

Received January 18, 2021, accepted January 29, 2021, date of publication February 3, 2021, date of current version February 11, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3056919

Client Selection for Federated Learning With Non-IID Data in Mobile Edge Computing

WENYU ZHANG¹, XIUMIN WANG^{ID}¹, (Member, IEEE), PAN ZHOU^{ID}², (Senior Member, IEEE), WEIWEI WU^{ID}³, AND XINGLIN ZHANG^{ID}¹, (Member, IEEE)

¹School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China

²Hubei Engineering Research Center on Big Data Security, School of Cyber Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China

³School of Computer Science and Engineering, Southeast University, Nanjing 210096, China

Corresponding author: Xiumin Wang (xmwang@scut.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61872150 and Grant 61972448, in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2020A1515011209, and in part by the Science and Technology Program of Guangzhou under Grant 202002030426.

ABSTRACT Federated Learning (FL) has recently attracted considerable attention in internet of things, due to its capability of enabling mobile clients to collaboratively learn a global prediction model without sharing their privacy-sensitive data to the server. Despite its great potential, a main challenge of FL is that the training data are usually non-Independent, Identically Distributed (non-IID) on the clients, which may bring the biases in the model training and cause possible accuracy degradation. To address this issue, this paper aims to propose a novel FL algorithm to alleviate the accuracy degradation caused by non-IID data at clients. Firstly, we observe that the clients with different degrees of non-IID data present heterogeneous *weight divergence* with the clients owning IID data. Inspired by this, we utilize weight divergence to recognize the non-IID degrees of clients. Then, we propose an efficient FL algorithm, named *CSFedAvg*, in which the clients with lower degree of non-IID data will be chosen to train the models with higher frequency. Finally, we conduct simulations using publicly-available datasets to train deep neural networks. Simulation results show that the proposed FL algorithm improves the training performance compared with existing FL protocol.

INDEX TERMS Federated learning, mobile edge computing, client selection.

I. INTRODUCTION

With the rapid advancement of Internet of Things (IoT), a large volume of data has been generated at local devices [1]. These large amounts of data can be utilized to obtain useful information for the detection, classification and the prediction of future events [2]. However, due to the limited network bandwidth and the privacy concerns, it is usually impractical to transfer all the data from local devices to a remote server for processing. Fortunately, the emerging of mobile edge computing (MEC) [3], [4] makes it possible to realize the local data storing and processing at the network edges, as the edge nodes such as mobile devices, home gateways, or small cells, are in general equipped with the storage and computation capabilities. In this context, multiple edge nodes can work together with the remote cloud to perform data processing, involving both local processing at the edge nodes and global coordination at the remote cloud.

The associate editor coordinating the review of this manuscript and approving it for publication was Sotirios Goudos^{ID}.

To obtain useful information from big data in IoT, machine learning techniques are often applied [5]–[7]. Particularly, Federated Learning (FL) [8], [9], also known as federated optimization, allows multiple parties to collaboratively train a model without data sharing and gathering, fitting well with the local data storing and processing in IoT. To collaboratively learn a shared prediction model while keeping all the data on the local devices, FL iteratively asks the participating nodes, which are also referred to as clients, to download a trained model from the server, train their own models based on their local data, and update the model parameters to the server, followed by letting the server aggregate the clients' updates to further improve the model. FL ensures the safety of the users' privacy because data never leave the clients during the whole process [10].

Due to the aforementioned advantages, FL has recently been widely applied in IoT and mobile edge computing [11]–[16]. For example, [14] uses FL to train deep reinforcement learning agents in a distributed manner. The work in [15] improves the next word predictions of Google keyboard through FL. Reference [16] builds a large scale FL system

in the domain of mobile devices. However, as the training data on a given client typically relies on the usage of the mobile device by a particular user, the client's local dataset may not be representative of the population distribution, i.e., non-independent and identically distribution (non-IID). Such an imbalanced data distribution incurred by non-IID data will bring the biases in the model training, and may cause an accuracy degradation of FL [8], [17]–[24]. To explore the influence of non-IID data on FL, [8] proposes a communication-efficient FL algorithm, named FedAvg, and evaluates its performance on non-IID MNIST dataset. Later, [19] shows that FedAvg on non-IID CIFAR-10 dataset has 37% accuracy loss.

To improve the training performance on non-IID data, [19] proposes that the public data could be distributed to clients such that the clients' data become IID. However, clients are in general reluctant to install unknown data on their devices for security reasons. The work in [22] proposes a Hybrid-FL scheme, which allows a few number of clients to upload their data to a FL server. As such, an approximate IID dataset can then be constructed even if data on each client is non-IID. The server then updates the model using the data gathered from the clients, and aggregates it with the models trained locally using clients' non-IID data. However, the random selection of clients to upload their data may not form the IID dataset, due to the heterogeneous of clients on their data. The work in [24] assumes that the clients are willing to send their data distribution/label information to the server. Based on this, [24] uses mediators to reschedule the training of clients, so as to improve the accuracy of FedAvg on imbalanced data. However, clients might still be reluctant to share their data distribution or class with others, for the privacy reasons.

Depending on the applications of devices, clients usually have different classes and amounts of data. As such, the data owned by the clients in general appear heterogeneity non-IID degrees. Intuitively, to improve the accuracy, the clients with lower degree of non-IID data should participate in training with higher frequency. However, few work studies how to identify the non-IID degrees of the data at clients. Although client selection problem [25]–[28] has also been studied in the literature, none of them considers it from the perspective of their non-IID degrees of data. For example, [25] formulates a joint learning, wireless resource allocation, and user selection problem, to minimize the FL loss function. However, it mainly considers the communication effects on the training, while ignoring the heterogeneity of their data distribution.

Considering the above issues, this paper mainly aims to propose a novel FL algorithm, which can efficiently improve the accuracy based on non-IID data. Similar to [19], we assume that there exists a limited amount of approximately uniform data, which could be gathered from clients (e.g., 1% of the collection of the data from clients) or public available data. Based on this dataset, we propose a novel FL algorithm, named *CSFedAvg*, so as to efficiently leverage the clients with uniform data distribution. Particularly, we define

weight divergence to recognize the non-IID degrees of the data owned by clients, i.e., the larger the divergence with the model trained by IID data, the higher the non-IID degree of the data in clients. Then, we propose a client selection mechanism, where the clients with lower degree of non-IID data could be selected to participate in FL with much higher frequency.

The main contribution of the paper can be summarized as follows:

- We define weight divergence between clients, and observe that larger value of weight divergence indicates higher non-IID degree of data at client.
- Based on weight divergence, we propose an efficient client selection algorithm, in which clients with lower non-IID degrees of data will be more often chosen in training.
- An efficient FL algorithm, *CSFedAvg*, is then proposed, which leverages the uniform distribution users to alleviate the accuracy degradation caused by non-IID data.
- We conduct simulations to confirm the effectiveness of the proposed *CSFedAvg* on two public datasets, CIFAR-10 and Fashion MNIST datasets.

The rest of the paper is organized as follows. Section II introduces the system model and formulates the problem. We present the proposed *CSFedAvg* algorithm in Section III. Section IV presents the simulation results. In Section V, we describe the related works on FL. Finally, we conclude the paper in Section VI.

II. PRELIMINARY

In this section, we first briefly introduce the preliminary of FL. Then, we describe the system model.

A. FEDERATED LEARNING

FL is a promising distributed machine learning technique that enables mobile clients to collaboratively train a shared global model without the requirement of uploading local data to a central server. A typical training process of FL includes the following steps [8], [9]: 1) the server generates the training task, and initializes the parameters and global model to the clients; 2) based on the global model, each client respectively uses its local data to update the local model parameters, and uploads its training model to the server; 3) the server aggregates the local models, and updates the global model to the clients. Repeat the above procedure, until the global loss function converges or a desirable training accuracy is achieved.

The FL training process can be used for multiple machine learning models [33], [34], aiming at minimizing their defined loss functions. Generally, machine learning models are trained based on plenty of training samples which consist of available data and corresponding labels. The loss function of sample j can be defined as $f(\mathbf{x}_j, y_j, \mathbf{w})$, where $\mathbf{x}_j$, y_j and $\mathbf{w}$ are the vector of input sample, the corresponding label, and current weight vector, respectively. We use $f_j(\mathbf{w})$ to denote $f(\mathbf{x}_j, y_j, \mathbf{w})$ in short. The loss function $F_i(\mathbf{w})$ on dataset D_i

can be written into

$$F_i(\mathbf{w}) = \frac{\sum_{j \in \mathcal{D}_i} f_j(\mathbf{w})}{|\mathcal{D}_i|}. \quad (1)$$

Similarly, the global loss function $F(\mathbf{w})$ on the collection of all the N datasets, $\mathcal{D} = \bigcup_{i=1}^N \mathcal{D}_i$, can be written into

$$F(\mathbf{w}) = \frac{\sum_{i=1}^N F_i(\mathbf{w})}{|\mathcal{D}|}, \quad (2)$$

which needs to be minimized.

