Client Scheduling for Multiserver Federated Learning in Industrial IoT With Unreliable Communications

Haitao Zhao, *Senior Member, IEEE*, Yuhao Tan, Kun Guo, *Member, IEEE*, Wenchao Xia, *Member, IEEE*, Bo Xu, *Member, IEEE*, and Tony Q. S. Quek, *Fellow, IEEE*

***Abstract*—The Industrial Internet of Things (IIoT) is emerging as a promising technology that can accelerate the application of industrial intelligence to smart factories. Because of the sensitive nature of user data, federated learning (FL) which performs distributed machine learning while preserving data privacy, is leveraged to meet the accuracy and privacy requirements of IIoT end devices/clients. However, the unreliable communications in IIoT may result in possible single-point failures in the typical single-server FL framework, thereby negatively affecting the training efficiency. In this article, we study on the client scheduling problem in a multiserver FL framework for the communication reliability and training efficiency improvement. Specifically, we focus on a semi-decentralized FL (SD-FL) framework, where edge servers and clients collaborate to train a shared global model through unreliable intracluster model aggregation and intercluster model consensus because of the model transmission error in client–server and server–server communication. Then, a client–server association optimization problem is formulated, with the objective of minimizing the global training loss. Resorting to the convergence analysis of SD-FL, the original problem is simplified and transformed into an integer nonlinear programming problem to guide us to design a high-efficiency client scheduling scheme. Finally, experimental results show that the proposed scheme significantly outperforms the baselines in terms of the test accuracy and training loss.**

***Index Terms*—Client scheduling, edge association, federated learning (FL), Industrial Internet of Things (IIoT), unreliable communication.**

Manuscript received 16 February 2023; revised 30 August 2023, 9 November 2023, and 3 January 2024; accepted 8 January 2024. Date of publication 16 January 2024; date of current version 25 April 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62201285, Grant 62371250, Grant 92367302, and Grant 62301222; in part by the Natural Science Foundation on Frontier Leading Technology Basic Research Project of Jiangsu under Grant BK20212001; in part by the Jiangsu Natural Science Foundation for Distinguished Young Scholars under Grant BK20220054; in part by the National Research Foundation, Singapore and Infocomm Media Development Authority under its Future Communications Research and Development Programme; in part by the China Postdoctoral Science Foundation under Grant 2022M722669; in part by the open research fund of National Mobile Communications Research Laboratory, Southeast University, under Grant 2023D04; and in part by the Natural Science Foundation of Chongqing, China, under Grant CSTB2022NSCQ-MSX0375. *(Corresponding authors: Wenchao Xia; Tony Q. S. Quek.)*

Haitao Zhao, Yuhao Tan, Wenchao Xia, and Bo Xu are with the Jiangsu Key Laboratory of Wireless Communications, Nanjing University of Posts and Telecommunications, Nanjing 210003, China (e-mail: zhaoht@njupt.edu.c; 1221014419@njupt.edu.cn; xiawenchao@njupt.edu.cn; xubo@njupt.edu.cn).

Kun Guo is with the National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China, also with the Chongqing Key Laboratory of Precision Optics, Chongqing Institute of East China Normal University, Chongqing 401120, China, and also with the School of Communications and Electronics Engineering, East China Normal University, Shanghai 200241, China (e-mail: kguo@cee.ecnu.edu.cn).

Tony Q. S. Quek is with the Singapore University of Technology and Design, Singapore 487372, and also with the Department of Electronic Engineering, Kyung Hee University, Yongin 17104, South Korea (e-mail: tonyquek@sutd.edu.sg).

Digital Object Identifier 10.1109/JIOT.2024.3354914

I. Introduction

THROUGH the deep integration of communication technology and industrial manufacturing, the Industrial Internet of Things (IIoT) has become one of the most promising and profitable technologies in the Internet of things system [1]. Particularly, by comprehensively leveraging the intelligent analysis technology and remote control technology, IIoT is able to optimize the operation and maintenance of industrial facilities, and to bring intelligence to industrial manufacturing [2]. To facilitate the intelligence in IIoT, the need for application of edge artificial intelligence (AI) is urgent [3]. In this way, large number of IIoT end devices can collaboratively train AI models through the edge network, to provide smart applications and services that satisfy ultrareliable and low-latency requirements [4].

In IIoT, the local data of end devices usually contain sensitive or private information [5], which makes the traditional edge AI training architecture infeasible. This is because, the traditional edge AI model training architecture is centralized, where the end device/clients need to upload their local data to the edge server to train the AI model. During this process, malicious attacks from the network may pose a threat to the security of user data. At the same time, the size of data uploaded by clients can be very large, which will result in a large network load. Therefore, a novel distributed machine learning framework, i.e., federated learning (FL) [6], is introduced in IIoT to cope with security and privacy issues while reducing communication costs [7]. The advantages of FL profit from the fact that the clients jointly train a global model by uploading the parameters of their local models instead of local data [8]. In the typical FL framework, all the clients coordinate with one edge server, thereby resulting in high-communication overhead and possible single-point failures in IIoT with massive end devices [9].

In this article, we focus on multiserver FL in IIoT to improve the reliability during model training process. As shown in Fig. 1, we focus on a SD-FL framework [10] which

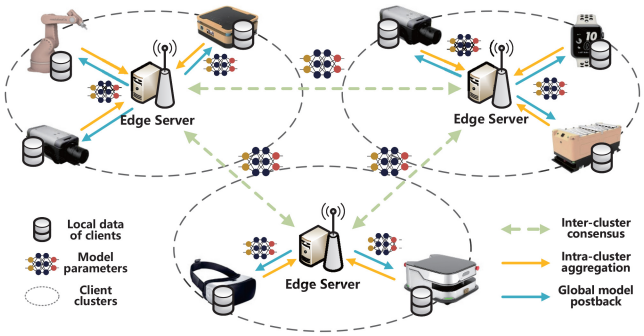


Fig. 1. Illustration of SD-FL framework in multiserver IIoT scenario.

involves multiple edge servers and clients training a shared global model in collaboration through the aggregation of local model by the servers (i.e., intracluster model aggregation), and the exchange and aggregation of model among multiple servers (i.e., intercluster model consensus). Due to the collaboration among multiple edge servers, SD-FL can effectively reduce the impact of single-point failure on model training and improve the scalability of the system, hence is more applicable for IIoT deployment. However, the work in [10] ignored practical issues in communications, such as client–server and server–server communication unreliability. In the considered SD-FL scenario, model data are transmitted through client–server and server–server wireless communication links for the intracluster model aggregation and intercluster model consensus, respectively. The noise and interference in the wireless links may induce erroneous data transmission, therefore affecting the model training efficiency. Hence, the unreliable communications in IIoT-based SD-FL framework has to be studied.

There have been efforts in solving the problem of unreliable communications in IIoT systems. Shi et al. [11] leveraged cloud-fog integrated computing to alleviate the impact of unreliable communication links in IIoT scenarios, which brings computing resources to the edge of network and reduces the average service latency. While, Lu et al. [12] introduced digital twin wireless networks to mitigate the performance degradation caused by failures of FL model transmission. Wang et al. [13] defined reliable/unreliable indicators of the sensing data, and leveraged deep neural network to predict the trust value of data, which guided an optimal data selection algorithm that maximized the control performance of the IIoT system. For communication unreliability caused by attacks and lack of trust among participating entities in IIoT networks, Kumar et al. [14] proposed a data delivering framework for decentralized data processing and learning by integrating blockchain and deep learning. To tackle the problem of multihop wireless link unreliability in IIoT edge computing networks, Xu et al. [15] constructed a link model which revealed the relationship between link reliability and transmission delay.

