Computing (2022) 104:1–19
https://doi.org/10.1007/s00607-021-01019-4

REGULAR PAPER

Bearing fault diagnosis based on improved federated learning algorithm

DaoQu Geng[1,2] · HanWen He[1,2] · XingChuan Lan[1,2] · Chang Liu[1,2]

Received: 13 January 2021 / Accepted: 2 September 2021 / Published online: 20 October 2021

Abstract

Bearing fault diagnosis can be used to accurately and automatically identify the type and severity of faults. Federation learning can perform learning without transferring local data among multiple local nodes with the same data features. The traditional federated learning algorithm is difficult to identify high-quality local models among class-unbalanced local nodes, which leads to the poor quality of the training model and the slow training speed. To improve the quality and training speed of the model, this paper proposes the FA-FedAvg algorithm for fault diagnosis based on the traditional federated learning algorithm. Specifically, the weighting strategy of the model is optimized, which is conducive to increasing the weight of high-quality local models, thereby improving the quality of training models. Then, a model aggregation strategy based on precision difference is proposed to reduce the number of iterations and accelerate the convergence of the training model. Finally, the proposed algorithm is compared with FedAvg and FedProx algorithms under different data distribution conditions. The experimental results show that, compared with the comparison algorithm, the number of model training iterations of the FA-FedAvg algorithm is reduced by 52.5% on average, and the fault classification accuracy has an average increment of about 8.6%. Moreover, the FA-FedAvg fault diagnosis method is robust under different classes and data volumes.

✉ DaoQu Geng
gengdq@cqupt.edu.cn

HanWen He
s180301028@stu.cqupt.edu.cn

XingChuan Lan
s180301030@stu.cqupt.edu.cn

Chang Liu
s180301079@stu.cqupt.edu.cn

1 Key Lab of Industrial Internet of Things and Networked Control (Chongqing University of Posts and Telecommunications), Ministry of Education, Chongqing 400065, China

2 College of Automation, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

Keywords Bearing fault diagnosis · Federated learning · Convolutional neural network (CNN) · Federated averaging (FedAvg)

Mathematics Subject Classification 68T05

1 Introduction

The motor is an important component of industrial equipment. As the core of the motor, a rolling bearing is the most prone to a mechanical fault. Impact alternating load, thermal fatigue, and mechanical wear are common causes of bearing fault. Bearing health directly affects the overall performance of the equipment, so the fault diagnosis method of rolling bearing has become the focus of scholars. The current main fault diagnosis methods are categorized into four types: signal-based [1–3], data-driven [4–6], model-based [7,8], and hybrid methods.

In most industries, data often exists in the form of islands due to industry competition, privacy, and security issues. Even implementing data integration between different departments of the same company faces numerous obstacles [9]. Besides, complex industrial data usually has the following problems: (1) In the massive data, the effective information is limited, and the fault-related feature information is lacking. (2) different sampling rates or packet loss usually lead to the disappearance of observation data at specific time points. (3) Non independent and identically distributed (Non-IID) [10,11]. The data on each device is generated independently, so no local data set can represent the overall distribution. Due to the lack and complexity of training data, the existing data-driven fault diagnosis methods are usually difficult to identify more fault categories. Because the model training based on federated learning (FL) does not require the exchange of raw data, data security is ensured to a certain extent [12]. Therefore, FL is adopted in this paper to solve the problem of insufficient training data of the fault diagnosis model.

Figure 1 shows an application scenario of FL, including an aggregation node and three types of terminal devices and their local servers. Industrial data generated by different terminal devices are stored on their respective local servers for local model training. After the local model training is completed, the local server decides whether to upload it to the aggregation node for aggregation processing. FL exchanges model parameters several times between the local server and the aggregation node until the aggregated model converges to a certain accuracy [13].

Under certain conditions, the FL method is a good choice when difficult to obtain valuable data. Meng et al. [14] proposed a non-interactive efficient and Privacy-Enhanced FL scheme for IAI. FL with DNN has achieved great success in mobile calculation and IoT. Han et al. [15] used FL to train deep reinforcement learning (DRL) agents and deployed these agents on multiple edge nodes to assist the decision-making of IoT devices. FL also has good results in the security field. Nguyen et al. [16] proposed DÏoT, an autonomous self-learning distributed system for detecting compromised IoT devices, employed the FL approach to aggregate behavior profiles, and conduct intrusion detection efficiently. Lu et al. [17] proposed a privacy-preserving asynchronous FL mechanism for edge network computing, which allows multiple edge

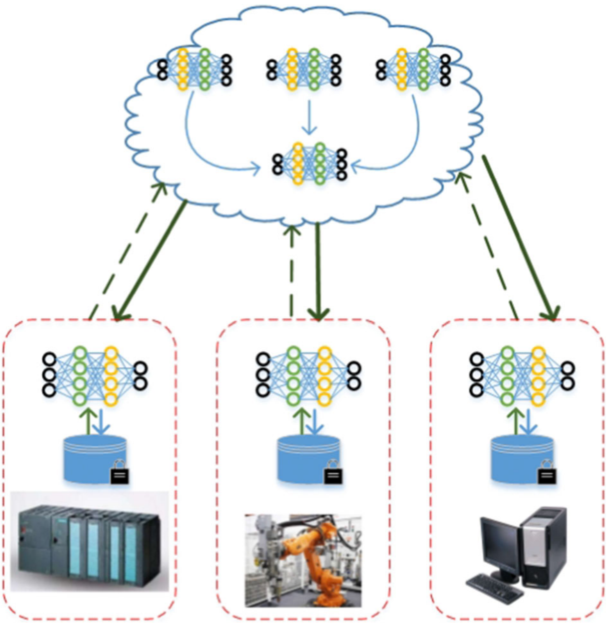

Fig. 1 Application scenarios of federated learning in industry

nodes to achieve more efficient FL without affecting accuracy without sharing their private data. Duan et al. [18] propose a self-balanced FL framework for edge systems and use data expansion and task rescheduling based on global data distribution to solve the training problem of unbalanced data. Sun et al. [19] proposed a General Gradient Sparsification (GGS) framework for improved optimizer is proposed to correct the sparse gradient update process and ultimately reduce communication overhead. Liu et al. [20] proposed a new communication-based efficient device-based FL deep anomaly detection framework is proposed for sensing time series data in IIot.

The main contributions of this paper are summarized as follows.

