MMD distance to reduce inter-domain distribution discrepancy. In addition, DWFA uses the MMD distance to measure distribution discrepancy and adopts a weighted federated averaging method based on distribution discrepancy for model aggregation. FMDAAN-V [21] jointly utilizes a global feature alignment module based on MK-MMD distance and a global domain discriminator to reduce the distribution discrepancies between the target and source domains, and employs the FedAvg strategy for model aggregation. LQKF [25] adopts a low-quality knowledge filtering strategy to generate high-confidence pseudo labels for the target domain, and utilizes the filtering idea to measure the contribution of each local model, so as to dynamically aggregate the local models. AFTBL [22] employs a FTL approach based on broad learning and attention mechanism for RMFD, and utilizes the FedAvg strategy for model aggregation. MSSA [10] is a multi-source sub-domain adaptation transfer learning method, which uses the LMMD distance to calculate the inter-domain distribution discrepancy and utilizes a multi-branch network structure to reduce the distribution discrepancies between the source and target domains. Note that MSSA does not need to consider data privacy, which means that the data from different domains could be used for centralized training.

4.4.1. RMFD under cross-working condition scenarios

The comparative experiments are carried out using FTLDAN, DWFA, FMDAAN-V, LQKF, AFTBL, MSSA, and RDFTL on the eight cross-working condition RMFD tasks listed in Table 5. Table 8 provides the accuracies achieved with different fault diagnosis methods under the cross-working condition scenarios. As shown in Table 8, RDFTL achieves the best average diagnosis accuracy among all the FTL methods, which shows that RDFTL is superior to the other FTL methods. The differences in the diagnosis accuracies achieved on the four transfer tasks of CWRU using these seven methods are slight. The main reasons are as follows. On the one hand, due to the relatively small distribution discrepancies among the four domains of CWRU, it is not difficult to train the local models that exhibit good diagnosis performance on the target domain. On the other hand, since each local model is comparably excellent, the performance of the global model is not particularly sensitive to the aggregation weights, and the issue of negative transfer brought by model aggregation is mitigated.

However, when facing the four relatively complex transfer tasks of PU, the average diagnosis accuracies of these seven different methods have declined, whereas RDFTL still maintains the highest average diagnosis accuracy of 98.87% among all the FTL methods. The main reasons are as follows. FTLDAN, DWFA, and FMDAAN-V only consider marginal distribution alignment in the domain adaptation, but this does not mean that the conditional distribution discrepancy can also be reduced implicitly, which restricts their performance on the complex transfer tasks to some extent. In contrast, RDFTL reduces both marginal distribution discrepancy and conditional distribution discrepancy through the global domain alignment and the sub-domain alignment. In addition, FTLDAN, FMDAAN-V, and AFTBL use the FedAvg strategy in model aggregation. Once the low-quality local models appear in the aggregation process, the FedAvg strategy will hurt the performance of the global model. DWFA only relies on the MMD distance to evaluate the aggregation weights of local models, which is prone to cause negative transfer in model aggregation for complex FTL tasks. LQKF uses the batch normalized MMD distance as a component of the loss, without directly utilizing the output of the feature extractor, which ensures data privacy to a certain extent but at the cost of some diagnosis performance. The negative transfer caused by low-quality local models can be effectively avoided in RDFTL that incorporates MPDDA strategy. Compared with MSSA without considering data privacy, the average diagnosis accuracy achieved by RDFTL on all tasks is only 0.04% lower. This proves that the proposed RDFTL that considers data privacy still has strong competitiveness in diagnosis performance.