The aforementioned works explored various methods to overcome communication unreliability caused by different reasons in the IIoT scenarios. In the considered SD-FL framework, multiple edge servers are deployed among a number of IIoT end devices, training a global model in a collaborative manner. For our considered SD-FL framework, the unreliability of communication is caused mainly by the failure of data transmission through wireless links. Hence, client scheduling which can alleviate the impact of transmission failure by simply managing the participation and association of client devices with multiple edge servers, is leveraged in our work. Considering both unreliable client–server and server–server communications, we study the client scheduling problem in IIoT-based multiserver FL framework where client devices are appropriately associated with edge servers, so as to alleviate the unreliability in wireless links and improve the training efficiency of the SD-FL. The main contributions of this article can be summarized as follows.

1) We model the intracluster model aggregation and the intercluster model consensus with the model (encapsulated as a packet) transmission error, and formulate the optimization problem which aims at minimizing the global training loss by means of client scheduling among multiple edge servers.
2) Through the convergence analysis, the expected impacts of packet error rate (PER) and client–server association on the global model convergence is derived. On this basis, the original optimization problem is simplified and transformed into an integer nonlinear programming problem, to guide a high-efficiency client scheduling scheme design.
3) With MNIST and CIFAR-10 data sets, and non-i.i.d. data distributions, experiment results demonstrate the performance improvement of the proposed client scheduling scheme in terms of the reliability of client participation and training efficiency.

The remainder of this work is organized as follows. Section II presents the related works. Section III introduces the system model and formulates the optimization problem. Section IV provides the convergence analysis and client scheduling algorithm. Section V presents the simulation results, followed by the conclusions in Section VI.

II. Related Works

There have been extensive studies on client scheduling in different FL frameworks. In the sequel, we thus summarize the related works in typical single-server FL framework and multiserver FL framework, respectively.

A. Client Scheduling in Single-Server FL

For the typical FL framework with a single edge server and multiple clients, Xia et al. [16] proposed multiarmed bandit (MAB) based client scheduling algorithms aiming at minimizing the training latency, with the consideration of both independent and identically distributed (i.i.d.) and non-i.i.d. local data sets among clients. To alleviate the communication overhead, Wang et al. [17] identified irrelevant model updates uploaded by the clients in each round which are then precluded from the model aggregation. Stem from the conclusion that prejudicially selecting clients with higher local training loss accelerates the convergence of global training

loss, a communication and computation efficient client selection method was proposed in [18] by striking a balance among the convergence acceleration, client selection bias, and communication/computation overhead. Except for the client scheduling, resource allocation provides another dimension to improve the AI model training. For instance, Shi et al. [19] studied the joint client scheduling and resource allocation problem of latency constrained wireless FL, which aims to maximize the global model accuracy under given total training time constraint. Ren et al. [20] proposed to scale the importance of a local update according to the divergence of gradient, based on which a joint client scheduling and resource allocation scheme was devised to work out an optimal tradeoff between the reliability of wireless link and the importance of local updates. In [21], the fairness guaranteed client selection problem was modeled as a Lyapunov optimization problem, then the model delivery time between the clients and the server was estimated through a C^2MAB-based method. A bipartite matching (BM) based method was proposed in [22] to settle the joint client scheduling and resource block (RB) allocation problem with low-computational complexity. Different from the above works, Yoshida et al. [23] and Xu et al. [24] assumed that the computation and communication resources of clients are not known a priori, and adopted an MAB based client scheduling scheme for the reduction of the training latency. Note that the above works only considered client scheduling in single-server FL, hence the methods are not applicable to our concern multiserver FL scenarios in this work.

B. Client Scheduling in Multiserver FL

Many works also focused on the client scheduling problem in multiserver FL framework, e.g., the hierarchical federated edge learning (H-FEEL) framework. The H-FEEL framework combines the advantages of cloud server equipped with strong computational power and edge servers which can significantly cut down the training latency and energy consumption. In H-FEEL, at least one edge server are deployed hierarchically with the centralized cloud server to collaboratively support the AI model training. In [25], the client scheduling and resource allocation problem of H-FEEL framework was studied, where the optimal resource allocation policy under the condition of a single edge server was first derived, followed by the optimization of the client-edge association policy among multiple edge servers through an iterative reduction process of global cost. Different from [25], [26] jointly considered the features of the wireless channel and the interval of edge model aggregation. Considering the non-i.i.d. distribution of local data sets, Mhaisen et al. [27] optimized the client-edge association to uniform the edge-level data distributions, thereby improving the learning efficiency of the FL system. In [28], a deep reinforcement learning (DRL) based method was proposed to optimize the edge association under computing capacity and transmission power constraints in a NOMA-enabled H-FEEL architecture.

There are also works focusing on the energy-efficient edge association design in H-FEEL. Reference [29] took into consideration both the local data distribution and the model transmission latency. Then, the edge association and resource allocation are jointly optimized to minimize the communication energy cost. In [30], the importance of local model updates were considered, and the edge association was optimized to minimize the training latency under a long-term energy cost budget. Different from the above works, by allowing each FL model to select their own participating clients, Wei et al. [31] studied multimodel H-FEEL which aims to minimize the total learning cost to better balance the training load and manage the competition among models. To further exploit the interactions among clients, Lim et al. [32] focused on a dynamic edge association problem where noncooperative game players (i.e., cloud server, edge server, and clients) maximize their profits and make association decisions in self-organizing H-FEEL.

However, the aforementioned works all studied edge association in H-FEEL framework, where server–server communication is assumed to be built reliably with the aid of cloud server. In this work, the considered SD-FL framework allows the direct communication between two interconnected edge servers through unreliable wireless links. The unreliable client–server and server–server communication in SD-FL has a significant impact on the edge association (or called as client scheduling) and training efficiency, yet which still remains an open problem.

III. System Model and Problem Formulation

In this section, we consider the SD-FL framework [10] in IIoT as shown in Fig. 1. Considering the unreliable communications, we aim to improve the training efficiency by the scheduling of clients among multiple edge servers. In the sequel, we elaborate on the preliminaries of SD-FL framework and system model, respectively, and then formulate the optimization problem.

A. SD-FL Preliminaries

In this article, we consider a set $\mathcal{S} = \{1, 2, \ldots, S\}$ of S edge servers and a set $\mathcal{U} = \{1, 2, \ldots, U\}$ of U client devices for T-round SD-FL training. In each round, the following procedures are included.

1) The clients first perform T_1 rounds of local training with their local data sets and then upload the latest updated local models to the associated edge server for intracluster model aggregation;
2) The models received at the edge servers are aggregated according to predefined principles, then the models aggregated at the edge servers are broadcasted to the associated clients for local model update;
3) After T_2 rounds of intracluster model aggregation, each edge server broadcasts their most recent aggregated models to the adjacent servers for intercluster model consensus where the models received at each edge server are aggregated according to predefined principles;
4) After the intercluster model consensus, the models aggregated at the edge servers are also broadcasted to the associated clients for the next training round.

Note that setting $T_1 = T_2 = 1$ can reduce the training rounds required for the convergence [10]. Thus, in this article we assume that the clients perform one round of local model update, followed by the intracluster model aggregation and the intercluster model consensus.

B. System Model

During each training round, the clients only communicate with the associated edge server. Without loss of generality, we assume that each client can associate with up to one edge server at the same time. We denote the association between client $i \in \mathcal{U}$ and edge server $s \in \mathcal{S}$ as $a_{i,s} \in \{0, 1\}$, where $a_{i,s} = 1$ indicates that client i is associated with edge server s, while $a_{i,s} = 0$ means that client i and edge server s are not associated. Then, the client–server association matrix can be denoted as $\boldsymbol{a}^{U\times S}$, and the set of clients associated with server s is denoted as $\mathcal{U}_s = \{i \in \mathcal{U} | a_{i,s} = 1\}$.