1) Weighted aggregation policy based on F1-scores: An improved model weighted policy is proposed. In model aggregation, local models are weighted based on the weighted F1-scores to optimize the training of unbalanced data sets. In this paper, Convolutional Neural Network (CNN) is chosen as the initial model, and the weighted policy based on the F1-scores is adopted as the aggregation method of the model. This policy improves the recognition rate of high-quality local models.
2) Improved aggregation policy: aiming at the problem of slow convergence of FedAvg during training, an improved aggregation policy based on accuracy difference is proposed.
3) The FL and the improved algorithm are applied to the industrial edge equipment, and the fault diagnosis model based on unbalanced data is trained on the distributed server.

The rest of this paper is arranged as follows. The second section briefly introduces the related work of horizontal joint learning. In Sect. 3, an improved aggregation policy based on F1-scores weighting is proposed. Firstly, a weighted policy based on F1-scores is proposed. Then this paper proposed an improved aggregation policy. Finally, the application of FL fault diagnosis based on the proposed policy is introduced. The fourth section evaluates the performance of the algorithm through simulation experiments. Finally, the conclusion is drawn in the fifth section.

2 Federated learning

For FL, the local node represents the owner of the data and the producer of the local model. And, it has sufficient autonomy to decide whether to participate in FL modeling. The local nodes and the aggregation node use the local training algorithm and aggregation algorithm respectively. The local training algorithms can be CNN, LSTM, or other models. As an aggregation algorithm of FL, federated averaging (FedAvg) is proposed by [12], which has been used in mobile phones to conduct FL on the premise of protecting users' privacy. The principles of FedAvg are detailed below.
Problem formulation In this work, we consider the following distributed optimization model.

$$\min_{w}\left\{M(w)\triangleq\sum\nolimits_{i=1}^{P}\frac{n_i}{N}M(w)_i\right\} \tag{1}$$

Where P is the number of devices, n_i and N respectively represent the number of samples of the i-th local node and the total number of samples of all local nodes, and n_i and N constitute the weight of the i-th device. $M(w)_i$ is the local objective function of the i-th device on its local data set. Assume that the i-th device saves n_i pieces of training data. The aggregation node calculates the weighted average of the model results according to formula (1) and sends the aggregated model parameter M to each local node.

The local objective function is defined by formula (2).

$$M(w)^{iter}\triangleq M(w)^{iter}-\eta\nabla l(w;b) \tag{2}$$

where $l(;)$ represents a local loss function during the local training task. $M(w)^{iter}$ is the local model. η is a constant of the step size, and $\nabla l(w;b)$ is a one-step stochastic gradient of the objective function evaluated on the data set of the i-th device. Local nodes send the locally updated model parameter M^{iter} to the aggregation node.

The pseudocode of FedAvg algorithm is shown in Algorithm 1. The following describes the specific role of pseudocode in FedAvg algorithm.

1. ClientUpdate: Receiving the initial model M_i^{init} from the aggregation node and training with the local node's data set is the main task of the client. Line 4 represents the initial model M_i^{init} received by node i from the aggregation node. Lines 5–7 indicates that local SGD will be executed in batches during the local training task, the participating nodes use the local data set to train their M_i^{iter} in parallel. After a round of training, local nodes send trained M_i^{iter} and n_i to the aggregation node for model aggregation and Convergence Check.

Algorithm 1 FedAvg

1: function ClientUpdate Run on local nodes
2: **while** $iter < max\_iter$ **do**
3: //Received initialize model M^{init} from aggregation node
4: $M_i^{iter} = M^{init}$
5: **for** batch $b \in B$ **do**
6: $M(w)_i^{iter} \triangleq M(w)_i^{iter} - \eta \nabla l(w; b)$
7: **end for**
8: $\{\mathrm{M}_i^{iter}, \mathrm{n}_i\}$ to the aggregation node
9: **if** ConvergenceCheck **then**
10: break
11: **end if**
12: $iter = iter + 1$
13: **end while**
14:
15: function ServerAggregation Run on aggregation node
16: **for** each participant $i \in P$ **do**
17: Received $\{\mathrm{M}_i^{iter}, \mathrm{n}_i\}$ from local nodes
18: **end for**
19: $M(w)^{init} \triangleq \sum_{i=1}^{P} \frac{n_i}{N} M(w)_i^{iter}$
20: Send M^{init} to local nodes

2. ServerAggregation: Collecting trained M_i^{iter} and performing model aggregation are the two tasks of the aggregation node. Lines 16–18 means to collect all the models trained by local nodes. Line 19 is a weighted model aggregation based on the number of local samples. In model aggregation, all participants' models are weighted and averaged, and then the results M_i^{init} are sent to the local nodes.

3 Approach

As a distributed learning approach, FL has great potential applications in the field of fault diagnosis. However, the traditional FedAvg algorithm only considers the difference in data volume among nodes, ignoring the distribution of different classes in the data set. The FedAvg algorithm is the key technology of model aggregation. Inspired by this, this section first proposes a weighting strategy based on F1-scores, then introduces an improved aggregation strategy based on accuracy difference, and finally proposes an improved FedAvg algorithm FA-FedAvg based on the above two strategies, and its application on fault diagnosis.

3.1 F1-scores weighting policy

Data-driven algorithms have higher requirements for training samples. Even if the same type of equipment is used for data acquisition in the industry, the distribution of equipment is usually different. Therefore, the data collected by these devices is usually unbalanced in type and quantity. However, most fault diagnosis and prediction classifiers are designed based on balanced samples, which makes the process of training and fault identification more complex. Traditional FL only considers the difference in data volume among nodes. Therefore, in the case of data imbalance, the performance problem of FL needs to be solved.

The weight of the local model needs a unified evaluation standard in the model aggregation stage. The traditional FedAvg algorithm based on balanced samples only depends on the difference of data volume between nodes. To improve the traditional FedAvg algorithm, the accuracy is added to the weight of the model aggregation, but the performance of the classifier is not significantly improved [21]. However, more factors need to be considered in the model aggregation based on unbalanced samples. During the process of model training with a large number of local nodes, the data types and numbers of these nodes are different. The number of samples of a class will affect the accuracy of prediction and classification. Specifically, if a class is held by most local nodes and has a considerable number of samples, then the class will have higher classification accuracy. On the contrary, for a class with small sample size, or a class held by only a few nodes, the classification accuracy of this class is lower. This kind of class is called missing class in this paper. In short, the initial model is difficult to learn from classes that are only stored on a few nodes. Therefore, aggregation strategies need to focus on the predicted results of a missing class. In this paper, the local model weighting policy is improved based on the traditional FedAvg algorithm, and the factors such as recall rate, precision, and missing fault class are fully considered in the weight calculation. The calculation method of local model weight is shown in formula (3).

$$
\begin{aligned}
Precision_{ij} &= \frac{TP_{ij}}{TP_{ij} + FP_{ij}} \\
F1_{ij} &= \frac{2 \times Precision_{ij} \times Recall_{ij}}{Precision_{ij} + Recall_{ij}} \\
F1_i &= \frac{\sum_{j=1}^{K} W_{ij} F1_{ij}}{\sum_{j=1}^{K} W_{ij}}
\end{aligned}
\tag{3}
$$

where K is the total number of classes of node i , TP_{ij} is True Positive, FP_{ij} is False Positive, W_{ij} means the weights of the j-th class of node i, which can be used to optimize the leaning process by manually adjusting its value. $F1_i$ is the F1-scores of the i-th local node, which can affect its own weight when the model is aggregated.