Minibatch stochastic gradient descent (minibatch SGD) can be used to solve the federated optimization. Denote $\mathbf{w}_i(t-1)$ as the model of client i in round $t-1$. Then, client model $\mathbf{w}_i(t)$ in the t -th round can be updated as follows,

$$\mathbf{w}_i(t) = \mathbf{w}_i(t-1) - \eta_t \nabla F_i(\mathbf{w}_i(t-1)) \quad (3)$$

where η_t is the learning rate in round t . Then, the server aggregates the local models received from the clients, into a new global model $\mathbf{w}(t)$. For example, FedAvg [8] generates the global model using the weighted average of models of clients, i.e.,

$$\mathbf{w}(t) = \sum_{i=1}^K \frac{|\mathcal{D}_i|}{|\mathcal{D}|} \mathbf{w}_i(t). \quad (4)$$

B. SYSTEM MODEL

This paper considers a general IoT system, where there is a server s and K edge nodes. We use $\mathbb{K} = \{c_1, c_2, \dots, c_K\}$ to denote the set of the K edge nodes. For each edge node $c_i \in E$, suppose that it has a local dataset $\mathcal{D}_i$. For the simplicity, we use $|\cdot|$ to denote the size of the dataset, and denote $\mathcal{D}$ as the data collection of all the edge nodes, i.e. $\mathcal{D} = \bigcup_{i=1}^K \mathcal{D}_i$. As edge nodes in general have the capabilities of performing local data processing, the server can collaboratively work with them to train a model over the data residing on them. Similar to [25], for uplink, we assume each client transmits its local FL model parameter to the server based on an orthogonal frequency division multiple access (OFDMA) technique. The downlink achieved by the BS when transmitting the parameters of global FL model to each client is constrained by its bandwidth, transmission power and the interference. As the communication issue is not the main concern of this paper, we omit the details for the simplicity.

Unlike traditional distributed learning, datasets on clients in FL have unique properties which make them imbalanced. One of the properties is that the local data on clients are not always IID. Several previous works [19] demonstrate that the non-IID data can result in the accuracy loss of model trained with FedAvg. Besides that, depending on the usage of a particular user, the datasets owned by clients are usually with different classes and amounts, appearing heterogeneous non-IID degrees. Reference [19] further analyzes that the accuracy reduction caused by extremely non-IID data is more than slightly non-IID data. Inspired by this observation, there are the following two questions:

- How to differentiate the clients according to the degrees of their non-IID data, and choose them in the model training?
- If we let the clients with lower degree of non-IID data participate in the training more often than the clients with higher degree of non-IID data, whether the accuracy of FL can be improved or not?

Based on the above questions, this paper aims to design a novel FL algorithm, named *CSFedAvg*, which identifies the clients with heterogeneous degrees of non-IID data and chooses them to participate in the training, aiming at improving the accuracy of the learning models.

III. THE DESIGN OF CSFedAvg

In this section, we first introduce the divergence among client models influenced by their heterogeneity data distribution. Then, we present the framework of the proposed algorithm, named Client-Selected Federated Averaging (*CSFedAvg* in short). Finally, we introduce the detail of the proposed *CSFedAvg*.

A. OBSERVATION ON WEIGHT DIVERGENCE BETWEEN CLIENT MODELS

Generally, the convergence trends of client models trained by IID data are close with each other, because they should follow the similar trend as the centralized training model using IID data [19]. Hence, an intuition is that the divergence of models between clients with all IID data should be smaller than one of clients with non-IID data. Inspired by the above observations, we define the weight divergence $d_{m,n}(t)$ between client m and n in the t -th iteration as follows,

$$d_{m,n}(t) = \frac{\|\mathbf{w}_m(t) - \mathbf{w}_n(t)\|}{\|\mathbf{w}_n(t)\|}, \quad (5)$$

where $\mathbf{w}_m(t)$, $\mathbf{w}_n(t)$ denote the weights of client m and client n at the t -th iteration.

To clarify whether the non-IID data of the client affects its weight divergence with the client holding the IID-data, we conduct a simulation to compare the weight divergences of the clients. Particularly, we generate 100 clients, and equally divide them into two sets, denoted by C_1 and C_2 respectively. The clients in the first set C_1 are all with non-IID data, while the clients in C_2 hold IID data. Furthermore, we set a new client denoted by c_0 with IID data as a comparable baseline. We calculate the weight divergences of these 100 clients with client c_0 during the training process using the existing FedAvg. Then, we present the average weight divergences of these two kinds of clients in each communication round. As shown in Fig. 1, we conduct simulations on two datasets, CIFAR-10 [35] and Fashion MNIST [36]. It is observed that the average weight divergence of the clients with non-IID data is always higher than the clients with IID data, which confirms our intuition. To further verify the above observation, we also study the weight divergence of clients to centralized training. As shown in Fig. 2, We present the weight divergence of 10 clients with IID data, and 10 clients with non-IID data, respectively. It is shown that the weight

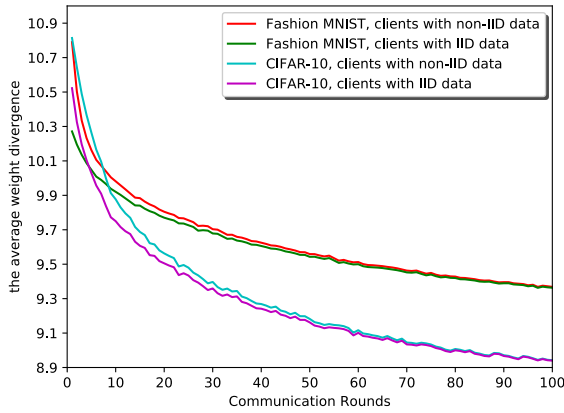


FIGURE 1. The difference of weight divergence between clients with IID and non-IID data.

divergence of client with IID data to the centralized training is always less than client with non-IID data. Furthermore, the weight divergence of clients with IID data is very close to each other.

The weight divergence can help us differentiate the non-IID degrees of the data at clients, i.e., larger value of weight divergence means higher non-IID degree of data in client. To compare the weight divergences of clients in the system, this paper introduces an auxiliary IID data at the federated server, which serves as an auxiliary client with IID data. Such auxiliary data can be collected from public dataset or purchased from a small number of clients. These data are trained at the federated server. After receiving the local model from clients, the server can then calculate the weight divergence using the local model received from clients and the model trained with the auxiliary data. Based on the weight divergences between the training clients and the auxiliary client, we can differentiate the clients according to the degrees of their non-IID data. Although weight divergence is also defined in [19], [20], it is mainly used to present the divergence between FL and centralized training. None of previous works uses weight divergence to differentiate clients and improves the performance of FL.

It is worth noting that training the auxiliary data may introduce extra overhead, e.g., computation at the federated server. Fortunately, considering that the server is in general equipped with strong computation capability, such extra computation overhead could be neglected. In addition, as the weight divergence is locally calculated at the federated server, the training model trained by the auxiliary data does not incur any communication overhead.

B. THE FRAMEWORK OF CSFEDAVG

To ease understanding, we firstly briefly present the framework of CSFedAvg. As shown in Fig. 3, we assume that at the federated server, there exists a small amount of IID data, called *auxiliary data*, which is collected from public dataset or purchased from users. With the help of auxiliary data at the federated server, we then calculate the weight divergence of clients. Based on this, the proposed CSFedAvg can

continually choose clients with IID data or low non-IID data from the clients during the training process. The federated server records two kinds of client sets, i.e., the *client candidate set* and the *final selected client set*. Specifically, the client candidate set includes the clients that are the candidates of the final selected clients. The details of their selections will be introduced in the next subsection.

The main procedure of CSFedAvg algorithm in each communication round can be summarized as follows.