In the SD-FL framework, the clients upload their local model parameters to the edge servers, instead of transmitting the local data sets directly. We assume that the frequency resources owned by different edge servers are not overlapped. For the local model transmission (i.e., the uplink transmission), the orthogonal frequency division multiple access (OFDMA) is employed in one edge server where each client device associated with the edge server will be allocated with one RB. On the allocated RB, the local model of client is assumed to be encapsulated and transmitted as a single data packet. We utilize the cyclic redundancy check (CRC) mechanism to inspect whether there are packet errors at the receiving end due to wireless factors. Once packet error is detected, the whole packet will be discarded meaning that the local model in this packet will not be used for any model aggregation.

In this article, we use the quasi static Rayleigh fading to characterize the wireless channels. Referring to [22], we then use the PER to denote the probability that the transmitted packet contains errors, as is given in [33] as follows:

$$q_i(a_{i,s}) = \mathbb{E}_{h_{i,s}}\left(1 - \exp\left(-\frac{\tau B^U N_0}{P_i h_{i,s}}\right)\right) \tag{1}$$

where $q_i(a_{i,s})$ expresses the PER of client i transmitting its local model parameters to the associated edge server under the client–server association indicator of $a_{i,s}$. P_i denotes the transmit power of client i. $h_{i,s}$ is the channel gain between client i and server s, including both the large-scale and small-scale fading gain. B^U is the uplink bandwidth of each RB, and N_0 is the noise power spectral density. τ is the threshold parameter [33], which is correlated with the system settings. Specifically, τ can be given by the inverse coding gain of client–server data transmission.

Further, we introduce the client–server model transmitting indicator denoted as $C_i(a_{i,s}) \in \{0, 1\}$, where $C_i(a_{i,s}) = 1$ indicates that the packet (i.e., local model) of client i is successfully transmitted under the client–server association indicator of $a_{i,s}$, otherwise $C_i(a_{i,s}) = 0$. The indicator $C_i(a_{i,s})$ satisfies the following requirements:

$$C_i(a_{i,s}) = \begin{cases} 1, & \text{with prob. } 1 - q_i(a_{i,s}) \\ 0, & \text{with prob. } q_i(a_{i,s}). \end{cases} \tag{2}$$

Without loss of generality, we consider interference-free server–server communication, which is orthogonal to the client–server communication. Hence, during the intercluster model consensus process, the PER of server s sending the model to server j is similar to (1) and is given by

$$p_{s,j} = \mathbb{E}_{h_{s,j}}\left(1 - \exp\left(-\frac{\tau' B^S N_0}{P_s h_{s,j}}\right)\right) \tag{3}$$

where P_s is the transmit power of server s. $h_{s,j}$ is the channel gain between server s and server j, B^S is the bandwidth of server–server communication and N_0 is the noise power spectral density, and τ' is another threshold parameter, given by the inverse coding gain of server–server data transmission. Then, the server–server model transmitting indicator is given as follows likewise:

$$C_{s,j}^{(S)} = \begin{cases} 1, & \text{with prob. } 1 - p_{s,j} \\ 0, & \text{with prob. } p_{s,j}. \end{cases} \tag{4}$$

Based on the above settings, the intracluster model aggregation at server s in training round t can be expressed by

$$\boldsymbol{w}_{s,t}^{(S)} = \frac{\sum_{i\in\mathcal{U}} K_i \boldsymbol{w}_{i,t} a_{i,s} C_i(a_{i,s})}{\sum_{i\in\mathcal{U}} K_i a_{i,s} C_i(a_{i,s})} \tag{5}$$

where $\boldsymbol{w}_{s,t}^{(S)}$ denotes the model aggregated at the server s during the model aggregation in round t, K_i is the data size of the local data set of client i, and $\boldsymbol{w}_{i,t}$ is the local model of client i in round t. Also, the intercluster model consensus at server s in training round t can be expressed by

$$\boldsymbol{w}_{s,t}^{(C)} = \frac{\boldsymbol{w}_{s,t}^{(S)} + \sum_{j\in\mathcal{S}_s} \boldsymbol{w}_{j,t}^{(S)} C_{s,j}^{(S)}}{1 + \sum_{j\in\mathcal{S}_s} C_{s,j}^{(S)}} \tag{6}$$

where $\boldsymbol{w}_{s,t}^{(C)}$ denotes the model aggregated at the server s during the model consensus in round t. $\mathcal{S}_s$ is the set of servers connected with the server s in one-hop. As has been described before, after the intercluster model consensus process in round t, edge server s sends back the aggregated model to the associated clients. Note that for the broadcast of global model, since the transmission power of the server is relatively high, the downlink PER is negligible. Meanwhile, since the edge server has less energy constraints than the client device, packet retransmission can be adopted for the broadcast of global model, hence the downlink packet error is not considered. Then, the clients associated with edge server s update their local models as $\boldsymbol{w}_{i,t} = \boldsymbol{w}_{s,t}^{(C)}$, $i \in \mathcal{U}_s$. For reasons of conciseness, we define $\boldsymbol{g}_{s,t}(\boldsymbol{a}) \triangleq \boldsymbol{w}_{s,t}^{(C)}$, and then the updated local model can be expressed as $\boldsymbol{w}_{i,t} = \boldsymbol{g}_{s,t}(\boldsymbol{a})$, $i \in \mathcal{U}_s$.

Further, at the beginning of training round $t + 1$, each client conducts local model training based on the updated local model and the local data sets via the following standard gradient descent:

$$\boldsymbol{w}_{i,t+1} = \boldsymbol{g}_{s,t}(\boldsymbol{a}) - \frac{\lambda}{K_i}\sum_{k=1}^{K_i} \nabla f\left(\boldsymbol{g}_{s,t}(\boldsymbol{a}), \boldsymbol{x}_{ik}, \boldsymbol{y}_{ik}\right), i \in \mathcal{U}_s \tag{7}$$

with $\boldsymbol{w}_{i,t+1}$ being the local model of client i in training round $t + 1$. λ is the learning rate and $\nabla f(\boldsymbol{g}_{s,t}(\boldsymbol{a}), \boldsymbol{x}_{ik}, \boldsymbol{y}_{ik})$ is the gradient of local training loss function, with respect to model

Algorithm 1: SD-FL

Initialize local model $\boldsymbol{w}_{i,0} = \boldsymbol{w}_0\ \forall i \in \mathcal{U}$;
for $t = 1, 2, ..., T$ **do**
  $\tau = 0$;
  repeat
    for $\forall i \in \mathcal{U}$ **do**
      Update local model $\boldsymbol{w}_{i,t}$ for T_1 times according to (7);
      Transmit the updated model to the correlated server;
    end
    for $\forall s \in \mathcal{S}$ **do**
      Receive $\boldsymbol{w}_{i,t}$ transmitted from clients $\forall i \in \mathcal{U}_s$;
      Aggregate the received models according to (5);
      Broadcast the aggregated model $\boldsymbol{w}_{s,t}^{(S)}$ to clients $\forall i \in \mathcal{U}_s$;
    end
    $\tau = \tau + 1$;
  until $\tau = T_2$;
  for $\forall s \in \mathcal{S}$ **do**
    Broadcast the latest updated model $\boldsymbol{w}_{s,t}^{(S)}$ to servers $\forall j \in \mathcal{S}_s$;
    Receive the models from servers $\forall j \in \mathcal{S}_s$;
    Aggregate the received models according to (6);
    Broadcast the aggregated model $\boldsymbol{w}_{s,t}^{(C)}$ to clients $\forall i \in \mathcal{U}_s$;
  end
end

$\boldsymbol{g}_{s,t}(\boldsymbol{a})$. Besides, $\boldsymbol{x}_{ik}$ and $\boldsymbol{y}_{ik}$ are the input and output of the training data sets of client i, respectively. Note that at timeslot $t = 0$, the local models of all clients are randomly initialized with the same initial global model $w_{i,0}$. With the training and model aggregation process introduced, details of the SD-FL training process are shown in Algorithm 1.