(a) Input

(b) Residual-block1

(c) Residual-block2

(d) Residual-block3

(e) Residual-block4

(f) Output

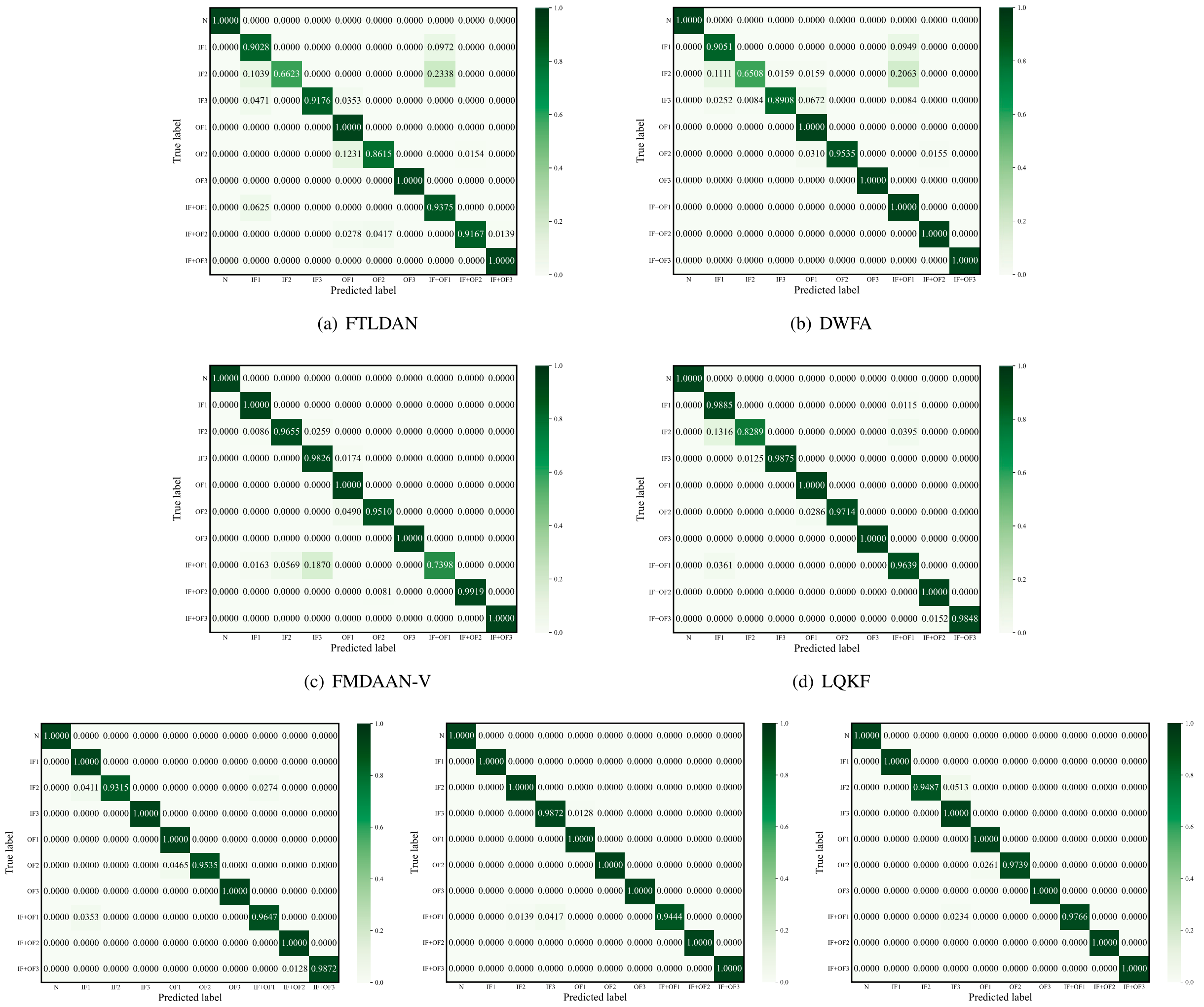
Fig. 17. Feature visualization of the outputs of different layers of the final global model trained with RDFTL under the task T4.

Table 8
Accuracies achieved with different fault diagnosis methods under cross-working condition scenarios.