- **Step 1 (Global Model Downloading):** The federated server firstly randomly choose some clients. These randomly chosen clients and the clients in the client candidate set together download the global model from the federated server, and participate in the training.
- **Step 2 (Client Model Update):** After receiving the global model from the server, each participated client respectively trains and updates its own client model using its local data. Simultaneously, the federated server uses its auxiliary data to update the auxiliary model based on the global model.
- **Step 3 (Client Set Update and Model Aggregation):** Using the weight divergence between the auxiliary model and client models uploaded by the participated clients, the federated server updates the client candidate set and the final selected client set, according to a certain rule. Based on this, the federated server aggregates the models trained by the randomly chosen clients and the final selected clients. For the clients in the client candidate set, if they are not chosen into the final selected client set, their models will be abandoned to avoid the accuracy loss due to their non-IID data.

Continue the above procedure until achieving a target accuracy or running a given number of communication rounds.

C. THE ALGORITHM DESIGN OF CSFEDAVG

We now introduce the design of CSFedAvg in details.

At the federated server, it maintains three kinds of client sets: 1) random client set $\mathbb{N}_t$, 2) client candidate set $\mathbb{C}_t$, and 3) final selected client set $\mathbb{S}$. Specifically, $\mathbb{N}_t$ includes the randomly selected clients at round t , $\mathbb{C}_t$ denotes the candidates of the final selected clients at round t . The federated server continually selects the clients with low non-IID data from $\mathbb{C}_t$ to $\mathbb{S}$, until the number of clients in $\mathbb{S}$ is enough. Here, we use p to denote the proportion of clients randomly selected into $\mathbb{N}_t$, where $0 < p \leq 1$, i.e., $|\mathbb{N}_t| = pK$. In addition, we set that the maximum number of selected clients in $\mathbb{S}$ should not exceed αK , where α is the selection factor to determine the size of $\mathbb{S}$. Besides that, we assume the server has an auxiliary data which can serve as an auxiliary client to help differentiate the weight divergences of clients. For the simplicity, we refer to such an auxiliary client as client 0, denoted by c_0 in the following section. Then, we use $\mathbf{w}_k(t)$ to denote the training model of client k at round t , where $\mathbf{w}_0(t)$ represents the training model based on the auxiliary data.

Initially, we have $\mathbb{N}_t = \mathbb{C}_t = \mathbb{S} = \emptyset$. The federated server initializes the global model $\mathbf{w}(0)$ and auxiliary model

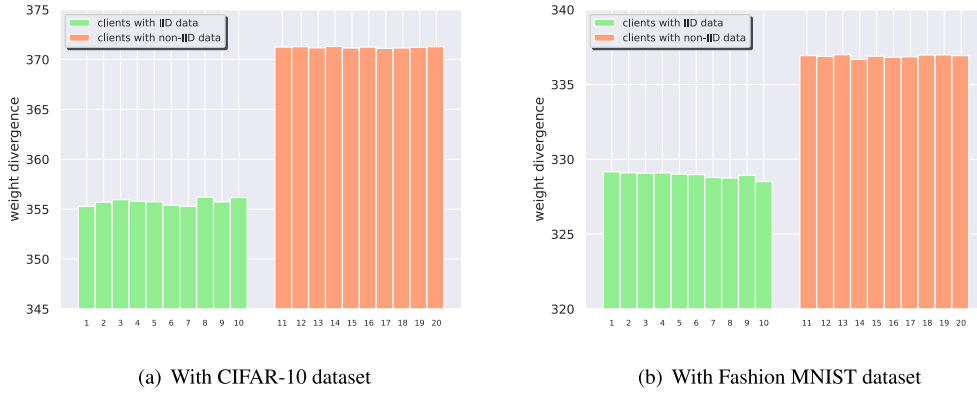


FIGURE 2. The weight divergence between clients with IID or non-IID data and centralized training, based on CIFAR-10 dataset and Fashion MNIST dataset.

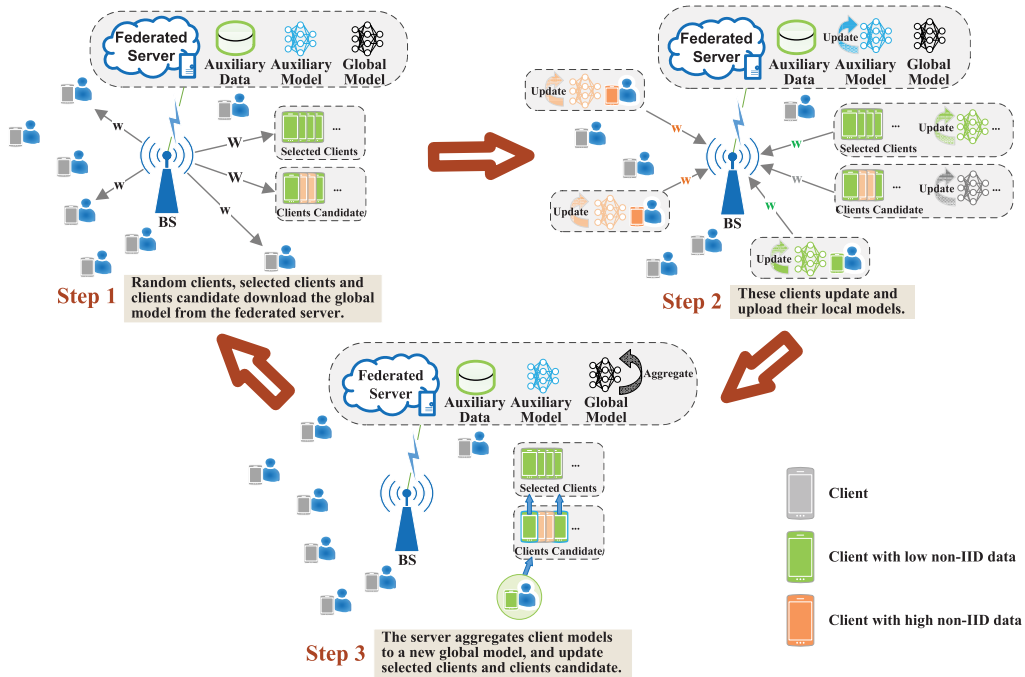


FIGURE 3. The framework of CSFedAvg.

$\mathbf{w}_0(t)$. Then, the CSFedAvg runs round by round, and in each communication round t , it conducts the following procedure:

- The federated server randomly choose pK clients from $\mathbb{K}$ to constitute the random client set $\mathbb{N}_t$, where $0 < p \leq 1$. The clients in $\mathbb{N}_t$, $\mathbb{C}_t$ and $\mathbb{S}$, download the global model and update their client models using local data. Simultaneously, the federated server trains the model $\mathbf{w}_0(t)$ based on the previous auxiliary model $\mathbf{w}_0(t-1)$ with auxiliary data.
- The clients in $\mathbb{N}_t$, $\mathbb{C}_t$ and $\mathbb{S}$, upload their training models to the federated server.
- After receiving the training models from the clients, the federated server calculates the weight divergence between each training client and the auxiliary client c_0 , based on their training models $\mathbf{w}_k(t)$ and $\mathbf{w}_0(t)$. For example, the weight divergence of client k and auxiliary

client c_0 can be denoted as $d_k(t)$, which is

$$d_k(t) = \frac{\|\mathbf{w}_k(t) - \mathbf{w}_0(t)\|}{\|\mathbf{w}_0(t)\|}. \quad (6)$$

- Based on the above calculation, the federated server re-orders client candidates in $\mathbb{N}_t$, according to the non-decreasing order of their weight divergences, i.e., $\mathbb{N}_t = \{c_{k'_1}, c_{k'_2}, \dots\}$, where $d_{k'_1}(t) \leq d_{k'_2}(t) \leq \dots$. If the final selected clients in $\mathbb{S}$ are still not enough, i.e., $|\mathbb{S}| < \alpha K$, the server will select the clients from $\mathbb{C}_t$ to $\mathbb{S}$ as follows:
 - For each client candidate in $\mathbb{C}_t$, e.g., $c_k \in \mathbb{C}_t$, if its weight divergence $d_k(t)$ is lower than the first $\alpha \cdot 100\%$ smallest weight divergences of clients in $\mathbb{N}_t$, it will be added into the final selected

client set $\mathbb{S}$, i.e.,

$$d_k(t) \leq d_{k'}(t), \text{ where } i = \lfloor \alpha |\mathbb{N}_t| \rfloor. \quad (7)$$

- Continue the above step, until $|\mathbb{S}| = \lfloor \alpha K \rfloor$ or all the clients in $\mathbb{C}_t$ have been considered.