C. Problem Formulation

The purpose of client scheduling in the considered scenario is to find the optimal client–server association matrix $\boldsymbol{a}$ that minimizes the global training loss function given by

$$F(\boldsymbol{g}) = \frac{1}{K}\sum_{s=1}^{S}\sum_{i\in\mathcal{U}_s} F_i(\boldsymbol{g}_s) \tag{8}$$

where K is the total data size owned by all the clients in the system. $\boldsymbol{g}_s$ is short for $\boldsymbol{g}_{s,t}(\boldsymbol{a})\ \forall t \in T$. $\boldsymbol{g}$ is the equivalent global model. $F_i(\boldsymbol{g}_s)$ represents the local training loss of client i which is given by

$$F_i(\boldsymbol{g}_s) = \sum_{k=1}^{K_i} f(\boldsymbol{g}_s, \boldsymbol{x}_{ik}, \boldsymbol{y}_{ik}),\ \ i \in \mathcal{U}_s. \tag{9}$$

Hence, the client–server association matrix can be optimized by solving the following global training loss minimization problem:

$$\min_{\boldsymbol{a}} \frac{1}{K}\sum_{s=1}^{S}\sum_{i\in\mathcal{U}_s}\sum_{k=1}^{K_i} f(\boldsymbol{g}_s, \boldsymbol{x}_{ik}, \boldsymbol{y}_{ik}) \tag{10a}$$

$$\text{s.t. } a_{i,s} \in \{0, 1\}\ \forall i \in \mathcal{U}\ \forall s \in \mathcal{S} \tag{10b}$$

$$\sum_{\forall s\in\mathcal{S}} a_{i,s} \le 1,\ \ \forall i \in \mathcal{U} \tag{10c}$$

$$\sum_{\forall i\in\mathcal{U}} a_{i,s} \le U_{\max},\ \ \forall s \in \mathcal{S} \tag{10d}$$

where (10a) restricts the client–server association indicator, (10b) implies that each client can associate with no more than one edge server at the same time, and (10c) restricts the maximum number of clients (i.e., $U_{\max}$) that one edge server can associate with. Problem (10) is hard to solve due to the implicit relationship between the objective function and the optimization variable. In the subsequent section, we thus conduct the convergence analysis of the FL system to make such a relationship explicit and use the analytical results for high-efficient client scheduling algorithm design.

IV. Convergence Analysis and Algorithm Design

In this section, we analyze the impact of PER in client–server and server–server communication on the convergence performance of the considered SD-FL, and then recast the optimization problem (10) to solve it with high efficiency. Generally speaking, the convergence rate of an FL system increases when more clients successfully participates in each training iteration [34]. Considering the unreliability of both client–server and server–server communications, it is necessary to further study the effects of the above factors on the convergence rate of the multiserver FL system, so as to guide an optimal client–server association strategy.

In the convergence analysis, the mathematical expectation of the difference of the global training loss with regard to global model $\boldsymbol{g}_{t+1}$ and the optimal global model $\boldsymbol{g}^*$ is derived, which is further proved to be upper bounded when t is toward to infinity. On this basis, the original optimization problem (10) is simplified and further transformed into an integer nonlinear programming problem. Based on the conclusions of the convergence analysis, a high-efficiency client scheduling scheme is proposed.

A. Convergence Analysis

According to the intracluster model aggregation and the intercluster model consensus described in (5) and (6), the aggregated model on server s in round $t+1$ after model consensus is given by

$$\boldsymbol{g}_{s,t+1} = \frac{\boldsymbol{g}_{s,t} + \sum_{j\in\mathcal{S}_s} C_{s,j}^{(S)} \boldsymbol{g}_{j,t}}{1 + \sum_{j\in\mathcal{S}_s} C_{s,j}^{(S)}} - \frac{\lambda}{1 + \sum_{j\in\mathcal{S}_s} C_{s,j}^{(S)}} \left(\frac{\sum_{i\in\mathcal{U}} a_{i,s} C_i(a_{i,s}) \nabla F_i(\boldsymbol{g}_{s,t})}{\sum_{i\in\mathcal{U}} K_i a_{i,s} C_i(a_{i,s})} + \sum_{j\in\mathcal{S}_s} \frac{\sum_{i\in\mathcal{U}} a_{i,j} C_i(a_{i,j}) \nabla F_i(\boldsymbol{g}_{j,t})}{\sum_{i\in\mathcal{U}} K_i a_{i,j} C_i(a_{i,j})} \right). \tag{11}$$

Based on the conclusions given in [10], during the training process, the difference between the models aggregated at the servers tend to narrow. Hence, we assume that in (11), $\boldsymbol{g}_{s,t}$ and

$\boldsymbol{g}_{j,t}$ satisfy $\boldsymbol{g}_{s,t} = \boldsymbol{g}_{j,t}\ \forall s, j \in \mathcal{S}$ for a large t. Then, we have $\left[(\boldsymbol{g}_{s,t} + \sum_{j\in\mathcal{S}_s} C_{s,j}^{(S)} \boldsymbol{g}_{j,t})/(1 + \sum_{j\in\mathcal{S}_s} C_{s,j}^{(S)})\right] = \boldsymbol{g}_{s,t}$ and (11) can be rewritten as

$$\boldsymbol{g}_{s,t+1} = \boldsymbol{g}_{s,t} - \lambda\left(\nabla F(\boldsymbol{g}_{s,t}) - \boldsymbol{o}_s\right) \tag{12}$$

where

$$\boldsymbol{o}_s = -\frac{1}{1 + \sum_{j\in\mathcal{S}_s} C_{s,j}^{(S)}}\left(\frac{\sum_{i\in\mathcal{U}} a_{i,s} C_i(a_{i,s}) \nabla F_i(\boldsymbol{g}_{s,t})}{\sum_{i\in\mathcal{U}} K_i a_{i,s} C_i(a_{i,s})} + \sum_{j\in\mathcal{S}_s} \frac{\sum_{i\in\mathcal{U}} a_{i,j} C_i(a_{i,j}) \nabla F_i(\boldsymbol{g}_{j,t})}{\sum_{i\in\mathcal{U}} K_i a_{i,j} C_i(a_{i,j})}\right) + \nabla F(\boldsymbol{g}_{s,t}) \tag{13}$$

and $\nabla F(\boldsymbol{g}_{s,t}) = (1/S) \sum_{j\in\mathcal{S}} ([\sum_{i\in\mathcal{S}} a_{i,j} \nabla F_i(\boldsymbol{g}_{j,t})]/[\sum_{i\in\mathcal{S}} K_i a_{i,j}])$.

Furthermore, we consider that the local models of all clients converge to an optimal global model $\boldsymbol{g}^*$ and deduce Theorem 1, under the following assumptions [35], [36].

1) *Assumption 1:* The gradient of $F(\boldsymbol{g})$ in (8), i.e., $\nabla F(\boldsymbol{g})$, is assumed to be uniformly Lipschitz continuous with regard to $\boldsymbol{g}$. Based on this assumption, the following inequation holds:
$$\left\|\nabla F(\boldsymbol{g}_{t+1}) - \nabla F(\boldsymbol{g}_t)\right\| \le L\left\|\boldsymbol{g}_{t+1} - \boldsymbol{g}_t\right\| \tag{14}$$
where $\|\cdot\|$ represents the norm calculation and L is a positive parameter.
2) *Assumption 2:* The global loss function $F(\boldsymbol{g})$ is assumed to be strongly convex, hence we can derive
$$F(\boldsymbol{g}_{t+1}) \ge F(\boldsymbol{g}_t) + (\boldsymbol{g}_{t+1} - \boldsymbol{g}_t)^{\mathrm{T}} \nabla F(\boldsymbol{g}_t) + \frac{\mu}{2}\left\|\boldsymbol{g}_{t+1} - \boldsymbol{g}_t\right\|^2 \tag{15}$$
where μ is a positive parameter.
3) *Assumption 3:* The global loss function $F(\boldsymbol{g})$ is assumed to be twice-continuously differentiable, i.e., $\mu \boldsymbol{I} \preceq \nabla^2 F(\boldsymbol{g}) \preceq L\boldsymbol{I}$.
4) *Assumption 4:* We assume that $\|\nabla f(\boldsymbol{g}_t, \boldsymbol{x}_{ik}, \boldsymbol{y}_{ik})\|^2 \le \omega_1 + \omega_2 \|\nabla F(\boldsymbol{g}_t)\|^2$, where ω_1 and ω_2 are nonnegatives.