Generally speaking, for nodes with missing classes, their learning effect for missing classes is better than that for nodes without such classes. Moreover, if the number of samples of a missing class is small, it's precision and recall rate are low, and then the F1-scores of this class is reduced. As a result, the weight of nodes with this class decreases faster than that of the traditional FedAvg algorithm.

On the other hand, for the class with a large number of samples and held by most nodes, its impact on the weight is mainly reflected in the following two stages: (1) in the local model training stage, sufficient samples are conducive to the local model to learn data features better in multiple iterations. This kind of class has a higher $F1_{ij}$ value, which in turn affects the value of $F1_i$. (2) In the model aggregation stage, if a class can be well predicted by most nodes, then the initial model after aggregation will have a similar prediction effect, and can always maintain a higher $F1_{ij}$ value in

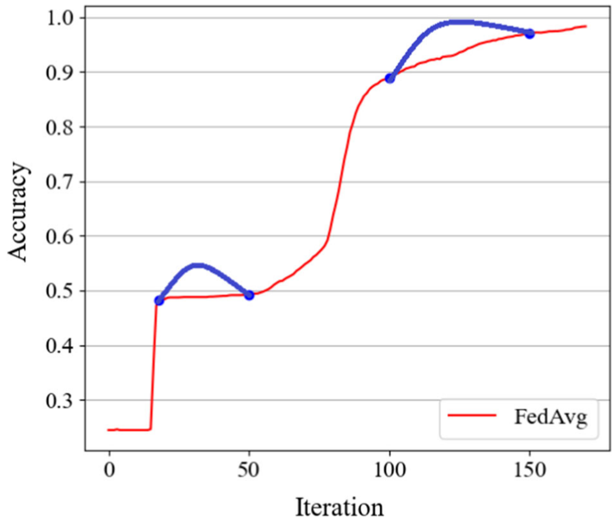

Fig. 2 The trend of training accuracy of the FedAvg algorithm

the subsequent iterations, and then affect the value of $F1_i$.

$$M^{t+1} = \frac{\sum_{i=1}^{P} n_i F1_i M_i^{t+1}}{\sum_{i=1}^{P} n_i F1_i} \tag{4}$$

The value of $F1_i$ affects the weight of model aggregation which is based on the formula (4). Where n_i means the total number of samples of node i.

3.2 Improved aggregation policy

FL can effectively improve the classification accuracy of the model. However, when the model is trained by the FedAvg algorithm, the quality of the model is reduced due to the excessive loss-function value [22]. Moreover, the lower model quality will lead to an increment in the number of model aggregation iterations, which will consume more model transmission time.

As shown in Fig. 2, the accuracy between the two pairs of blue dots improves slowly in multiple iterations. The reason for this phenomenon may be that the model quality of the local nodes participating in aggregation is poor, and the improvement of the initial model quality is still limited after successive iterations. Although the FedAvg algorithm can effectively extract the model features of local nodes, it also has the problem of slow training speed. Therefore, this paper proposes an improved aggregation algorithm based on accuracy difference, which can improve the accuracy of model aggregation and effectively reduce the total time of model transmission.

The improved aggregation strategy based on the accuracy difference proposed in this paper is shown in Fig. 3. Local nodes refer to different nodes participating in

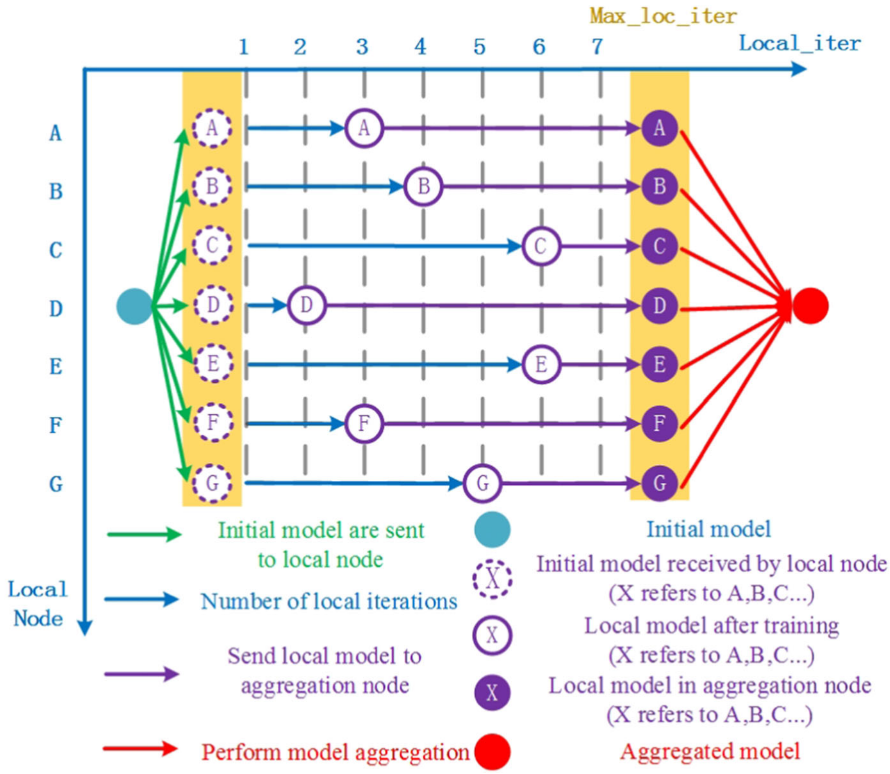

Fig. 3 Improved aggregation policy based on accuracy differences

the training. Besides, $Local\_iter$ and Max_loc_iter represent the number of local training iterations and the maximum number of local training iterations, respectively. The length of the blue arrow indicates the number of iterations of the training model of the local node, and the temporary accuracy is calculated after each iteration. If the difference between the temporary accuracy and the base accuracy is greater than the limit value $diff$, the local node will send the model parameters to the aggregation node, as shown by the purple arrows. Otherwise, the local node will continue the next model training iteration, and then determine whether the accuracy difference is greater than the limit value $diff$. Max_loc_iter is used to prevent the long-term iteration of the local model from affecting the aggregation efficiency. Compared with FedAvg aggregation policy, our improved aggregation policy ignores the details of each iteration of local nodes and pays more attention to the accuracy of the local model during training. Generally speaking, the improved aggregation policy increases the computational load of local nodes and prolongs the calculation time. However, the local model is of high quality, which can effectively reduce the aggregation times and network traffic.