The server then updates the client candidate set, by directly adding the first $\lfloor \alpha |\mathbb{N}_t| \rfloor$ clients in $\mathbb{N}_t$ into $\mathbb{C}_t$.

- The server aggregates the models using the following rules:

- If the final selected clients in $\mathbb{S}$ are still not enough, the server will only aggregate the models trained by the randomly selected clients in $\mathbb{N}_t$ and the final selected clients in $\mathbb{S}$, i.e.,

$$\mathbf{w}(t) = \sum_{c_k \in \mathbb{N}_t \cup \mathbb{S}} \frac{|\mathcal{D}_k|}{|\mathcal{D}|} \mathbf{w}_k(t),$$

$$\text{where } |\mathcal{D}| = \sum_{c_k \in \mathbb{N}_t \cup \mathbb{S}} |\mathcal{D}_k|. \quad (8)$$

It is worth noting that the auxiliary model $w_0(t)$ is not aggregated in the current round, due to the fact that the auxiliary IID data still needs to be used to select the IID clients in the next round.

- If the final selected clients in $\mathbb{S}$ are enough, the server will aggregate the auxiliary model $w_0(t)$, and the models trained by the randomly selected clients in $\mathbb{N}_t$ and the final selected clients in $\mathbb{S}$, i.e.,

$$\mathbf{w}(t) = \frac{|\mathcal{D}_0|}{|\mathcal{D}|} \mathbf{w}_0(t) + \sum_{c_k \in \mathbb{N}_t \cup \mathbb{S}} \frac{|\mathcal{D}_k|}{|\mathcal{D}|} \mathbf{w}_k(t),$$

$$\text{where } |\mathcal{D}| = |\mathcal{D}_0| + \sum_{c_k \in \mathbb{N}_t \cup \mathbb{S}} |\mathcal{D}_k|. \quad (9)$$

Continue the above procedure in the next communication round, until achieving the desirable accuracy or the training time expired. The detailed process is presented in Algorithm 2.

IV. SIMULATION RESULTS

In this section, we evaluate the performance of the proposed CSFedAvg through simulations based on the public datasets.

A. SETUP

We adopt two available datasets in the simulations, which are commonly used in the literature. The first dataset is CIFAR-10 [35], which includes 50,000 training images and 10,000 testing images in 10 classes, with 6000 images per class. The second dataset is Fashion MNIST [36], which contains 70,000 gray-scale images (including 60000 training images and 10,000 testing images) from 10 categories, with 7,000 images per category. These two datasets will be distributed to $K = 500$ clients, with each client holding 500 samples of CIFAR-10, and each client holding 600 samples of Fashion MNIST.

Similar to previous work [19], the IID data means the client is randomly assigned a uniform distribution over all the classes of the dataset. Alternatively, the non-IID data

Algorithm 1 The Design of Client-Selected Federated Averaging (CSFedAvg)

```

1: procedure Client-SelectedFederatedAveraging
2:   Initialize  $t = 0$ ,  $\mathbf{w}(t)$ ,  $\mathbf{w}_0(t) \leftarrow \mathbf{w}(t)$ ,  $\mathbb{S} \leftarrow \{\}$ ,  $\mathbb{C}_t \leftarrow \{\}$ ;
3:   for each communication round  $t = 1, 2, \dots$  do
4:      $\mathbb{N}_t \leftarrow$  (Randomly select  $pK$  clients from  $\mathbb{K}$ );
5:     for each client  $c_k \in \mathbb{N}_t \cup \mathbb{C}_t \cup \mathbb{S}$  paralelly do
6:        $\mathbf{w}_k(t) \leftarrow \text{ClientUpdate}(c_k, \mathbf{w}(t-1))$ ;
7:     end for
8:     if  $|\mathbb{S}| < \alpha K$  then
9:       The server updates  $\mathbf{w}_0(t)$  using  $(\mathbf{w}_0(t-1))$ ;
10:      Update  $\mathbb{S}$  and  $\mathbb{C}_t \leftarrow$  Algorithm 2;
11:       $\mathbf{w}(t) \leftarrow \sum_{c_k \in \mathbb{N}_t \cup \mathbb{S}} \frac{|\mathcal{D}_k|}{|\mathcal{D}|} \mathbf{w}_k(t)$ ;
12:     else
13:       The server updates  $\mathbf{w}_0(t)$  using  $(\mathbf{w}(t-1))$ ;
14:        $\mathbf{w}(t) \leftarrow \frac{|\mathcal{D}_0|}{|\mathcal{D}|} \mathbf{w}_0(t) + \sum_{c_k \in \mathbb{N}_t \cup \mathbb{S}} \frac{|\mathcal{D}_k|}{|\mathcal{D}|} \mathbf{w}_k(t)$ ;
15:     end if
16:   end for
17: end procedure
18:
19: function ClientUpdate( $c_k, \mathbf{w}(t-1)$ )
20:    $\mathbf{w}_k(t-1) = \mathbf{w}(t-1)$ ;
21:   for each local training epoch from 1 to  $E$  do
22:     for batch  $b \in B$  do
23:        $\mathbf{w}_k(t) \leftarrow \mathbf{w}(t-1) - \eta_t \nabla F_k(\mathbf{w}(t-1), b)$ ;
24:     end for
25:   end for
26:   return  $\mathbf{w}_k(t)$ ;
27: end function

```

indicates that the client holds the data from only a few limited classes. Then, in the simulation, we investigate three settings of clients, named *IID setting*, *Hybrid I setting* and *Hybrid II setting*, respectively. For the IID setting, the samples of each client are randomly generated with uniform data distribution, i.e., IID data. For both Hybrid I and Hybrid II settings, only a small number of clients, $\gamma \cdot 100\%$ of clients ($\gamma = 0.06$ in this simulation), are assigned with IID data, while other clients are randomly assigned by images from a few certain limited classes. In other words, Hybrid I setting includes two kinds of clients, one with IID data, and the other with samples from a single class/category, i.e., non-IID data. Similarly, Hybrid II setting also contains two kinds of clients, one with IID data, while the other with samples from two classes/categories. Compared with Hybrid I setting, the collection of the data in Hybrid II setting is more uniform, although it is still non-IID.

We adopt a convolutional neural network (CNN), which is a class of deep neural networks and commonly applied in the literature [8], [12], [13], [23], [24]. Specifically, we set four 3×3 convolution layers (32, 32, 64, 64 channels, each of which was activated by ReLU and batch normalized, and every two of which were followed by 2×2 max pooling), two fully connected layers (384 and 192 units with

Algorithm 2 Update Selected Client Set and Candidate Client Set

```

1: // Update the final selected client set  $\mathbb{S}$ .
2: if  $\mathbb{C}_t \neq \emptyset$  then
3:   for  $c_k(t) \in \mathbb{C}_t$  and  $c_j(t) \in \mathbb{N}_t$  do
4:      $d_k(t) = \frac{\|\mathbf{w}_k - \mathbf{w}_0(t)\|}{\|\mathbf{w}_0(t)\|}$ ;
5:      $d_j(t) = \frac{\|\mathbf{w}_j - \mathbf{w}_0(t)\|}{\|\mathbf{w}_0(t)\|}$ ;
6:   end for
7:   Resort clients in  $\mathbb{N}_t$  in their weight divergences,
8:    $\mathbb{N}_t = \{c_{k'_1}, c_{k'_2}, \dots\}$ , where  $d_{k'_1}(t) \leq d_{k'_2}(t) \leq \dots$ ;
9:   for  $c_k(t) \in \mathbb{C}_t$  do
10:    if  $d_k(t) \leq d_{k'_i}(t)$ , where  $i = \lfloor \alpha |\mathbb{N}_t| \rfloor$  then
11:       $\mathbb{S} \leftarrow \mathbb{S} \cup \{c_k\}$ ;
12:    end if
13:     $\mathbb{C}_t \leftarrow \mathbb{C}_t \setminus \{c_k\}$ ;
14:  end for
15: end if
16: // Update the client candidate set  $\mathbb{C}_t$ .
17: Add the first  $\lfloor \alpha |\mathbb{N}_t| \rfloor$  clients in  $\mathbb{N}_t$  into  $\mathbb{C}_t$ ;
18: return  $\mathbb{S}, \mathbb{C}_t$ ;

```

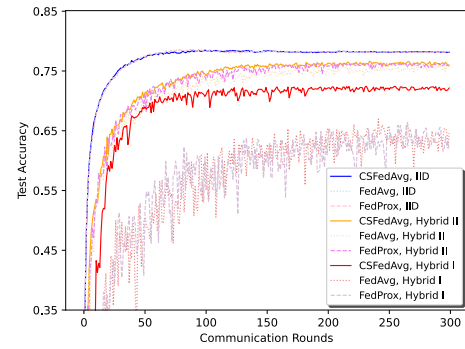
ReLU activation), and a final output layer with 10 units for training both CIFAR-10 and Fashion MNIST datasets. As presented in the algorithm, only a fraction of clients will be randomly selected in each communication round, where we set the fraction p to be 0.1 without specific demonstration. For local training, clients employ mini-batch SGD to update models. Similar to [8], the initial learning rate, the learning-rate decay, the mini-batch size, and the local epoches are set to be 0.05, 0.992, 32 and 5, respectively. Although adjusting the parameters may improve the performance, it is out of scope of this paper, as we mainly focus on how to improve the training accuracy under non-IID dataset at clients.