Theorem 1: With the learning rate set as $\lambda = (1/L)$, we can derive the upper bound of $\mathbb{E}(F(\boldsymbol{g}_{t+1}) - F(\boldsymbol{g}^*))$ as follows:

$$\mathbb{E}\left(F(\boldsymbol{g}_{t+1}) - F(\boldsymbol{g}^*)\right) \le A^t \mathbb{E}\left(F(\boldsymbol{g}_0) - F(\boldsymbol{g}^*)\right) + \frac{2\omega_1 B}{L}\frac{1 - A^t}{1 - A} \tag{16}$$

where we have

$$A = 1 - \frac{\mu}{L} + \frac{4\omega_2 \mu B}{L} \tag{17}$$

and

$$B = \sum_{s\in\mathcal{S}} \sum_{m\in\mathcal{S}} \frac{1}{K_m^{(S)}}\left(\sum_{i\in\mathcal{U}} K_i a_{i,m}\left(\frac{1 - p_{s,m}}{1 + \sum_{j\in\mathcal{S}_s}(1 - p_{s,j})} - \frac{1}{S}\left(1 - q_i(a_{i,m})\right)\right)\right) = \sum_{m\in\mathcal{S}} \frac{1}{K_m^{(S)}}\left(\sum_{i\in\mathcal{U}} K_i a_{i,m}\left(D_m - \left(1 - q_i(a_{i,m})\right)\right)\right). \tag{18}$$

In the definition of B, the following equations hold:

$$K_m^{(S)} = \sum\nolimits_{i\in\mathcal{U}} K_i a_{i,m} \tag{19}$$

and

$$D_m = \sum_{s\in\mathcal{S}} \frac{1 - p_{s,m}}{1 + \sum_{j\in\mathcal{S}_s}(1 - p_{s,j})}. \tag{20}$$

Proof: See the Appendix.

It is observed from (16) that $\mathbb{E}(F(\boldsymbol{g}_{t+1}) - F(\boldsymbol{g}^*))$ is upper bounded by $(2\omega_1 B/[L(1 - A)])$ when t is toward to infinity and A is less than zero. This upper bound gives the gap between the convergent global training loss and the optimal global training loss. This observation motivates us to minimize the upper bound to pursue the global training loss minimization in problem (10). Therefore, we can transform the global training loss minimization in problem (10) as $[2\omega_1 B/(1 - A)]$ minimization, further as B minimization. Besides, we can find from (18), that B is related to the PER of client–server and server–server communication. That is, through B, we can not only build the explicit relationship between optimization variable $\boldsymbol{a}$ and the global training loss, but also depict the impact of unreliable communications on the learning performance.

B. Algorithm Design

Based on Theorem 1, we can recast the global training loss minimization in problem (10) as B minimization. Specifically, the reformulated problem is given by

$$\min_{\boldsymbol{a}} \sum_{m\in\mathcal{S}} \frac{1}{K_m^{(S)}}\left(\sum_{i\in\mathcal{U}} K_i a_{i,m}\left(D_m - \left(1 - q_i(a_{i,m})\right)\right)\right), \tag{21a}$$

$$\text{s.t.}\quad (10a)-(10c). \tag{21b}$$

Problem (21) is an integer nonlinear programming problem, because both the objective function and the constraints are nonlinear, and the optimization variables are integers. To tackle the integer nonlinear programming problem, a variety of mature optimizers have been developed. In this regard, we then employ Gurobi optimizer [37] for a client scheduling scheme to address problem (21).

To optimize the client–server associations, first, the edge servers should obtain the PER of each client in all possible client–server association cases, i.e., $q_i(a_{i,s})\ \forall i \in \mathcal{U}\ \forall s \in \mathcal{S}$, which requires the servers to learn the SNR of wireless channels in all possible client–server association cases, together with the local data size of each client, i.e., $K_i\ \forall i \in \mathcal{U}$. Second, the servers obtain the PER when receiving model updates from adjacent servers by learning the SNR of wireless channels between two adjacent servers, i.e., $p_{s,j}\ \forall s \in \mathcal{S}\ \forall j \in \mathcal{S}_s$. Third, the servers calculate D_s according to $p_{s,j}\ \forall s \in \mathcal{S}\ \forall j \in \mathcal{S}_s$. After obtaining the above parameters, one edge server is then selected to gather the above information, where the Gurobi optimizer is leveraged to obtain the optimal client–server association matrix $\boldsymbol{a}^*$, which is then broadcasted to all edge servers. The clients in the system are then associated with the edge servers according to $\boldsymbol{a}^*$. The FL process in Algorithm 1 begins right after the clients and servers are connected. The

Algorithm 2: Client Scheduling Scheme Among Multiple Edge Servers

1 Calculate the PER of clients in all possible client-server association cases $q_i(a_{i,s})\ \forall i \in \mathcal{U}\ \forall s \in \mathcal{S}$ according to (1);
2 Obtain the local data size of each client $K_i\ \forall i \in \mathcal{U}$;
3 Calculate the PER between adjacent edge servers $p_{s,j}\ \forall s \in \mathcal{S}\ \forall j \in \mathcal{S}_s$ according to (3), then obtain D_s;
4 Optimize the optimal $\boldsymbol{a}$ according to (21) via Gurobi;
5 Broadcast the optimal $\boldsymbol{a}$ to all edge servers, then allocate the clients to associated edge servers;
6 The clients and servers collaboratively train the FL model according to Algorithm 1.

TABLE I
SIMULATION SETTINGS

Parameter	Value
Size of the area	500 m * 500 m
Transmission power of client devices	10 mW
Transmission power of servers	1 W
Uplink resource block bandwidth	1 Mbps
Inter-server communication bandwidth	10 Mbps
Noise power spectral density	-173 dBm/Hz
Batch size	64
Learning rate (MNIST / CIFAR-10)	0.05 / 0.02

process of the proposed client scheduling scheme is specified in Algorithm 2.

V. Simulation Results

In this section, we evaluate the performance of the proposed client scheduling scheme. All the simulations are conducted on a personal computer with CPU @ 2.10 GHz Intel Core i7-12700F and 32 GB of RAM. Specifically, we assume that there are 5 edge servers and 25 clients (if not specified) randomly located in a square area with the size of 500 m * 500 m, with $U_{\max}$ set as U/S. The edge servers are fully connected. The transmission power of clients and servers are 10 mW and 1 W, respectively. The uplink RB bandwidth is 1 Mb/s, the interserver communication bandwidth is 10 Mb/s, and the noise power spectral density is –173 dBm/Hz referring to [22] (if not specified). The simulation settings are summarized in Table I.

For the training data sets, we consider non-i.i.d. local training data among clients and conduct the experiments using the MNIST and CIFAR-10 data sets, respectively. Specifically, the MNIST data set contains a total of 70 000 images of handwritten numbers 0-9 for training a number recognition model, while the CIFAR-10 data set contains 60 000 color images in 10 categories for training an image classification model. Before the training process, each client is allocated with two random categories of training images to form a non-i.i.d. local data set distribution. For the MNIST data set, a multilayer perceptron (MLP) consisting of one fully connected layer before the output layer, with totally 101 770 trainable parameters, is deployed on each client device. For the CIFAR-10 data set, a convolutional neural network (CNN) is deployed instead, with two convolutional layers, pooling layers, and one fully connected layer before the output layer, containing totally 313 802 trainable parameters. During the training process, the batch size is set as 64. The learning rate is set as 0.05 and 0.02 for MNIST and CIFAR-10 data set, respectively. To evaluate the performance of FL training, we record the classification accuracy of the current global model on the test data set (namely, test accuracy), and the average training loss of clients, for every 50 rounds during the training process.