3.3 FA-Fedavg algorithm and its application in fault diagnosis

Based on the weighting strategy and the aggregation strategy mentioned above, this paper proposes an improved FedAvg algorithm FA-FedAvg. The processing flow of the FA-FedAvg algorithm is as follows.

Algorithm 2 FA-FedAvg

function ClientUpdate Run on local nodes
while $iter < max\_iter$ **do**
  //Received initialize model M^{init} from aggregation node
  $Local\_iter = 1$
  $acc^{Base\_Accuracy} = M_i^{iter}.evaluate$
  while $Local\_iter < max\_loc\_iter$ **do**
    if $acc^{iter} - acc^{Base\_Accuracy} < diff$ **then**
      for batch $b \in B$ **do**
        $M(w)_i^{iter} \triangleq M(w)_i^{iter} - \eta \nabla l(w; b)$
      end for
      $acc^{iter} = M_i^{iter}.evaluate$
      Local_iter $= Local\_iter + 1$
    else
      break
    end if
  end while
  $F1_i = F1 - scores(M_i^{Local\_iter}, B)$
  Send {M_i^{iter}, n_i, $F1_i$} to the aggregation node
  if ConvergenceCheck **then**
    break
  end if
  $iter = iter + 1$
end while

function ServerAggregation Run on aggregation node
for each participant $i \in P$ **do**
  Received {M_i^{iter}, n_i, $F1_i$} from local nodes
end for
$M^{init} = \frac{\sum_{i=1}^{P} n_i F1_i M_i^{t+1}}{\sum_{i=1}^{P} n_i F1_i}$
Send M^{init} to local nodes

FA-FedAvg algorithm can be applied to the fault diagnosis of bearing and other industrial equipment. Figure 4 shows the structure of the federal fault diagnosis system. The system includes two types of nodes: aggregation node and local nodes. This paper assumes that there is a 'data island' problem between local nodes, that is, local nodes cannot transfer data to each other. As a trusted third party, the aggregation node is responsible for encryption, decryption, and aggregation of the model. As the initiator and participant of the federal training task, local nodes are responsible for training their own local model. They can be intelligent gateways and edge servers with computing power, and their training data may come from the Internet, sensors, or other data sets.

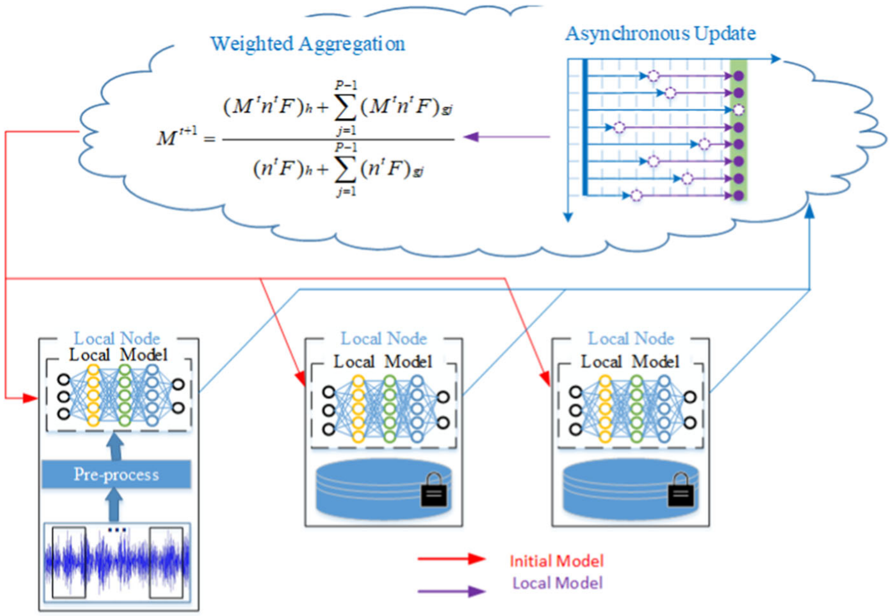

Fig. 4 A fault diagnosis application based on FA-FedAvg

Local nodes usually manage various types of data through partitions. Taking the bearing fault diagnosis as an example, in the local model training stage, each node uses the same way to preprocess the bearing vibration information stored in it. Due to the unique time sequence and periodicity of the vibration signal, image data set enhancement technology is not fully applicable. Besides, the problem of insufficient fault data is often faced. In this paper, the over-sampling data expansion method is used to preprocess the original vibration data to obtain 1n-dimensional vector data, which is used as the input of the neural network.

As shown in Fig. 4, the preprocessed data is sent to CNN for training. When the value of $Local\_iter$ is equal to that of Max_loc_iter or the accuracy difference meets the aggregation condition, this round of training is ended. Then, the F1-scores of the local model is calculated according to formula (3). Finally, the model parameters, F1-scores, and the total number of samples are encrypted and sent to the aggregation node.

After receiving the information sent by all local nodes, the aggregation node decrypts the information, and then weights and aggregates all local models based on the F1-scores weighting policy to get a new initial model. Finally, the initial model is broadcast to the participating local nodes.