For fairly comparison, we introduce two baseline algorithms, named FedAvg and FedProx. Specifically, FedAvg is a classic FL algorithm that has been applied in Google's Gboard [37] to improve the next-word prediction. FedProx is proposed in [31], which tackles the heterogeneity in federated networks. By adding a proximal term in the objective, FedProx aims to improve the stability of FL. In the following simulations, we will implement the training of CNN models using both the FedAvg and the proposed CSFedAvg algorithms, under three different dataset settings.

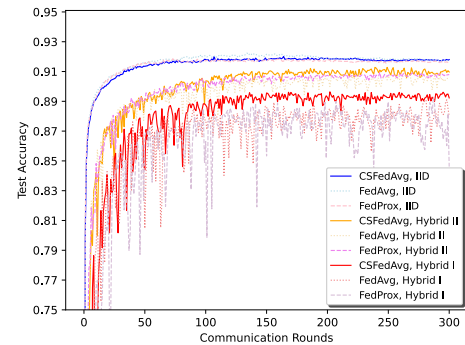
B. PERFORMANCE COMPARISON ON TRAINING ACCURACY

We now evaluate the performance of the proposed CSFedAvg in improving the test accuracy, by comparing with the baseline algorithm. To ease understanding, we first study the case when the selection factor of CSFedAvg equals the proportion of clients with IID data in both Hybrid I and Hybrid II settings, i.e., $\alpha = \gamma = 6\%$.

As shown in Fig. 4, it is observed that in ideal FL environment, i.e., in IID setting, the gap between CSFedAvg and FedAvg is negligible. The reason is that when the clients are



(a) With CIFAR-10 dataset



(b) With Fashion MNIST dataset

FIGURE 4. The accuracy comparison over communication rounds under IID, Hybrid I and Hybrid II settings, based on (a) CIFAR-10 dataset and (b) Fashion MNIST dataset.

all with IID data, the weight divergence cannot be used to differentiate clients, and the proposed CSFedAvg achieves similar performance as FedAvg. Alternatively, in Hybrid II setting, there exist clients with non-IID data. We can see that the accuracy of the CNN model trained by CSFedAvg is a little higher than both FedAvg (improving about 0.35% and 0.18% accuracy respectively for CIFAR-10 and Fashion MNIST datasets) and FedProx. Compared with Hybrid II setting, The data distribution of clients with Hybrid I setting is more extreme. In this context, the performance improved by the proposed CSFedAvg is more significant than in Hybrid II setting. Among these three schemes, FedAvg performs the worst. Compared with FedProx, the proposed CSFedAvg under Hybrid I setting improves more than 5.3% and 0.52% accuracy based on CIFAR-10 and Fashion MNIST datasets, respectively. Although FedAvg may outperform our algorithm with IID data, its performance will be significantly degraded in the context of non-IID data. By appropriately selecting the clients with IID data, the proposed algorithm achieves much better accuracy than the baseline algorithms.

There are two reasons why CSFedAvg can mitigate more accuracy loss on Hybrid I setting. Firstly, clients with higher degree of non-IID data will lead to larger weight divergence, when the number of clients with IID data and non-IID data is constant. Therefore, it is convenient for CSFedAvg to select

TABLE 1. Number of communication rounds to reach a target accuracy for CSFedAvg and FedAvg on CIFAR-10.

ToA@ x	Hybrid I			Hybrid II			IID		
	FedAvg	FedProx	CSFedAvg	FedAvg	FedProx	CSFedAvg	FedAvg	FedProx	CSFedAvg
ToA@0.6	95	93	24	18	20	16	4	5	4
ToA@0.7	—	—	58	45	44	41	13	14	12
Accuracy	0.669	0.675	0.725	0.731	0.739	0.766	0.785	0.783	0.785

TABLE 2. Number of communication rounds to reach a target accuracy for CSFedAvg and FedAvg on Fashion MNIST.

ToA@ x	Hybrid I			Hybrid II			IID		
	FedAvg	FedProx	CSFedAvg	FedAvg	FedProx	CSFedAvg	FedAvg	FedProx	CSFedAvg
ToA@0.88	68	66	52	26	27	23	4	5	5
ToA@0.89	279	286	102	44	44	40	7	8	8
Accuracy	0.891	0.893	0.896	0.907	0.908	0.909	0.918	0.917	0.918

the clients with IID data faster and more accurately. Secondly, clients with higher degree of non-IID data can accelerate the degradation of the global model performance. In this context, the proposed CSFedAvg algorithm, which selects clients with IID data to reduce the impact of uneven data distribution, is more advantageous.

C. PERFORMANCE COMPARISON ON THE NUMBER OF COMMUNICATION ROUNDS

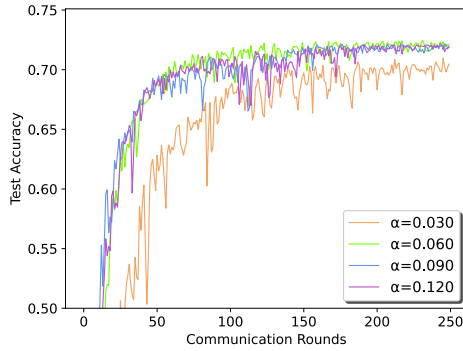
We now evaluate the performance of the proposed CSFedAvg on the number of communication rounds required to reach a target accuracy. Similar to [26], we record the Time of Arrival at a desired accuracy, denoted by ToA@ x , where x denotes the target accuracy. Similarly, we measure the accuracy of each algorithm after running 300 rounds, denoted by *Accuracy* in the table.

Table 1 and Table 2 present the number of communication rounds required to achieve desired accuracy by three schemes, FedAvg, FedProx and CSFedAvg, on CIFAR-10 and Fashion MNIST datasets, respectively. It is particularly noted that “—” in Table 1 and Table 2 denotes the target accuracy cannot be achieved within a finite number of communication rounds. That is, both FedAvg and FedProx cannot achieve 70% accuracy in Hybrid I setting, under CIFAR-10 dataset. It is also observed that the proposed CSFedAvg accelerates the convergence and improves the stability of the global model, in both Hybrid I and Hybrid II settings. Although FedAvg performs well in IID setting, the gap between them is quite small and could be negligible. It is also observed that the proposed CSFedAvg achieves better accuracy after running 300 communication rounds in both Hybrid I and Hybrid II settings, compared with both FedAvg and FedProx.

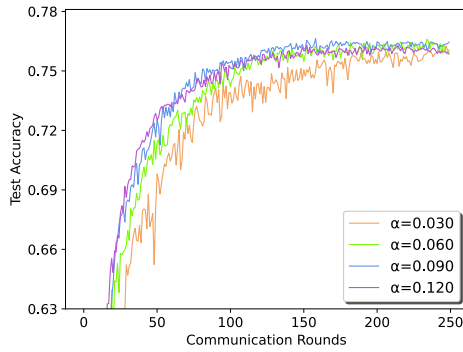
D. THE IMPACT OF SELECTION FACTOR α ON ACCURACY AND COMMUNICATION ROUNDS

The selection factor α is the percentage of clients selected by the server. Intuitively, the selection factor should be as close as possible to the proportion (which is set as γ in our simulation) of clients with IID data among all the clients. Although it is not easy to get the exact value of the proportion of clients with IID data, we can still estimate it by allowing clients to upload the types of their data. Then, the selection factor can be adjusted around this proportion, to optimize the performance.