Task	FTLDAN [19] (2022)	DWFA [23] (2022)	FMDAAN-V [21] (2023)	LQKF [25] (2023)	AFTBL [22] (2023)	MSSA [10] (2022)	RDFTL
T1	97.52 ± 0.11	98.85 ± 0.21	99.12 ± 0.06	99.10 ± 0.07	99.79 ± 0.05	99.57 ± 0.09	99.25 ± 0.12
T2	97.15 ± 0.15	100.00 ± 0.00	99.57 ± 0.13	99.26 ± 0.05	99.76 ± 0.06	100.00 ± 0.00	100.00 ± 0.00
T3	98.06 ± 0.07	99.58 ± 0.04	100.00 ± 0.00	98.89 ± 0.08	100 ± 0.00	99.81 ± 0.05	100.00 ± 0.00
T4	96.57 ± 0.23	97.34 ± 0.32	100.00 ± 0.00	98.90 ± 0.12	99.49 ± 0.05	100.00 ± 0.00	100.00 ± 0.00
Avg.	97.33	98.94	99.67	99.04	99.76	99.85	99.81
T5	88.14 ± 0.47	91.09 ± 0.45	93.99 ± 0.55	96.16 ± 0.18	98.15 ± 0.12	98.96 ± 0.16	98.82 ± 0.38
T6	90.32 ± 0.42	93.20 ± 0.55	96.77 ± 0.56	98.05 ± 0.14	97.37 ± 0.18	98.23 ± 0.33	99.21 ± 0.15
T7	93.80 ± 0.35	94.21 ± 0.28	94.73 ± 0.76	96.60 ± 0.21	97.89 ± 0.21	99.11 ± 0.17	98.39 ± 0.22
T8	92.51 ± 0.63	93.77 ± 0.87	96.11 ± 0.79	97.25 ± 0.12	98.26 ± 0.13	99.34 ± 0.12	99.07 ± 0.19
Avg.	91.19	93.07	95.40	97.02	97.92	98.91	98.87
Total avg.	94.26	96.01	97.54	98.03	98.84	99.38	99.34

As shown in Fig. 17, the output features of different layers of the global model trained with RDFTL under the task T4 are visualized by t-SNE. Fig. 17(a) shows that there are significant distribution discrepancies between the source domains and the target domain without performing domain alignment. Figs. 17(b)–17(e) show that the feature distributions of the same fault category in different domains are gradually aligned. As can be seen from Fig. 17(f), after passing the classifier, the feature distributions of the same fault category in all domains are well aligned, and the differences between the feature distributions of the different fault categories are very obvious, which indicates that the features extracted by the global model trained with RDFTL achieve good clustering and separability under all fault categories.

Fig. 18 illustrates the confusion matrices of different fault diagnosis methods under the task T8. As shown in Fig. 18(a), FTLDAN misclassifies 9.72% of IF1 as IF+OF1 and misclassifies 23.38% IF2 as IF+OF1. As shown in Fig. 18(b), DWFA misclassifies 9.49% of IF1 as IF+OF1 and misclassifies 20.63% of IF2 as IF+OF1. As shown in Fig. 18(c), FMDAAN-V misclassifies 18.70% of IF+OF1 as IF3 and misclassifies 4.90% of OF2 as OF1. As shown in Fig. 18(d), LQKF misclassifies 13.16% of IF2 as IF1 and misclassifies 3.95% of IF2 as IF+OF1. As shown in Figs. 18(e), 18(f), and 18(g), AFTBL, MSSA, and RDFTL have slight misclassifications, e.g., AFTBL misclassifies 4.65% of OF2 as OF1, MSSA misclassifies 4.17% of IF+OF1 as IF3, and RDFTL misclassifies 2.61% of OF2 as OF1. This is because IF1, IF2, IF3, and IF+OF1 contain the fault features of the inner-race, which increases the difficulty of distinguishing between IF1, IF2, IF3 and IF+OF1. Both OF1 and OF2 are the outer-race faults and have the same damage degree, and the difference between them is only that they have different damage modes, thus it is difficult to distinguish between OF1 and OF2. The results indicate that the proposed RDFTL method can still accurately distinguish the fault categories with slight differences under the protection of data privacy.



(a) FTLDAN

(b) DWFA

(c) FMDAAN-V

(d) LQKF

(e) AFTBL

(f) MSSA

(g) RDFTL

Fig. 18. Confusion matrices of different fault diagnosis methods under the task T8.