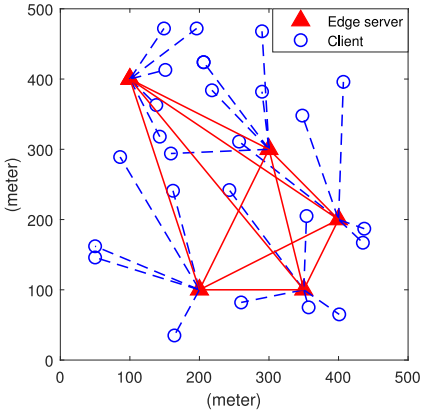


Fig. 2. Example of the client–server association under Algorithm 2.

We adopt the following baselines: a) a random client scheduling algorithm that randomly allocates clients to servers while satisfying (10a)–(10c); b) a greedy client scheduling algorithm that allows the clients to greedily select the server with the smallest PER, without considering the PER between the servers; and c) the FedAvg algorithm [6] which is a classic single-server FL framework. Note that for baseline c), we assume that there is a single central server located in the middle of the square area, and all the clients are connected to the server via wireless channels (thus scheduling is not considered). The local models are aggregated at the central server every local update, with the remaining simulation settings being the same. For reasons of simplicity, we denote baseline a), b) and c) as "Random," "Greedy" and "FedAvg" for short, respectively, and our proposed Algorithm 2 as "Proposed."

Fig. 2 gives an example of the association of the clients with the edge servers which is obtained by the client scheduling scheme in Algorithm 2. From Fig. 2 we can infer that, instead of being associated with the edge server with the shortest physical distance (which means the smallest uplink PER for this client if the influence of small-scale fading is not considered), some client devices are allocated to servers that are still further. Empirically, for each client, a lower PER implies higher model transmission reliability and training efficiency. However, in this article, to minimize the global training loss of the system, we minimize the objective function given in (21) to obtain the optimal association matrix. Scheduling the clients according to this matrix can thus maximize the overall client participation reliability in the considered FL system.

Figs. 3 and 4 show the test accuracy and training loss over the training process for different schemes with MINIST data set, and Figs. 5 and 6 for CIFAR-10 data set. It can be inferred from the figure that compared with baseline a) and b) (i.e., the

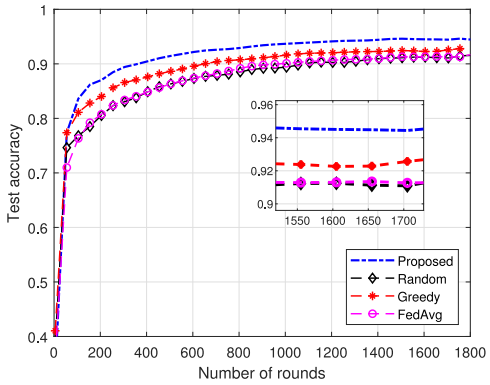


Fig. 3. Test accuracy over number of rounds with MNIST data set.

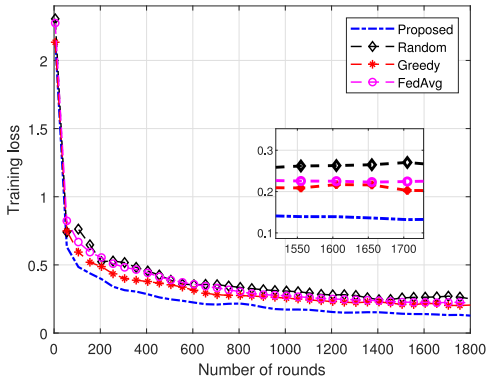


Fig. 4. Training loss over number of rounds with MNIST data set.

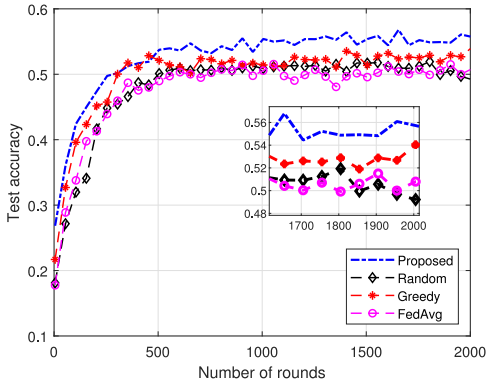


Fig. 5. Test accuracy over number of rounds with CIFAR-10 data set.

Random and Greedy), the proposed client scheduling scheme can achieve higher test accuracy and lower training loss within given number of training rounds. This means that the proposed scheme can effectively improve the reliability of the clients participating in the training process, hence improve the training efficiency. On the other hand, compared with baseline c) (i.e., FedAvg), aside from improving the training efficiency, the proposed scheme among multiple edge servers can naturally alleviate the impact of single-point failures while FedAvg, based on single–server FL, cannot.

In Fig. 7, we show the mean value and standard deviation of test accuracy between round 1500 and 2000 (where the test accuracy tends to converge), over different number of clients with MNIST data set. On one hand, as the total number

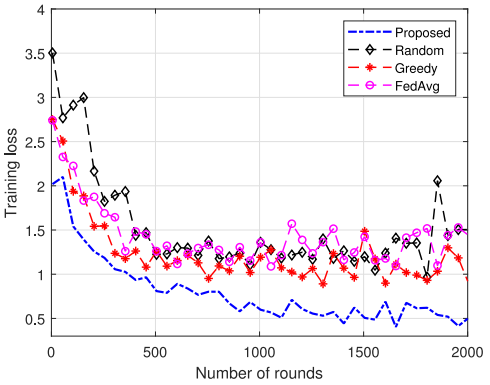


Fig. 6. Training loss over number of rounds with CIFAR-10 data set.

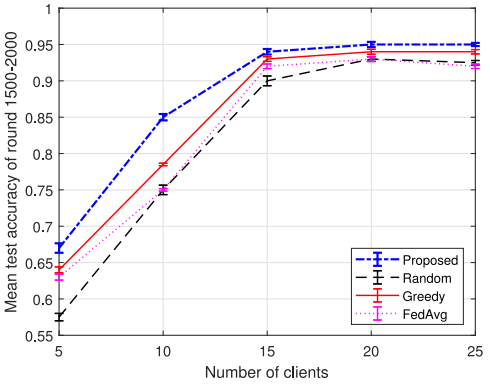


Fig. 7. Mean test accuracy of round 1500–2000 over number of clients with MNIST data set.

of clients in the system varies from 5 to 25, the mean test accuracy increases accordingly. This is due to the fact that the local training data of clients are non-i.i.d., meaning that the lack of training data sets in the system will result in significant performance degradation due to the missing of certain kinds of patterns for identification. However, the effect of continuously increasing the number of clients on improving accuracy is gradually decreasing. This is because the FL system is gaining enough training data samples for an accurate model. On the other hand, the proposed scheme achieves the highest mean test accuracy over different number of clients, which shows the effectiveness of the proposed scheme in improving the reliability of client participation and the training efficiency.

Fig. 8 demonstrates the mean objective value of problem (21) over different number of total clients with MNIST data set. Specifically, the mean objective value is denoted as $(1/U)\sum_{m=1}^{S}(1/K_m^{(S)})\sum_{i=1}^{U}K_i a_{i,m}(D_m-(1-q_{i,m}))$. We can infer from the figure that the mean objective value of the proposed scheme is the smallest for different number of clients, meaning that the proposed scheme can reduce the training loss during the FL training process and accelerate the convergence of the global model. Meanwhile, as we add more clients to the considered FL system, the mean objective value decreases accordingly, meaning that adding clients appropriately can increase the efficiency of the client scheduling to some degree.