Fig. 5 and Fig. 6 illustrate the accuracy trained by CSFedAvg with different values of selection factor α , based on CIFAR-10 and Fashion MNIST datasets, respectively. In this simulation, the proportion of clients with IID data is approximate to be 0.06, i.e., $\gamma = 0.06$. As estimated, the accuracy achieved in Hybrid I setting is the best when the selection factor is equal to the proportion of clients with IID data, i.e., $\alpha = 0.06$. However, in Hybrid II setting, the best accuracy is achieved when the selection factor is 0.09, a little bit higher than 0.06. This is also reasonable, as compared with Hybrid I setting, it is more difficult to identify the IID clients in Hybrid II setting. Hence, in Hybrid II setting, slightly higher selection factor can increase the probability of selecting IID clients. However, further increasing the selection factor cannot improve the accuracy, e.g., $\alpha = 0.12$, because this will lead to too many clients with non-IID data selected in training. Fig. 5 and Fig. 6 also show that the accuracy of CSFedAvg decreases when the selection factor α is smaller than the proportion of clients with IID data γ , e.g., $\alpha = 0.03$. The reason is that with small selection factor, CSFedAvg is unable to select enough clients with IID data to improve the global model.



(a) Hybrid I setting on CIFAR-10 dataset



(b) Hybrid II setting on CIFAR-10 dataset

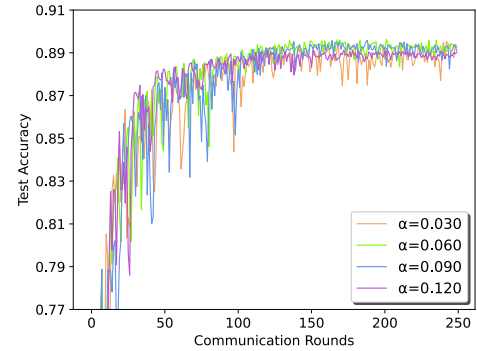
FIGURE 5. The impact of the selection factor on training accuracy of CSFedAvg based on CIFAR-10 dataset.

Besides accuracy, we also investigate the impact of α on the number of communication rounds. Similar to Section IV-C, we record the number of communication rounds required to reach a target accuracy based on both CIFAR-10 and Fashion MNIST datasets. As shown in Table 4 and Table 4, when the target accuracy is not high, e.g., 0.68 in Hybrid I setting or 0.9 in Hybrid II setting, larger value of α reduces the number of communication rounds required. However, when the target accuracy is high, e.g., 0.71 in Hybrid I setting or 0.905 in Hybrid II setting, increasing the value of α cannot always accelerate the accuracy convergence. The reason is that higher selection factor α can choose more clients in training, which however may increase the risk of selecting clients with non-IID data, and degrade the training performance.

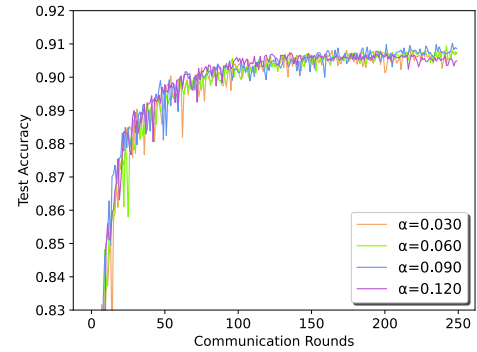
E. THE IMPACT OF THE NUMBER OF CLIENTS

To evaluate the number of clients on the accuracy achieved by the proposed CSFedAvg, we vary the total number of clients from 100 to 500.

In the first setting, we present the accuracy achieved after running 250 communication rounds. As shown in Fig. 7, in both algorithms, the accuracy increases with the number of clients in the system. Furthermore, compared with FedAvg and FedProx, the proposed CSFedAvg achieves higher accuracy on both CIFAR-10 and Fashion MNIST datasets. The reason is that CSFedAvg always uses weight divergence to choose the clients with lower non-IID degree data. It is also



(a) Hybrid I setting on Fashion MNIST dataset



(b) Hybrid II setting on Fashion MNIST dataset

FIGURE 6. The impact of the selection factor on accuracy of CSFedAvg, based on Fashion MNIST dataset.**TABLE 3.** Number of communication rounds to reach a target accuracy for CSFedAvg with different select factor on CIFAR-10 dataset.

selection factor	Hybrid I			Hybrid II		
	ToA@0.68	ToA@0.71	Accuracy	ToA@0.74	ToA@0.76	Accuracy
$\alpha = 0.03$	97	—	0.709	90	235	0.763
$\alpha = 0.06$	46	74	0.725	81	129	0.766
$\alpha = 0.09$	42	86	0.722	67	121	0.766
$\alpha = 0.12$	41	99	0.720	65	149	0.763

TABLE 4. Number of communication rounds to reach a target accuracy for CSFedAvg with different selection factor on Fashion MNIST dataset.

selection factor	Hybrid I			Hybrid II		
	ToA@0.88	ToA@0.89	Accuracy	ToA@0.900	ToA@0.905	Accuracy
$\alpha = 0.03$	70	118	0.891	67	119	0.908
$\alpha = 0.06$	52	102	0.896	65	114	0.909
$\alpha = 0.09$	48	102	0.896	63	102	0.910
$\alpha = 0.12$	44	112	0.892	61	106	0.908

observed that CSFedAvg on Hybrid II performs better than on Hybrid I setting. This is also reasonable, as the clients in Hybrid II hold data with lower non-IID degree.

In the second setting, we vary the number of communication rounds from 50 to 250, and study the accuracy achieved under different number of clients. Fig. 8 and Fig. 9

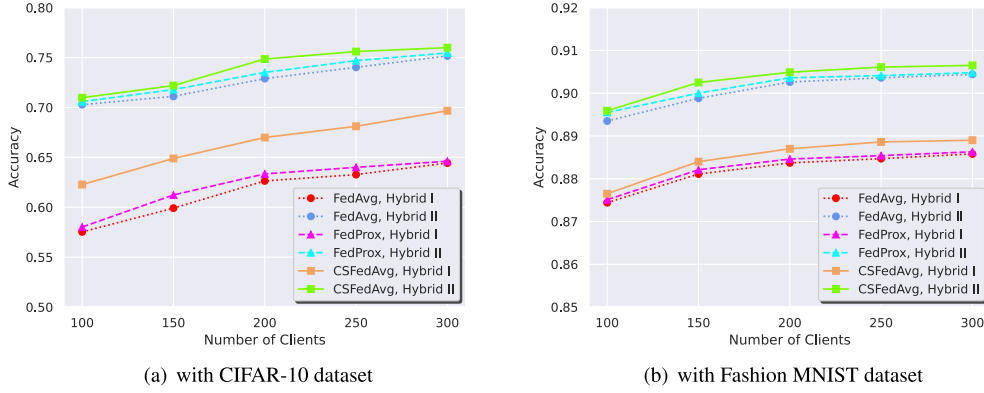


FIGURE 7. The impact of the number of clients based on CIFAR-10 and Fashion MNIST datasets.

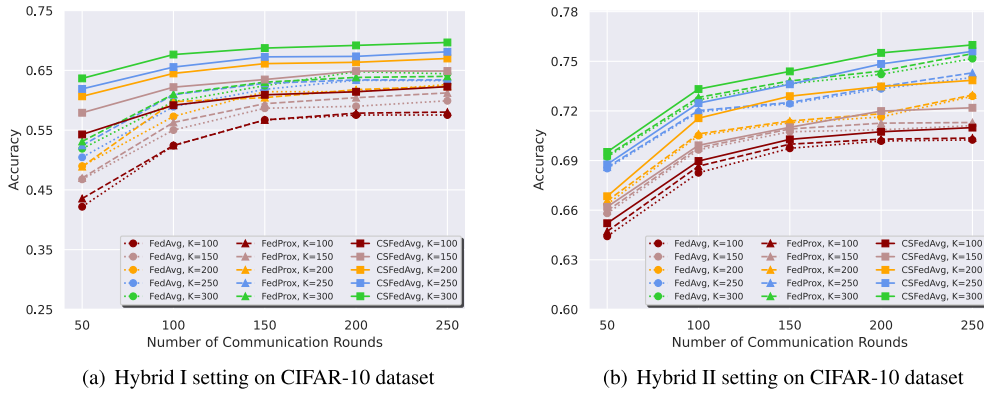


FIGURE 8. The impact of the number of clients and communication rounds, on CIFAR-10 dataset.