Table 9
Description of the dataset used under cross-device scenarios.

Health condition	Number of samples		Label
	CWRU	PU	
N	1200	1200	0
IF	1200	1200	1
OF	1200	1200	2

Table 10
Description of the cross-device RMFD tasks.

Task name	Source client #1	Source client #2	Source client #3	Target client
T9	A1	A2	A3	B1
T10	A1	A2	A3	B2
T11	A1	A2	A3	B3
T12	B1	B2	B3	A1
T13	B1	B2	B3	A2
T14	B1	B2	B3	A3

4.4.2. RMFD under cross-device scenarios

To verify the effectiveness of the proposed RDFTL method for RMFD under cross-device scenarios, the normal condition samples, inner-race fault samples, and outer-race fault samples of the CWRU and PU bearing datasets under different working conditions are selected to carry out the experiment, as shown in Table 9. Six cross-device RMFD tasks are designed, as shown in Table 10. Under the tasks T9, T10, and T11, the three source domains come from CWRU, and the target domain comes from PU. Under the tasks T12, T13, and T14, the three source domains come from PU and the target domain comes from CWRU.

Table 11 presents the accuracies achieved with different fault diagnosis methods under cross-device scenarios. As shown in Table 11, the average diagnosis accuracy achieved by RDFTL under the three cross-device RMFD tasks from CWRU to PU is 82.10%, and the average diagnosis accuracy achieved by RDFTL under the three cross-device RMFD tasks from PU to CWRU is 94.41%, which indicates that the proposed RDFTL method also has excellent diagnosis performance in the cross-device RMFD. In this experiment, the real damage data of PU are selected, while only artificial damage data are provided by CWRU, which means that PU as the source domain can provide richer fault information than CWRU as the source domain. Therefore, the better diagnosis performance is achieved when using PU as the source domain.

As can be seen from Table 11, under these six cross-device RMFD tasks, the average diagnosis accuracy of RDFTL reaches 88.26%, which is higher than that of the other FTL methods. Under the cross-device scenarios, the distribution discrepancies between the data provided by different clients are usually significant, which greatly increases the difficulty of domain adaptation and is more likely to cause negative transfer in model aggregation. Since RDFTL considers both marginal distribution alignment and conditional distribution alignment, which

Table 11
Accuracies achieved with different fault diagnosis methods under cross-device scenarios.

Task	FTLDAN [19] (2022)	DWFA [23] (2022)	FMDAAN-V [21] (2023)	LQKF [25] (2023)	AFTBL [22] (2023)	MSSA [10] (2022)	RDFTL
T9	57.28 ± 2.12	68.15 ± 2.32	75.25 ± 1.84	79.87 ± 0.88	76.30 ± 1.17	79.23 ± 0.66	77.14 ± 0.55
T10	62.54 ± 1.89	71.00 ± 1.76	79.56 ± 0.82	83.36 ± 0.77	82.56 ± 0.71	89.34 ± 0.41	86.95 ± 0.21
T11	61.12 ± 1.56	75.30 ± 1.21	75.63 ± 1.13	79.20 ± 1.23	80.54 ± 0.42	86.24 ± 0.62	82.22 ± 0.32
Avg.	60.31	71.48	76.81	80.81	79.80	84.94	82.10
T12	72.84 ± 1.08	79.52 ± 0.67	86.85 ± 0.33	91.76 ± 0.65	92.27 ± 0.58	96.25 ± 0.08	93.91 ± 0.11
T13	81.60 ± 0.89	86.25 ± 0.52	91.23 ± 0.25	91.28 ± 0.31	89.46 ± 1.14	91.75 ± 0.15	93.75 ± 0.23
T14	70.69 ± 1.41	78.82 ± 0.32	85.34 ± 0.56	92.10 ± 0.47	92.14 ± 0.72	93.20 ± 0.21	95.58 ± 0.19
Avg.	75.04	81.53	87.81	91.71	91.29	93.73	94.41
Total avg.	67.68	76.51	82.31	86.26	85.55	89.34	88.26