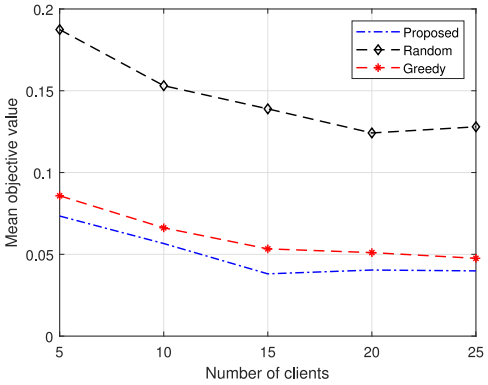


Fig. 8. Mean objective value over number of clients with MNIST data set.

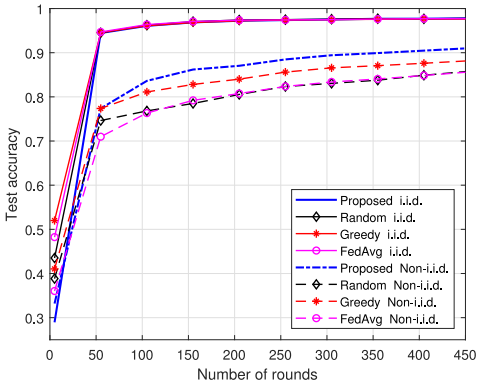


Fig. 10. Test accuracy over number of rounds under different data distributions.

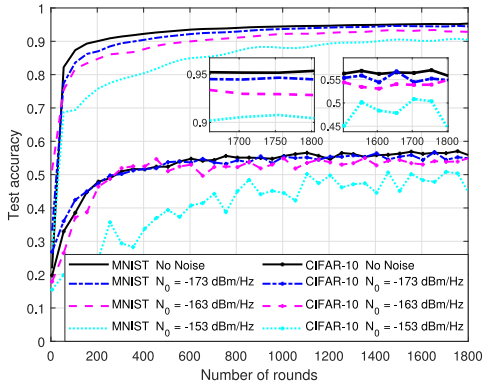


Fig. 9. Test accuracy over number of rounds under different noise power spectral densities.

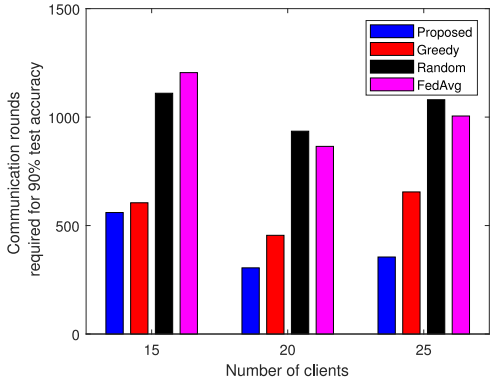


Fig. 11. Communication overhead over number of clients with MNIST data set.

In Fig. 9, in order to evaluate the impact of unreliable communications on the training performance, we demonstrate the test accuracy over number of training rounds with both MNIST and CIFAR-10 data sets under increasing noise power spectral density. Specifically, we increase N_0 to -163 dBm/Hz and –153 dBm/Hz from –173 dBm/Hz [22], respectively, and perform the SD-FL training. The clients are scheduled according to the proposed client scheduling scheme. An ideal situation where there is no noise in wireless channels is also considered in contrast, which means the PER in model transmission is equal to zero. Experimental results show that for both MNIST and CIFAR-10 data sets, the increase of noise power spectral density slows down the convergence speed of global model, and enlarges the fluctuation of test accuracy during model training. This is due to the fact that, with all other factors being equal, an increase in the noise power spectral density will lead to an increase in the PER of model transmission, hence more model packets have to be discarded during the training process, adversely affecting the convergence of the global model. We can also infer from Fig. 9 that the situation where there is no noise gives an upper bound of the training performance of the system.

In Fig. 10, we compare the model training process of aforementioned scheduling methods with both i.i.d. and non-i.i.d. local data set distributions. Note that for the non-i.i.d. scenario, each client is allocated with only two random categories of training images. We can conclude from the results that for the non-i.i.d. scenario, the proposed scheme achieves the highest test accuracy within given number of training rounds among all four methods. Moreover, the global model converges faster in i.i.d. scenario than in non-i.i.d. scenario, which is common in FL. It is worth noting that there is no significant difference in the training results of the proposed scheduling scheme and the baselines in i.i.d. scenario. This is due to the fact that in the i.i.d. scenario, the packet error does not cause the local models trained on certain categories of images to not be included in global model aggregation in some training rounds, which can cause slower convergence of global model in non-i.i.d. scenario. In i.i.d. scenario, due to the sufficient training data samples in the system, the occasional absence of some local models caused by packet errors will not have a significant impact on the convergence of global model.

Furthermore, the communication overhead in terms of the number of total communication rounds for reaching a certain degree of test accuracy (i.e., 0.9) with MNIST data set over different number of clients is demonstrated in Fig. 11. We find from the figure that the number of communication rounds required by the proposed scheme is smaller than those of the baselines for different number of clients. This further verifies that the proposed scheme can effectively reduce the communication overhead in terms of communication rounds during the model training.

VI. Conclusion

In this article, the client scheduling problem over a realistic multiserver IIoT network has been studied. Both client–server

and server–server communication unreliability has been considered. First, a client–server association optimization problem has been formulated, with the objective of minimizing the global training loss. Second, the original problem has been simplified and transformed into an integer nonlinear programming problem through convergence analysis. Finally, a client scheduling scheme based on Gurobi optimizer has been proposed with low-computational complexity. Simulation results have shown that the proposed scheme can effectively improve the reliability of client participation and cut down the communication overhead, hence improving the training efficiency.

APPENDIX
PROOF OF THEOREM 1

For server s, $F(\boldsymbol{g}_{t+1})$ can be given by

$$F^{(s)}(\boldsymbol{g}_{s,t+1}) = \frac{1}{K_s^{(S)}} \sum_{i\in\mathcal{U}_s} \sum_{k=1}^{K_i} f(\boldsymbol{g}_{s,t+1}, \boldsymbol{x}_{ik}, \boldsymbol{y}_{ik}) = \frac{1}{K_s^{(S)}} \sum_{i\in\mathcal{U}_s} F_i(\boldsymbol{g}_{s,t+1}) \tag{22}$$

where $K_s^{(S)} = \sum_{i\in\mathcal{U}} K_i a_{i,s}$ is the local data size of the edge server s. Then, the second-order Taylor expansion of (22) is given by

$$\begin{aligned} F^{(s)}(\boldsymbol{g}_{s,t+1}) &= F^{(s)}(\boldsymbol{g}_{s,t}) + (\boldsymbol{g}_{s,t+1} - \boldsymbol{g}_{s,t})^{\mathrm{T}} \nabla F^{(s)}(\boldsymbol{g}_{s,t}) \\ &\quad + \frac{1}{2}(\boldsymbol{g}_{s,t+1} - \boldsymbol{g}_{s,t})^{\mathrm{T}} \nabla^2 F^{(s)}(\boldsymbol{g}_{s,t})(\boldsymbol{g}_{s,t+1} - \boldsymbol{g}_{s,t}) \\ &\le F^{(s)}(\boldsymbol{g}_{s,t}) + (\boldsymbol{g}_{s,t+1} - \boldsymbol{g}_{s,t})^{\mathrm{T}} \nabla F^{(s)}(\boldsymbol{g}_{s,t}) \\ &\quad + \frac{L}{2}\left\|\boldsymbol{g}_{s,t+1} - \boldsymbol{g}_{s,t}\right\|^2 \end{aligned} \tag{23}$$

where the inequality holds according to assumption 3. Based on (12), the mathematical expectation of (23) can be given by