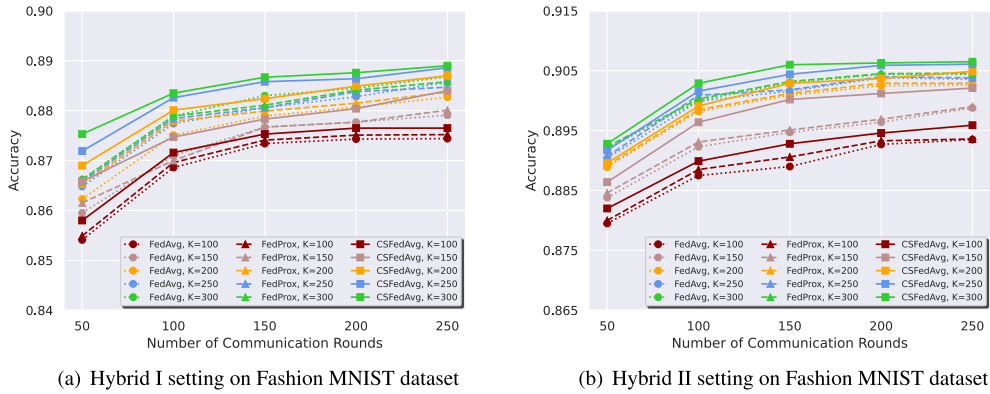


FIGURE 9. The impact of the number of clients and communication rounds, on Fashion MNIST dataset.

present the accuracy comparison under CIFAR-10 and Fashion MNIST datasets, respectively. It is observed that the proposed CSFedAvg accelerates the convergence in all the settings. Furthermore, similar to Fig. 7, the accuracy increases with the number of clients and the proposed CSFedAvg on Hybrid II performs better than on Hybrid I setting.

F. THE IMPACT OF GLOBAL IMBALANCED DISTRIBUTION

Besides local imbalance, global imbalance may also degrade the training accuracy of FL [24]. We now evaluate the impact of such global imbalance on the performance of the proposed CSFedAvg.

To simulate global imbalance, we set the data collection of all clients to an approximate power-law distribution [38]. Specifically, in Hybrid I setting, clients with non-IID data only own data belonging to one class. Without loss of generality, we use n_i to denote the number of clients that only own the i -th class data, which is defined as follows,

$$n_i = a \cdot i^{-b}, i \in [1, 10], \quad (10)$$

where

$$\sum_{i=1}^{10} n_i = K - \alpha K \quad (11)$$

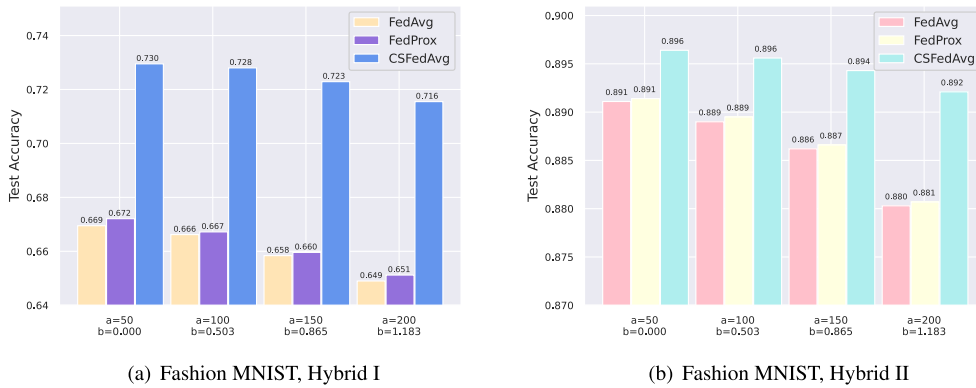


FIGURE 10. The impact of global imbalance based on CIFAR-10 and Fashion MNIST datasets.

In the above distribution, the parameters a and b control the degree of global imbalance, e.g., greater values of a and b reveal higher degree of global imbalance. Particularly, when $a = K/10$ and $b = 0$, there is only local imbalance (non-IID) and no global imbalance.

As shown in Fig. 10, we compare the accuracy achieved by three schemes, i.e., FedAvg, FedProx, and CSFedAvg, based on CIFAR-10 and Fashion MNIST datasets. In the simulation, we consider four global imbalance settings, where a is set to be 50, 100, 150, 200, respectively. It is observed that higher degree of global imbalance, i.e., higher value of a , indeed degrades the training accuracy of both algorithms. However, the accuracy of our algorithm degrades much slower than both FedAvg and FedProx, which confirms the robustness of CSFedAvg against the global imbalance. This is reasonable as the selected clients with IID data are in general with global-balanced data, which alleviates the adverse effect of global imbalance.

V. RELATED WORKS

In this section, we present some related works on FL in IoT or mobile computing.

FL is an emerging technique in machine learning community, and has received considerable attention in recent years. As a pioneering work, [8], [9] identify that there are mainly two kinds of data imbalance that may degrade the performance of FL: 1) local imbalance, i.e., non-IID data in clients; 2) global imbalance, i.e., the collection of data in all clients is class imbalance. Regarding the imbalance incurred by non-IID data, several researches have been made in the literature. For example, [18] analyzes the convergence of FedAvg on non-IID data and establishes a convergence rate for convex and smooth problems. Reference [19] shows that the non-IID CIFAR-10 dataset at clients degrades the accuracy of FedAvg. To address this issue, [19] allows the public available data to be distributed to clients such that the clients' data becomes IID. However, clients are in general reluctant to install unknown data on their devices for security reasons. The work in [22] proposes a Hybrid-FL, which allows a few number of clients to upload their data to the server, so as to form an approximately IID dataset. [23] constructs a sparse

ternary compression (STC) for FL, which requires both fewer gradient evaluations and communicated bits to converge to a target accuracy. Regarding the global imbalance of the data, [24] collects the data distribution/class of clients, and uses the mediators to reschedule the clients to improve the accuracy of FedAvg. The above works perform well in improving the accuracy incurred by data imbalance. However, most of them either introduce extra communication overhead [19], [22], i.e., requiring the clients to distribute or upload the data, or impose privacy threaten to clients, i.e., collecting data class/distribution from clients [24].

Due to the heterogeneity of clients in computation and communication capabilities, client selection and scheduling have also become critical issues in FL. For example, [22], [26] chooses the clients to participate in model training, according to their training and communication latency. Reference [27] studies incentive mechanisms for participating in training and client selection, so as to provide reliable federated learning. [28] proposes SAFA, a semi-asynchronous FL protocol, to address the problems such as low round efficiency and poor convergence rate in extreme weak communication conditions. The work in [29] studies three kinds of scheduling policies for FL to select clients, including Random Scheduling (RS), Round Robin (RR), and Proportional Fair (PF). Inspired by fair resource allocation in wireless networks, [30] proposes q -FFL and q -FedAvg to improve the fair accuracy distribution in FL. Reference [31] proposes a novel FL algorithm, named *FedProx*, which generalizes FedAvg and allows variable amounts of tasks to be performed locally across devices based on their available systems resources. The work in [32] proposes an orthogonal approach that identifies irrelevant updates made by clients and precludes them from being uploaded, so as to reduce the communication overhead. However, most of previous works considers to select the clients from the perspective of their heterogeneity in computation and communication capabilities, ignoring the imbalance of their data.

Compared with previous works, the CSFedAvg algorithm studied in this paper mainly has the following differences. Firstly, different from [19], the public dataset does not need to be distributed to the clients, which not only reduces

the communication overhead but also alleviates the possible security threaten of public data. Secondly, instead of randomly selecting the clients participating in FL, we use weight divergence to recognize the non-IID degrees of data in clients, based on which, the clients with low non-IID data will be chosen to train the models with higher frequency.

VI. CONCLUSION

This paper proposes a novel FL algorithm, CSFedAvg, to alleviate the accuracy degradation incurred by the non-IID data at clients. To recognize the clients with non-IID data, we first use weight divergence to differentiate the heterogeneity degrees of non-IID data at clients. Based on this, we then propose a weight-divergence aware client selection mechanism. Specifically, the clients with lower degree of non-IID data will be chosen to train the models with higher frequency. Finally, we conduct simulations to train deep neural networks based on two publicly-available datasets, CIFAR-10 and Fashion MNIST datasets. Simulation results show that the proposed CSFedAvg improves the training performance such as accuracy and convergence rate.

It is challenging if the auxiliary IID data is maliciously damaged. To address this issue, it is recommended that the auxiliary data should be updated/changed periodically or an appropriate malicious data detection algorithm should be designed at the server. As this is not the main concern of this paper, we would like to study this interesting problem in our future.