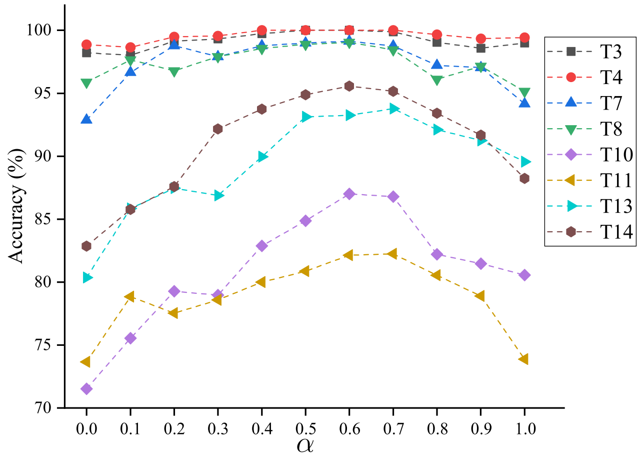


Fig. 19. Impact of changes in the value of parameter α on the diagnosis accuracy.

can greatly reduce the inter-domain distribution discrepancies, thereby effectively improving the diagnosis performance of the local models. The aggregation weights have a more significant impact on the diagnosis performance of the global model in the cross-device RMFD. RDFTL can still reasonably evaluate the aggregation weights under the cross-device scenarios, which can effectively reduce negative transfer, thereby improving the diagnosis performance of the global model. Under cross-working condition scenarios, AFTBL outperforms LQKF. However, LQKF exhibits better average diagnosis accuracy than AFTBL under cross-device scenarios, as shown in Table 11. This may be because there is a greater likelihood of encountering source domains with significant distribution discrepancies from the target domain under cross-device scenarios, leading to a higher incidence of low-quality local models. The FedAvg strategy adopted by AFTBL cannot filter out low-quality local models in model aggregation, and may even cause serious negative transfer. This once again demonstrates the importance of the aggregation strategy in FTL. Compared with MSSA without considering data privacy, the average diagnosis accuracy achieved by RDFTL under the six cross-device RMFD tasks is only 1.07% lower, and RDFTL outperforms MSSA under the tasks T13 and T14. This is because the source domains with low-quality data are more likely to appear in the cross-device RMFD scenarios. In this case, directly using all domains for centralized training may reduce the diagnosis accuracy. A lower aggregation weight is assigned to a low-quality local model in the model aggregation process of RDFTL, which can effectively alleviate the negative impact of low-quality data, thereby obtaining better diagnosis performance.

4.5. Parameter sensitivity analysis

To evaluate the contributions of the global domain alignment and the sub-domain alignment to domain adaptation, a series of experiments are conducted on the eight RMFD tasks through adjusting the values of parameters α and β in Eq. (9). The diagnosis accuracies are recorded as the values of α and β vary from 0 to 1, where α represents the weight of the global domain alignment, β indicates the weight of the sub-domain alignment, and $\beta = 1 - \alpha$. Fig. 19 shows the impact of changes in the value of parameter α on the diagnosis accuracy. For the four cross-working RMFD tasks of T3, T4, T7, and T8, the range of α corresponding to the optimal diagnosis accuracy of each task is in [0.6, 0.7]. The diagnosis accuracies outside this range have declined, but it is not significant. For the four cross-working RMFD tasks of T10, T11, T13, and T14, the range of α corresponding to the optimal diagnosis accuracy of each task is also in [0.6, 0.7]. However, different from the cross-working condition scenarios, the diagnosis accuracies obtained under the cross-device scenarios are sensitive to the change of parameters. This is because the feature distributions under the cross-device scenarios are more complex and diverse than that under the cross-working condition scenarios, which leads to more dependence on appropriate parameter settings to obtain better diagnosis performance under the cross-device scenarios. Overall, when α is set to approximately 0.65, the diagnosis accuracies gained under different RMFD tasks are relatively excellent and stable.