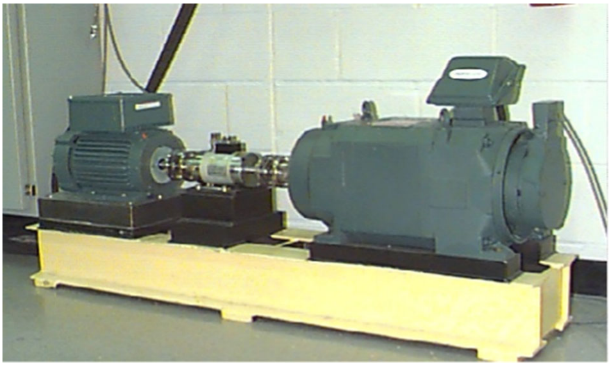

Fig. 5 The bearing fault data acquisition experimental bench

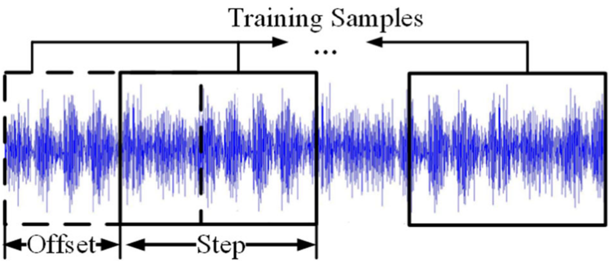

Fig. 6 The data reconstruction procedure of input data

4 Experiments and evaluations

4.1 Experiment settings

At present, bearing fault diagnosis algorithm updates quickly. To show the superiority of the FA-FedAvg algorithm, the most objective method is to use the third-party standard database for comparison. Therefore, the experimental data in this paper are all based on the open data set of the rolling bearing data center of Case Western Reserve University [23]. The experimental object is the active end bearing shown in Fig. 5. The type of bearing to be diagnosed is deep groove ball bearing skf6205. The fault bearing is made by EDM. The sampling frequency of the system is 12 kHz. There are three kinds of defects in the bearing, which are rolling element damage, outer ring damage, and inner ring damage. The damage diameters were 0.007 inches, 0.014 inches, and 0.021 inches respectively. Besides, there were 9 kinds of damage states.

At present, the commonly used feature processing methods include feature extraction based on the mathematical model [24,25] and the overlapping sampling method. In this paper, overlapping sampling is adopted for feature processing.

Table 1 Data distribution of different fault types

Damage type	Damage inch	Number of training set	Number of prediction set	Fault number
Rolling body	0.007,0.014,0.021	2100	300	1,2,3
Outer ring	0.007,0.014,0.021	2100	300	4,5,6
Inner ring	0.007,0.014,0.021	2100	300	7,8,9

As shown in Fig. 6, the original signal is processed by overlapping sampling to obtain training data. The number of samples is shown in formula (5).

$$D = \frac{N - \text{Step}}{Offset} + 1 \tag{5}$$

N is the single fault type data point, and the average value is 23236. The $Step$ is the signal length, and the average value is 864. The $Offset$ is the data sampling interval, and the average value is 28. And, 800 pieces of data for each fault type can be obtained by formula (5).

$$\tilde{x} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{6}$$

To facilitate the training of the convolution neural network, each signal is normalized. Where x is the current sample data value, x_{max} is the maximum value of the sample data, and x_{min} is the minimum value of the sample data. The normalization processing is based on the formula (6):

The data quantity of each fault type can be obtained from formula (5). The original data can be divided into 10 types and are numbered respectively. Among them, No. 1-9 corresponds to 9 fault types respectively, and No. 0 indicates a normal type. The amount of data of each type is 800, and the total amount of data is 8000. To verify the validity of the experiment, the bearing fault diagnosis data set is divided into 7000 training samples and 1000 prediction samples. There are four nodes in this experiment, including three local nodes and one aggregation node. To simulate the problem of data island in industrial computing, it is assumed that each local node has only limited types of fault diagnosis data. According to different experiments, all fault types are divided into three groups and assigned to three local nodes respectively. Finally, the normal data is added to each node. The data distribution of different fault types is shown in Table 1.

The FA-FedAvg algorithm proposed in this paper is compared with two typical algorithms FedAvg and FedProx [26]. The three algorithms have the same local training method and different aggregation methods. In the FA-FedAvg algorithm, the threshold of the accuracy difference $diff$ = 0.5, the Maximum number of local training iterations Max_loc_iter = 3.

At present, CNN is an effective deep learning model, which has been widely used in fault diagnosis. Therefore, CNN is chosen as the local training method in this paper. CNN structure can affect the training time, the quality of local models, and the

Table 2 Parameters setting for the CNN

Layer	Convolution kernel/strides
Conv1D_1	641/161
Conv1D_2	31/11
Dense_1	64
Dense_2	10

Table 3 The training accuracy and communications on IID data set

Algorithm	Communicates	Accuracy (%)
FedAvg	1476	95.6
FedProx	1224	99.1
FA-FedAvg	1182	99.5

aggregation model. For this reason, this paper selects the appropriate CNN structure as the training model after many times of optimization. The parameter settings for CNN are listed in Table 2. It has two 1D-convolution layers (the first one has a 641 convolution kernel, and the second one has a 31 convolution kernel) and 21 max-pooling layers. Then, a fully connected layer with 64 units and ReLu activation are followed. An output layer is a softmax unit. The optimizer is Adam [27,28].

4.2 Performance comparison based on IID data set

The performance of the FA-FedAvg algorithm proposed in this paper is compared with the other two algorithms FedAvg and FedProx based on an independent and identically distributed (IID) data set. Figure 7 shows the variation trend of the training accuracy of the three algorithms on the training data set with the number of iterations. Moreover, the communications of the three algorithms and their fault recognition accuracy on the test data set are shown in Table 3.

It can be seen from Fig. 7 that the number of iterations when the FA-FedAvg algorithm reaches convergence is smaller than that of the FedAvg and FedProx algorithms. Specifically, the number of iterations of the FA-FedAvg algorithm is about 200, which has a slight advantage over the FedProx algorithm. However, compared to the FedAvg algorithm, the number of iterations of the FA-FedAvg algorithm has obvious advantages, which is reduced by nearly 20%. This may be because the improvement of the FA-FedAvg algorithm in the aggregation strategy guarantees the quality of the aggregation model in a single iteration, thereby reducing the number of iterations.