REFERENCES

- [1] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 854–864, Dec. 2016.
- [2] U. Ahsan and A. Bais, "A review on big data analysis and Internet of Things," in *Proc. IEEE 13th Int. Conf. Mobile Ad Hoc Sensor Syst. (MASS)*, Oct. 2016, pp. 325–330.
- [3] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1628–1656, 3rd Quart., 2017.
- [4] X. Chen, L. Pu, L. Gao, W. Wu, and D. Wu, "Exploiting massive D2D collaboration for energy-efficient mobile edge computing," *IEEE Wireless Commun.*, vol. 24, no. 4, pp. 64–71, Aug. 2017.
- [5] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge intelligence: Paving the last mile of artificial intelligence with edge computing," *Proc. IEEE*, vol. 107, no. 8, pp. 1738–1762, Aug. 2019.
- [6] J. Park, S. Samarakoon, M. Bennis, and M. Debbah, "Wireless network intelligence at the edge," *Proc. IEEE*, vol. 107, no. 11, pp. 2204–2239, Nov. 2019.
- [7] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "When edge meets learning: Adaptive control for resource-constrained distributed machine learning," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Apr. 2018, pp. 63–71.
- [8] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Arcas, "Communication-efficient learning of deep networks from decentralized data," *Proc. Mach. Learn. Res.*, vol. 54, pp. 1273–1282, Apr. 2017.
- [9] J. Konečný, B. McMahan, and D. Ramage, "Federated optimization: Distributed optimization beyond the datacenter," 2015, *arXiv:1511.03575*. [Online]. Available: <http://arxiv.org/abs/1511.03575>
- [10] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020.
- [11] Q. Wu, K. He, and X. Chen, "Personalized federated learning for intelligent IoT applications: A cloud-edge based framework," *IEEE Open J. Comput. Soc.*, vol. 1, pp. 35–44, 2020.
- [12] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "Adaptive federated learning in resource constrained edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1205–1221, Jun. 2019.
- [13] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 2031–2063, 3rd Quart., 2020.
- [14] X. Wang, Y. Han, C. Wang, Q. Zhao, X. Chen, and M. Chen, "In-edge AI: Intelligentizing mobile edge computing, caching and communication by federated learning," *IEEE Netw.*, vol. 33, no. 5, pp. 156–165, Sep. 2019.
- [15] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage, "Federated learning for mobile keyboard prediction," 2018, *arXiv:1811.03604*. [Online]. Available: <http://arxiv.org/abs/1811.03604>
- [16] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečný, S. Mazzocchi, H. Brendan McMahan, T. Van Overveldt, D. Petrou, D. Ramage, and J. Roselander, "Towards federated learning at scale: System design," 2019, *arXiv:1902.01046*. [Online]. Available: <http://arxiv.org/abs/1902.01046>
- [17] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, Feb. 2011.
- [18] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of FedAvg on non-IID data," 2019, *arXiv:1907.02189*. [Online]. Available: <http://arxiv.org/abs/1907.02189>
- [19] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-IID data," 2018, *arXiv:1806.00582*. [Online]. Available: <http://arxiv.org/abs/1806.00582>
- [20] K. Hsieh, A. Phanishayee, O. Mutlu, and P. B. Gibbons, "The non-IID data quagmire of decentralized machine learning," 2019, *arXiv:1910.00189*. [Online]. Available: <https://arxiv.org/abs/1910.00189>
- [21] H. Wang, Z. Kaplan, D. Niu, and B. Li, "Optimizing federated learning on non-IID data with reinforcement learning," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Jul. 2020, pp. 1698–1707.
- [22] N. Yoshida, T. Nishio, M. Morikura, K. Yamamoto, and R. Yonetani, "Hybrid-FL for wireless networks: Cooperative learning mechanism using non-IID data," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–7.
- [23] F. Sattler, S. Wiedemann, K.-R. Muller, and W. Samek, "Robust and communication-efficient federated learning from non-i.i.d. Data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3400–3413, Sep. 2020.
- [24] M. Duan, D. Liu, X. Chen, Y. Tan, J. Ren, L. Qiao, and L. Liang, "Astraea: Self-balancing federated learning for improving classification accuracy of mobile deep learning applications," in *Proc. IEEE 37th Int. Conf. Comput. Design (ICCD)*, Nov. 2019, pp. 246–254.
- [25] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 269–283, Jan. 2021, doi: [10.1109/TWC.2020.3024629](https://doi.org/10.1109/TWC.2020.3024629).
- [26] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–7.
- [27] J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang, "Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10700–10714, Dec. 2019.
- [28] W. Wu, L. He, W. Lin, R. Mao, C. Maple, and S. A. Jarvis, "SAFA: A semi-asynchronous protocol for fast federated learning with low overhead," *IEEE Trans. Comput.*, early access, May 14, 2020, doi: [10.1109/TC.2020.2994391](https://doi.org/10.1109/TC.2020.2994391).
- [29] H. H. Yang, Z. Liu, T. Q. S. Quek, and H. V. Poor, "Scheduling policies for federated learning in wireless networks," *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 317–333, Jan. 2020.
- [30] T. Li, M. Sanjabi, and V. Smith, "Fair resource allocation in federated learning," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–27.
- [31] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," 2018, *arXiv:1812.06127*. [Online]. Available: <http://arxiv.org/abs/1812.06127>
- [32] L. Wang, W. Wang, and B. Li, "CMFL: Mitigating communication overhead for federated learning," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2019, pp. 954–964.

- [33] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining Knowl. Discovery*, vol. 2, no. 2, pp. 121–167, 1998.
- [34] R. H. Myers and R. H. Myers, *Classical and Modern Regression With Applications*, vol. 2. Belmont, CA, USA: Duxbury Press, 1990.
- [35] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," M.S. thesis, Apr. 2009.
- [36] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," 2017, *arXiv:1708.07747*. [Online]. Available: <https://arxiv.org/abs/1708.07747>
- [37] T. Yang, G. Andrew, H. Eichner, H. Sun, W. Li, N. Kong, D. Ramage, and F. Beaufays, "Applied federated learning: Improving Google keyboard query suggestions," 2018, *arXiv:1812.02903*. [Online]. Available: <http://arxiv.org/abs/1812.02903>
- [38] A. Clauset, C. R. Shalizi, and M. E. J. Newman, "Power-law distributions in empirical data," *SIAM Rev.*, vol. 51, no. 4, pp. 661–703, Nov. 2009.



WENYU ZHANG received the B.S. degree from the School of Civil Engineering, Zhengzhou University, Zhengzhou, China, in 2014. He is currently pursuing the master's degree with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China. His research interests include mobile computing and edge computing.



South China University of Technology, Guangzhou, China. Her research interests include mobile computing, wireless networks, and algorithm design and optimization.



PAN ZHOU (Senior Member, IEEE) received the B.S. degree (Hons.) from the Advanced Class, Huazhong University of Science and Technology (HUST), Wuhan, China, in 2006, the M.S. degree from the Department of Electronics and Information Engineering, HUST, in 2008, and the Ph.D. degree from the School of Electrical and Computer Engineering, Georgia Institute of Technology (Georgia Tech), Atlanta, USA, in 2011. He was a Senior Technical Member of Oracle America Inc., from 2011 to 2013, where worked on Hadoop and distributed storage system for big data analytics at Oracle Cloud Platform. He is currently a Full Professor and a Ph.D. Advisor with the Hubei Engineering Research Center on Big Data Security, School of Cyber Science and Engineering, HUST. His current research interests include security and privacy, big data analytics, machine learning, and information networks. He received the Merit Research Award of HUST during his M.S. degree and the Rising Star in Science and Technology of HUST, in 2017. He is also an Associate Editor of IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING.



WEIWEI WU received the B.S. degree from the School of Computer Science and Engineering, South China University of Technology, China, and the joint Ph.D. degree from the City University of Hong Kong and the University of Science and Technology of China, China, in 2011. He is currently a Professor with the School of Computer Science and Engineering, Southeast University, Nanjing, China. His research interests include optimization in wireless networks, algorithm design and analysis, scheduling, cloud computing, and game theory.



XINLIN ZHANG (Member, IEEE) received the B.E. degree from the School of Software, Sun Yat-sen University, in 2010, and the Ph.D. degree from the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, in 2014. He is currently an Associate Professor with the South China University of Technology. His research interests include mobile crowdsourcing, wireless ad-hoc/sensor networks, and pervasive computing. He is a member of the ACM.

...