The effect of the proposed MPDDA strategy is affected by the adjustable parameters η and φ. To evaluate the contributions of the MK-MMD distance, domain discrimination deviation, and their corresponding variances to MPDDA strategy, the diagnosis accuracies are recorded as the values of η and φ vary from 0 to 1, as shown in Fig. 20. η and $1 - \eta$ represent the weight of MK-MMD distance and that of the domain discrimination deviation, respectively. φ and $1 - \varphi$ represent the weight of the variance of MK-MMD distance and that of the variance of the domain discrimination deviation, respectively. The grid-search method is adopted for finding the optimal combination of η and φ from $\eta = \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$ and $\varphi = \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$. Fig. 20 shows the impact of different combinations of parameters η and φ on the diagnosis accuracy. As seen in Fig. 20, the diagnosis accuracies are not ideal and fluctuate greatly when the values of η and φ are quite extreme. The diagnosis accuracies related to φ significantly improve when the value of η is in [0.6, 0.8]. Similarly, the diagnostic accuracies related to η also significantly improve when the value of φ is in [0.2, 0.4]. Overall, when η and φ are set to approximately 0.7 and 0.3 respectively, the accuracy reaches its optimal value.

5. Conclusions

In this paper, an RDFTL method for intelligent fault diagnosis is proposed, which can obtain a cross-domain fault diagnosis model with excellent performance in RMFD with data privacy at a fast training speed. Firstly, unlike the traditional FTL methods based on client–server architecture, a ring-based decentralized federated transfer learning framework is adopted in the proposed method, which effectively reduces the communication overhead. Secondly, an asynchronous domain adaptation strategy is used in the proposed method, which effectively avoids idle waiting between clients and reduces additional communication overhead. Thirdly, a multi-perspective distribution discrepancy aggregation strategy is employed in the proposed method,

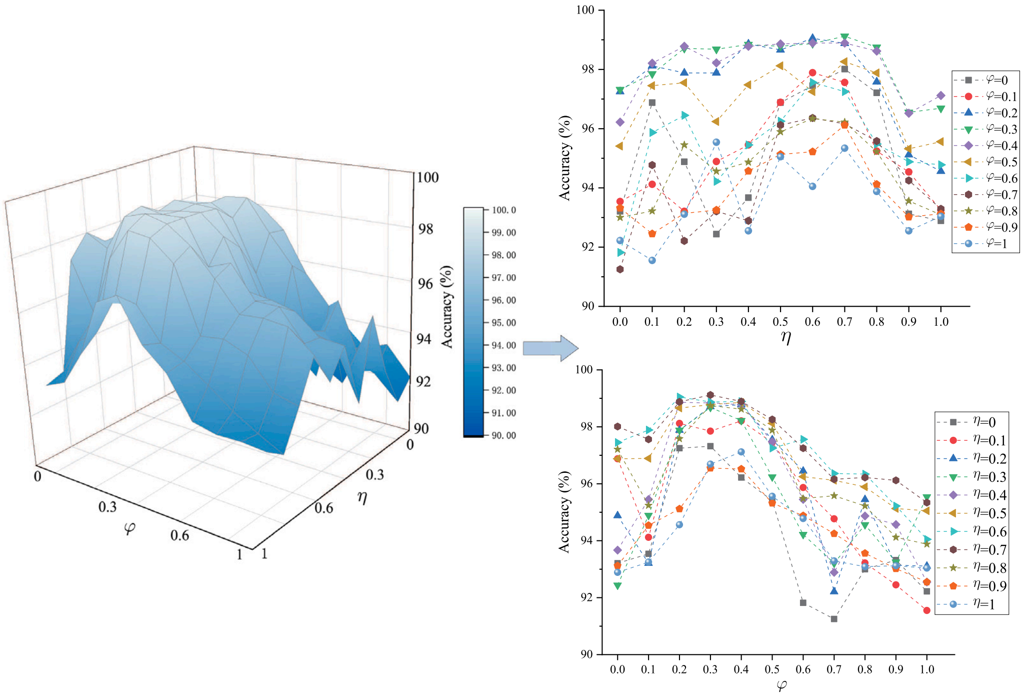


Fig. 20. Impact of different combinations of parameters η and φ on the diagnosis accuracy.

which effectively alleviates the negative transfer caused by model aggregation, thereby improving the diagnosis performance of the global model. Finally, the effectiveness of the proposed method is verified by a series of experiments. Compared with the traditional FTL method based on client–server architecture, the computation time and communication time of the proposed method are reduced by 30.92% and 58.77% respectively, and the overall training efficiency of the proposed method is improved by 44.09%. The average diagnosis accuracies obtained with the proposed method reach 99.34% and 88.26% under the cross-working condition RMFD tasks and the cross-device RMFD tasks, respectively.