Judging from the number of communications during the training process, the FA-FedAvg algorithm still has an advantage over the two comparison algorithms. As shown in Table 3, in the training process shown in Fig. 7, the number of communications of the FA-FedAvg algorithm is about 1182, which is 19.9% and 3.4% less than the number of communications of the FedAvg and FedProx algorithms, respectively. Generally, in a single-round model training process, the transfer time of model parameters accounts for a large proportion. Due to the improvement of the FA-FedAvg algorithm

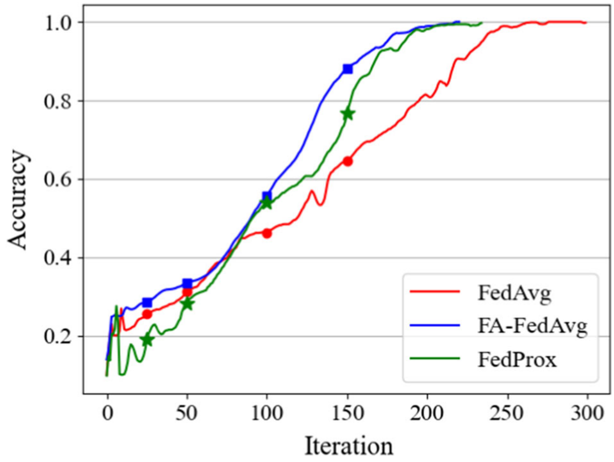

Fig. 7 Comparison of accuracy variation trends during iterative training(max_iter=300)

aggregation strategy, the number of communications is reduced, thereby reducing the training time

Table 3 shows the fault classification accuracy of the three algorithms based on the IID test set. Since the feature space of the local data set in the IID data set is similar, the local model after training has strong similarity, which reduces the difficulty of training, so the fault classification accuracy of the three algorithms is high. Specifically, the FA-FedAvg algorithm has the highest fault classification accuracy, which is an increase of 3.9% and 0.4% respectively compared with the FedAvg and FedProx algorithms. This may be because the FA-FedAvg algorithm adopts a weighting strategy based on F1-score, which makes the higher-quality local model occupy a larger proportion in the model aggregation process, and improves the quality of the aggregation model, thereby improving the fault classification accuracy of FA-FedAvg algorithm.

4.3 Performance comparison based on unbalance data set

Based on the unbalanced data set, three sets of comparative experiments are carried out under different conditions to evaluate the fault recognition accuracy and aggregation performance of the FA-FedAvg algorithm. We assume that each local node does not have enough fault type data for model training, and the missing fault type data of each node is not the same. The distribution of fault data types owned by each local node is shown in Table 4.

Where n is the number of samples of each type in the training data set, and L is the number of classes missing from the local node. In each experiment, the value of n is 700.

Besides, the test data set contains all fault types and a normal data type, and the total number of samples for all data types is 1000.

Three groups of experiments were carried out, and a single group of experiments was carried out 10 times, and the best value of the 10 experimental results was taken

Table 4 Data distribution of different fault types

ID	n	L	Node1	Node2	Node3
1	700	2	0, 1, 2, 3, 4, 5, 6, 7	1, 2, 3, 4, 5, 6, 7, 8	2, 3, 4, 5, 6, 7, 8, 9
2	700	4	0, 1, 2, 3, 4, 5	2, 3, 4, 5, 6, 7	4, 5, 6, 7, 8, 9
3	700	6	0, 1, 2, 3	2, 3, 4, 5	6, 7, 8, 9

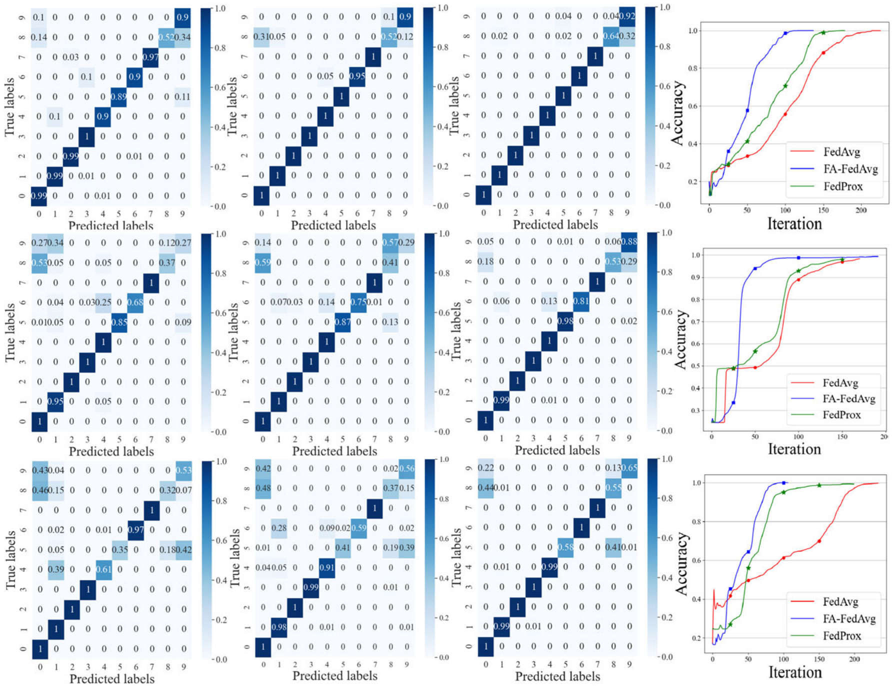

Fig. 8 Experimental curve and evaluation results. **a** ID (1) @ Table 4. **b** ID (2) @ Table 4. **c** ID (3) @ Table 4

as the final experimental result of the group. The experimental results are shown in Fig. 8 and Table 5, respectively. Among them, the confusion matrix in Fig. 8 shows the comparison of the classification accuracy of the three algorithms. Figure 8a–c correspond to the experimental results under the conditions of ID (1,2,3) in Table 4. The confusion matrix of each row corresponds to FedAvg, FedProx, and FA-FedAvg from left to right. The curve charts on the right side of Fig. 8 show the trend of the training accuracy of the three algorithms in the iterative process under different conditions. Next, the performance of the three algorithms will be compared and analyzed in terms of training accuracy change trend and fault classification accuracy.

(1) In the training accuracy change trend. The changing trend of training accuracy is shown in the curve on the right side of Fig. 8. In terms of convergence speed, FA-FedAvg algorithm converges faster, and the number of iterations at convergence is about 100, which is far less than the other two comparative algorithms. In particular,

Table 5 Results of performance evaluation

ID	FedAvg round (accuracy)	FedProx round (accuracy)	FA-FedAvg round (accuracy)
1	202 (90.5%)	152 (93.7%)	105 (95.6%)
2	211 (81.2%)	134 (83.2%)	89 (92%)
3	151 (77.8%)	143 (78.1%)	73 (87.6%)

Round: the number of iterations when the model converges
Accuracy: fault classification accuracy of model based on the test set
ID: the data set distribution number corresponding to Table 4

in the third group of experiments, the number of iterations at convergence is about half of that of the comparison algorithm. In terms of training accuracy, the convergence accuracy of FA-FedAvg algorithm is higher than that of the contrast algorithm. Specifically, at the beginning of training, the training accuracy of FA-FedAvg algorithm is lower than that of the contrast algorithm. When the number of iterations reaches about 30, the training accuracy of the three algorithms is basically the same, in the range of 0.2-0.4. After that, the training accuracy of FA-FedAvg algorithm is improved rapidly with the increase of iteration times, and the gap between the training accuracy of FA-FedAvg and that of the comparative algorithm is gradually enlarged. When the training accuracy is higher than 0.85, the increasing speed of training accuracy of FA-FedAvg gradually slows down until convergence.