In industrial applications, the bearing fault data provided by different clients may have different label spaces. In future research, a method that can effectively solve the heterogeneity of label spaces will be explored, so that RDFTL can be better applied in practical fault diagnosis.

CRediT authorship contribution statement

Lanjun Wan: Writing – review & editing, Writing – original draft, Software, Methodology, Investigation, Conceptualization. **Jiaen Ning:** Writing – original draft, Visualization, Validation, Software, Methodology, Data curation. **Yuanyuan Li:** Methodology, Software, Validation, Writing – review & editing. **Changyun Li:** Supervision, Funding acquisition. **Keqin Li:** Writing – review & editing, Methodology, Formal analysis.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by the Hunan Provincial Natural Science Foundation of China [grant number 2023JJ30217]; the Excellent Youth Foundation of Education Bureau of Hunan Province, China [grant number 21B0547]; and the National Natural Science Foundation for Young Scientists of China [grant number 61702177].

References

[1] X. Chen, R. Yang, Y. Xue, M. Huang, R. Ferrero, Z. Wang, Deep transfer learning for bearing fault diagnosis: A systematic review since 2016, IEEE Trans. Instrum. Meas. 72 (2023) 3508221.

[2] B.A. Tama, M. Vania, S. Lee, S. Lim, Recent advances in the application of deep learning for fault diagnosis of rotating machinery using vibration signals, Artif. Intell. Rev. 56 (5) (2023) 4667–4709.

[3] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, Y. Gao, A survey on federated learning, Knowl.-Based Syst. 216 (2021) 106775.

[4] W. Zhang, X. Li, H. Ma, Z. Luo, X. Li, Federated learning for machinery fault diagnosis with dynamic validation and self-supervision, Knowl.-Based Syst. 213 (2021) 106679.

[5] X. Ma, C. Wen, T. Wen, An asynchronous and real-time update paradigm of federated learning for fault diagnosis, IEEE Trans. Ind. Inform. 17 (12) (2021) 8531–8540.

[6] Y. Yu, L. Guo, H. Gao, Y. He, Z. You, A. Duan, FedCAE: A new federated learning framework for edge-cloud collaboration based machine fault diagnosis, IEEE Trans. Ind. Electron. 71 (4) (2024) 4108–4119.

[7] D. Geng, H. He, X. Lan, C. Liu, Bearing fault diagnosis based on improved federated learning algorithm, Computing 104 (2022) 1–19.

[8] J. Lin, J. Ma, J. Zhu, Hierarchical federated learning for power transformer fault diagnosis, IEEE Trans. Instrum. Meas. 71 (2022) 3520611.

[9] W. Li, R. Huang, J. Li, Y. Liao, Z. Chen, G. He, R. Yan, K. Gryllias, A perspective survey on deep transfer learning for fault diagnosis in industrial scenarios: Theories, applications and challenges, Mech. Syst. Signal Process. 167 (2022) 108487.

[10] J. Tian, D. Han, M. Li, P. Shi, A multi-source information transfer learning method with subdomain adaptation for cross-domain fault diagnosis, Knowl.-Based Syst. 243 (2022) 108466.

[11] Y. Zhang, T. Liu, M. Long, M. Jordan, Bridging theory and algorithm for domain adaptation, in: Proc. 36th Int. Conf. Mach. Learn., PMLR, 2019, pp. 7404–7413.

[12] C. Shen, X. Wang, D. Wang, Y. Li, J. Zhu, M. Gong, Dynamic joint distribution alignment network for bearing fault diagnosis under variable working conditions, IEEE Trans. Instrum. Meas. 70 (2021) 3510813.