On the whole, under different data distribution conditions, FA-FedAvg has a greater advantage over the comparison algorithms in the accuracy of fault classification and the number of iterations when convergence is reached. However, from the experimental results under the two data set conditions shown in Figs. 7 and 8, the number of iterations when the FA-FedAvg algorithm reaches convergence is highly dependent on the data distribution. Specifically, the number of iterations in the IID data set is large, while the number of iterations in the unbalanced data set is relatively small, and the number of iterations when it converges decreases as the number of missing classes increases. But its fault classification accuracy is gradually decreasing. As shown in Table 5, when the number of missing classes is 6, the number of iterations is 73, and the fault recognition accuracy is as low as 87.6%.

(2) In terms of fault classification accuracy. From the above three sets of experiments, as the number of nodes' missing classes increases (2 to 6 seen in Table 4), the local training difficulty of the three algorithms gradually increases, which in turn leads to a gradual decrease in their fault classification accuracy. In general, Fault classes with multiple intersections between local nodes are easier to be classified. For example, the experiment (a) and experiment (b) have 8 and 6 cross classes respectively, and the classification accuracy of the intersection classes is usually higher. On the contrary, when the number of intersection classes is small, the classification effect is poor. Taking experiment (c) as an example, the classification accuracy of fault types 5, 8, and 9 is obviously poor. Besides, the classification accuracy of fault types 0 and 1 is higher, which may be because their sample features are relatively simple and easy to be extracted.

Furthermore, in the three sets of experiments, the fault classification accuracy of the FA-FedAvg algorithm is better than the other two comparison algorithms. As shown

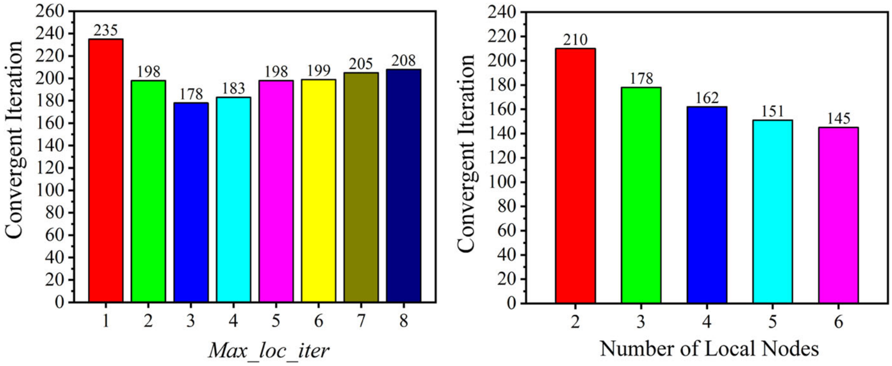

Fig. 9 The influence of Max_loc_iter and the number of local nodes

in Table 5, the average fault classification accuracy of the FA-FedAvg algorithm is between 87.6% and 95.6%, which is greatly improved compared to the other two algorithms. Especially in the experiment (b), it is 11.7% higher than that of FedAvg, and is 8.8% higher than that of FedProx. Through the analysis, we can see that the FA-FedAvg algorithm adds the weight parameter F1i which reflects the data quality of the nodes in the weighted aggregation strategy. This makes the nodes with better data quality play a greater role and solves the model training problem in the case of data imbalance to a certain extent.

On the other hand, the comparison algorithm such as FedAvg adopts the weighted method based on the amount of data. In the weighted aggregation stage, if there are nodes with a large amount of data and a poor local model, it will greatly affect the quality of the aggregation model and reduce the training effect. FA-FedAvg algorithm can make up for the deficiency in this aspect. However, the algorithm proposed in this paper also has some limitations. Through experiments, it is found that the classification accuracy of individual fault types is still low. This may be because the weighting effect of the algorithm on local high-quality models also decreases with the increase of data space heterogeneity, which needs further optimization.

4.4 Performance evaluation of Fa-Fedavg algorithm under different parameters

We evaluated the performance of the algorithm based on the IID data set under the conditions of different Max_loc_iter values and a different number of local nodes. The evaluation results are shown in Fig. 9a, b, respectively. It can be seen from Fig. 9a that the number of iterations when the model converges first decreases and then gradually increases with the increase of Max_loc_iter value. When its value is equal to 3, the number of iterations is the smallest. Through analysis, it can be seen that when the Max_loc_iter value is too large, the quality of the local model may deteriorate due to the increment in the number of training, thereby reducing the quality of the aggregation model and causing an increase in the number of iterations when the model converges. On the other hand, when the Max_loc_iter value is too small, the quality of the local model improves slowly, so that the training model requires more iterations to reach convergence. Besides, it can be seen from Fig. 9b that the number of iterations when the algorithm converges gradually decreases as the number of participating local nodes increases. This shows that more local nodes participating in model training can effectively accelerate the convergence of the training model. This conclusion is also consistent with the reference [26].

5 Final remarks

By improving the weighting and aggregation strategy of the FedAvg algorithm, this paper proposes a FA-FedAvg algorithm. Firstly, the federated learning and FedAvg algorithm are introduced. Secondly, the weighted and aggregation strategies of the FA-FedAvg algorithm are introduced in detail, and the application scenarios of the algorithm in fault identification are proposed. Finally, based on the bearing fault data set, the performance evaluation of FA-FedAvg and the comparison algorithms is carried out. The results show that the FA-FedAvg algorithm has a great improvement in the fault classification accuracy compared with the comparative algorithms.

In general, our research provides a solution to the fault classification of industrial equipment. This study also verified the feasibility of FL applications at the edge of the industry. However, other measures need further research and exploration to improve the accuracy of fault classification. For example, we can increase the input data type of the model and use new deep learning methods to explore global spatial correlation. These will be the focus of future research.