[13] L. Wan, Y. Li, K. Chen, K. Gong, C. Li, A novel deep convolution multi-adversarial domain adaptation model for rolling bearing fault diagnosis, Measurement 191 (2022) 110752.

[14] C. He, L. Zheng, T. Tan, X. Fan, Z. Ye, Manifold discrimination partial adversarial domain adaptation, Knowl.-Based Syst. 252 (2022) 109320.

[15] C. He, X. Fan, K. Zhou, Z. Ye, Unsupervised domain adaptation with asymmetrical margin disparity loss and outlier sample extraction, Neural Netw. 168 (2023) 602–614.

[16] Z. Han, H. Sun, Y. Yin, Learning transferable parameters for unsupervised domain adaptation, IEEE Trans. Image Process. 31 (2022) 6424–6439.

[17] S. Saha, T. Ahmad, Federated transfer learning: Concept and applications, Artificial Intelligence 15 (1) (2021) 35–44.

[18] W. Yang, J. Chen, Z. Chen, Y. Liao, W. Li, Federated transfer learning for bearing fault diagnosis based on averaging shared layers, in: 2021 Glob. Reliab. Progn. Health Manag. (PHM-Nanjing), IEEE, 2021, pp. 1–7.

[19] W. Zhang, X. Li, Federated transfer learning for intelligent fault diagnostics using deep adversarial networks with data privacy, IEEE ASME Trans. Mechatron. 27 (1) (2022) 430–439.

[20] W. Zhang, X. Li, Data privacy preserving federated transfer learning in machinery fault diagnostics using prior distributions, Struct. Health Monit. 21 (4) (2022) 1329–1344.

[21] K. Zhao, J. Hu, H. Shao, J. Hu, Federated multi-source domain adversarial adaptation framework for machinery fault diagnosis with data privacy, Reliab. Eng. Syst. Saf. 236 (2023) 109246.

[22] G. Liu, W. Shen, L. Gao, A. Kusiak, Active federated transfer algorithm based on broad learning for fault diagnosis, Measurement 208 (2023) 112452.

[23] J. Chen, J. Li, R. Huang, K. Yue, Z. Chen, W. Li, Federated transfer learning for bearing fault diagnosis with discrepancy-based weighted federated averaging, IEEE Trans. Instrum. Meas. 71 (2022) 3514911.

[24] W. Zhang, Z. Wang, X. Li, Blockchain-based decentralized federated transfer learning methodology for collaborative machinery fault diagnosis, Reliab. Eng. Syst. Saf. 229 (2023) 108885.

[25] R. Wang, F. Yan, L. Yu, C. Shen, X. Hu, J. Chen, A federated transfer learning method with low-quality knowledge filtering and dynamic model aggregation for rolling bearing fault diagnosis, Mech. Syst. Signal Process. 198 (2023) 110413.

[26] A. Sergeev, M.D. Balso, Horovod: fast and easy distributed deep learning in TensorFlow, 2018, http://dx.doi.org/10.48550/arXiv.1802.05799, arXiv:1802.05799.

[27] B. McMahan, E. Moore, D. Ramage, S. Hampson, B.A. y Arcas, Communication-efficient learning of deep networks from decentralized data, in: Proc. 20th Int. Conf. Artif. Intell. Stat., PMLR, 2017, pp. 1273–1282.

[28] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, V. Lempitsky, Domain-adversarial training of neural networks, J. Mach. Learn. Res. 17 (59) (2016) 1–35.

[29] W.A. Smith, R.B. Randall, Rolling element bearing diagnostics using the case western reserve university data: A benchmark study, Mech. Syst. Signal Process. 64 (2015) 100–131.

[30] C. Lessmeier, J.K. Kimotho, D. Zimmer, W. Sextro, Condition monitoring of bearing damage in electromechanical drive systems by using motor current signals of electric motors: A benchmark data set for data-driven classification, in: Proc. Eur. Conf. PHM Soc. (PHME16), Vol. 3, PHM Society, 2016, pp. 1–17.