Declarations

Conflict of interest This work was supported in part by the National Key R&D Program of China under Grant 2017YFE0123000, and in part by the Project of Technological Innovation and Application Development of Chongqing Science and Technology Commission of China under Grant cstc2018jszx-cyzd0078

References

1. Niu Y, Fei J, Li Y, Wu D (2020) A novel fault diagnosis method based on EMD, cyclostationary, SK and TPTSR. J Mech Sci Technol 34:1925–1935. https://doi.org/10.1007/s12206-020-0414-y
2. Yu Y, YuDejie JC (2006) A roller bearing fault diagnosis method based on EMD energy entropy and ANN. J Sound Vib 294:269–277. https://doi.org/10.1016/j.jsv.2005.11.002
3. Ni J, Zhang C, Yang SX (2011) An adaptive approach based on KPCA and SVM for real-time fault diagnosis of HVCBs. IEEE Trans Power Deliv 26:1960–1971. https://doi.org/10.1109/TPWRD.2011.2136441
4. Shi H, Guo L, Tan S, Bai X (2019) Rolling bearing initial fault detection using long short-term memory recurrent network. IEEE Access 7:171559–171569. https://doi.org/10.1109/ACCESS.2019.2954091
5. Li J, Li X, He D (2019) A directed acyclic graph network combined with CNN and LSTM for remaining useful life prediction. IEEE Access 7:75464–75475. https://doi.org/10.1109/ACCESS.2019.2919566
6. Liu Q, Huang C (2019) A fault diagnosis method based on transfer convolutional neural networks. IEEE Access 7:171423–171430. https://doi.org/10.1109/ACCESS.2019.2956052
7. Wu Y, Jiang B, Lu N (2019) A descriptor system approach for estimation of incipient faults with application to high-speed railway traction devices. IEEE Trans Syst Man Cybern Syst 49:2108–2118. https://doi.org/10.1109/TSMC.2017.2757264

8. Wu Y, Jiang B, Wang Y (2020) Incipient winding fault detection and diagnosis for squirrel-cage induction motors equipped on CRH trains. ISA Trans 99:488–495. https://doi.org/10.1016/j.isatra.2019.09.020
9. Tao F, Qi Q, Wang L, Nee AYC (2019) Digital twins and cyber-physical systems toward smart manufacturing and industry 4.0: correlation and comparison. Engineering 5:653–661. https://doi.org/10.1016/j.eng.2019.01.014
10. Chen Y, Sun X, Jin Y (2019) Communication-efficient federated deep learning with Layerwise asynchronous model update and temporally weighted aggregation. IEEE Trans Neural Networks Learn Syst 1:10. https://doi.org/10.1109/tnnls.2019.2953131
11. Hard A, Partridge K, Nguyen C, et al (2020) Training keyword spotting models on non-iid data with federated learning. arXiv:10.21437 https://doi.org/10.21437/interspeech.2020-3023
12. Brendan McMahan H, Moore E, Ramage D et al (2017) Communication-efficient learning of deep networks from decentralized data. In: Proceedings of the 20th international conference on artificial intelligence Stat AISTATS, p 54
13. Li X, Yang W, Huang K, et al (2019) On the convergence of FedAvg on Non-IID Data. arXiv 1–26
14. Hao M, Li H, Luo X et al (2019) Efficient and privacy-enhanced federated learning for industrial artificial intelligence. IEEE Trans Ind Inform 3203:1–1. https://doi.org/10.1109/tii.2019.2945367
15. Han Y, Li D, Qi H et al (2019) Federated learning-based computation offloading optimization in edge computing-supported internet of things. ACM Int Conf Proc Ser. https://doi.org/10.1145/3321408.3321586
16. Nguyen TD, Marchal S, Miettinen M, et al (2019) DÏoT: A federated self-learning anomaly detection system for IoT. In: Proceedings-international conference on distributed computing systems, pp 756–767. https://doi.org/10.1109/ICDCS.2019.00080
17. Lu X, Liao Y, Lio P, Hui P (2020) Privacy-preserving asynchronous federated learning mechanism for edge network computing. IEEE Access 8:48970–48981. https://doi.org/10.1109/ACCESS.2020.2978082
18. Duan M, Liu D, Chen X, et al (2019) Astraea: self-balancing federated learning for improving classification accuracy of mobile deep learning applications. In: Proceedings of the 2019 IEEE international conference on computer design ICCD 2019, pp 246–254. https://doi.org/10.1109/ICCD46524.2019.00038
19. Sun H, Li S, Yu FR et al (2020) Towards communication-efficient federated learning in the internet of things with edge computing. IEEE Internet Things J 14:1–1. https://doi.org/10.1109/jiot.2020.2994596
20. Liu Y, Garg S, Nie J et al (2020) Deep anomaly detection for time-series data in industrial IoT: a communication-efficient on-device federated learning approach. 4662:1–11. https://doi.org/10.1109/JIOT.2020.3011726
21. Feng C, Wang Y, Zhao Z, et al (2020) Joint optimization of data sampling and user selection for federated learning in the mobile edge computing systems. In: Proceedings of IEEE international conference on communications workshops, pp 1–6. https://doi.org/10.1109/ICCWorkshops49005.2020.9145182
22. Bonawitz K, Salehi F, Konecny J, et al (2019) Federated learning with autotuned communication-efficient secure aggregation. In: Conf Rec - Asilomar Conf Signals, Syst Comput 2019–Novem: 1222–1226. https://doi.org/10.1109/IEEECONF44664.2019.9049066
23. http://csegroups.case.edu/bearingdatacenter/home
24. Luo F, Zhang L, Du B, Zhang L (2020) Dimensionality reduction with enhanced hybrid-graph discriminant learning for hyperspectral image classification. IEEE Trans Geosci Remote Sens 58:5336–5353. https://doi.org/10.1109/TGRS.2020.2963848
25. Luo F, Huang H, Duan Y et al (2017) Local geometric structure feature for dimensionality reduction of hyperspectral imagery. Remote Sens 9:1–23. https://doi.org/10.3390/rs9080790
26. Li T, Sahu AK, Zaheer M, et al (2020) Federated optimization in heterogeneous networks. arXiv:1812.06127v5
27. Kingma DP, Ba JL (2015) Adam: a method for stochastic optimization. In: 3rd Internatinal Confernce Learning Represent ICLR 2015-Confernce Track Proceedings, pp 1–15
28. Liu W, Chen L, Chen Y, Zhang W (2020) Accelerating federated learning via momentum gradient descent. IEEE Trans Parallel Distrib Syst 31:1754–1766. https://doi.org/10.1109/TPDS.2020.2975189