$$\begin{aligned} &\mathbb{E}\left(F^{(s)}(\boldsymbol{g}_{s,t+1})\right) \\ &\le \mathbb{E}\left(F^{(s)}(\boldsymbol{g}_{s,t}) - \lambda\left(\nabla F^{(s)}(\boldsymbol{g}_{s,t}) - \boldsymbol{o}_s\right)^{\mathrm{T}} \nabla F^{(s)}(\boldsymbol{g}_{s,t}) \right. \\ &\quad \left. + \frac{L\lambda^2}{2}\left\|\nabla F^{(s)}(\boldsymbol{g}_{s,t}) - \boldsymbol{o}_s\right\|^2\right), \\ &\overset{(a)}{=} \mathbb{E}\left(F^{(s)}(\boldsymbol{g}_{s,t})\right) - \frac{1}{2L}\left\|\nabla F^{(s)}(\boldsymbol{g}_{s,t})\right\|^2 + \frac{1}{2L}\mathbb{E}\left(\|\boldsymbol{o}_s\|^2\right) \end{aligned} \tag{24}$$

where λ is set to be $\lambda = (1/L)$, and (a) in (24) holds according to $(L\lambda^2/2)\|\nabla F^{(s)}(\boldsymbol{g}_{s,t}) - \boldsymbol{o}_s\|^2 = (1/2L)\|\nabla F^{(s)}(\boldsymbol{g}_{s,t})\|^2 - (1/L)\boldsymbol{o}_s{}^T\nabla F^{(s)}(\boldsymbol{g}_{s,t}) + (1/2L)\|\boldsymbol{o}_s\|^2$. Based on (13), $\mathbb{E}(\|\boldsymbol{o}_s\|^2)$ in (24) can be given by (25), shown at the bottom of the page.

In (25), $i \in \mathcal{N}_1$ denotes the set of clients that have successfully transmitted their models, while $i \in \mathcal{N}_2$ denotes

$$\begin{aligned} &\mathbb{E}\left(\|\boldsymbol{o}_s\|^2\right) \\ &= \mathbb{E}\left(\left\|\sum_{m\in\mathcal{S}} \frac{\left(\sum_{i\in\mathcal{U}} K_i a_{i,m}\left(\frac{1}{S}C_i(a_{i,m}) - \frac{C_{s,m}^{(S)}}{1+\sum_{j\in\mathcal{S}_s} C_{s,j}^{(S)}}\right)\right)\sum_{i\in\mathcal{U},i\in\mathcal{N}_1} a_{i,m}\nabla F_i(\boldsymbol{g}_{m,t})}{K_m^{(S)}\sum_{i\in\mathcal{U}} K_i a_{i,m} C_i(a_{i,m})} + \frac{1}{S}\sum_{m\in\mathcal{S}} \frac{\sum_{i\in\mathcal{U},i\in\mathcal{N}_2} a_{i,m}\nabla F_i(\boldsymbol{g}_{m,t})}{K_m^{(S)}}\right\|\right) \\ &\le \mathbb{E}\left(\sum_{m\in\mathcal{S}} \frac{\left(\sum_{i\in\mathcal{U}} K_i a_{i,m}\left(\frac{C_{s,m}^{(S)}}{1+\sum_{j\in\mathcal{S}_s} C_{s,j}^{(S)}} - \frac{1}{S}C_i(a_{i,m})\right)\right)\sum_{i\in\mathcal{U},i\in\mathcal{N}_1} a_{i,m}\left\|\nabla F_i(\boldsymbol{g}_{m,t})\right\|}{K_m^{(S)}\sum_{i\in\mathcal{U}} K_i a_{i,m} C_i(a_{i,m})} + \frac{1}{S}\sum_{m\in\mathcal{S}} \frac{\sum_{i\in\mathcal{U},i\in\mathcal{N}_2} a_{i,m}\left\|\nabla F_i(\boldsymbol{g}_{m,t})\right\|}{K_m^{(S)}}\right)^2. \end{aligned} \tag{25}$$

$$\begin{aligned} \mathbb{E}\left(\|\boldsymbol{o}_s\|^2\right) &\le \mathbb{E}\left(\sum_{m\in\mathcal{S}} \frac{2\left(\sum_{i\in\mathcal{U}} K_i a_{i,m}\left(\frac{C_{s,m}^{(S)}}{1+\sum_{j\in\mathcal{S}_s} C_{s,j}^{(S)}} - \frac{1}{S}C_i(a_{i,m})\right)\right)}{K_m^{(S)}}\right)^2 \left(\omega_1 + \omega_2\left\|\nabla F^{(s)}(\boldsymbol{g}_{s,t})\right\|^2\right) \\ &= \sum_{m\in\mathcal{S}} \frac{4}{K_m^{(S)2}}\mathbb{E}\left(\sum_{i\in\mathcal{U}} K_i a_{i,m}\left(\frac{C_{s,m}^{(S)}}{1+\sum_{j\in\mathcal{S}_s} C_{s,j}^{(S)}} - \frac{1}{S}C_i(a_{i,m})\right)\right)^2 \left(\omega_1 + \omega_2\left\|\nabla F^{(s)}(\boldsymbol{g}_{s,t})\right\|^2\right) \\ &\le \sum_{m\in\mathcal{S}} \frac{4}{K_m^{(S)}}\mathbb{E}\left(\sum_{i\in\mathcal{U}} K_i a_{i,m}\left(\frac{C_{s,m}^{(S)}}{1+\sum_{j\in\mathcal{S}_s} C_{s,j}^{(S)}} - \frac{1}{S}C_i(a_{i,m})\right)\right)\left(\omega_1 + \omega_2\left\|\nabla F^{(s)}(\boldsymbol{g}_{s,t})\right\|^2\right) \\ &\overset{(b)}{=} \sum_{m\in\mathcal{S}} \frac{4}{K_m^{(S)}}\left(\sum_{i\in\mathcal{U}} K_i a_{i,m}\left(\frac{1-p_{s,m}}{1+\sum_{j\in\mathcal{S}_s}(1-p_{s,j})} - \frac{1}{S}(1-q_i(a_{i,m}))\right)\right)\left(\omega_1 + \omega_2\left\|\nabla F^{(s)}(\boldsymbol{g}_{s,t})\right\|^2\right). \end{aligned} \tag{28}$$

the set of clients that fails to transmit their models. Besides, due to $\|\nabla F_i(\boldsymbol{g}_{m,t})\| \le \sqrt{\omega_1+\omega_2\|\nabla F^{(m)}(\boldsymbol{g}_{m,t})\|^2}$, we have

$$\sum_{i\in\mathcal{U},i\in\mathcal{N}_1} a_{i,m}\left\|\nabla F_i\left(\boldsymbol{g}_{m,t}\right)\right\| \le \sqrt{\omega_1+\omega_2\left\|\nabla F^{(m)}\left(\boldsymbol{g}_{m,t}\right)\right\|^2}\sum_{i\in\mathcal{U}} K_i a_{i,m} C_i\left(a_{i,m}\right) \quad (26)$$

and

$$\sum_{i\in\mathcal{U},i\in\mathcal{N}_2} a_{i,m}\left\|\nabla F_i\left(\boldsymbol{g}_{m,t}\right)\right\| \le \sqrt{\omega_1+\omega_2\left\|\nabla F^{(m)}\left(\boldsymbol{g}_{m,t}\right)\right\|^2}\left(\sum_{i\in\mathcal{U}} K_i a_{i,m}\frac{SC_{s,m}^{(S)}}{1+\sum_{j\in\mathcal{S}_s} C_{s,j}^{(S)}} - \sum_{i\in\mathcal{U}} K_i a_{i,m} C_i\left(a_{i,m}\right)\right). \quad (27)$$

By substituting (26) and (27), (25) can be rewritten as (28), shown at the bottom of the previous page, where (b) in (28) stems from the fact that $\mathbb{E}(C_i(a_{i,m})) = 1 - q_i(a_{i,m})$ and $\mathbb{E}(C_{s,m}^{(S)}) = 1 - p_{s,m}$, according to (2) and (4). Further combining (24) and (28), we can deduce (29), shown at the bottom of the page.

Finally, considering all the servers in the system, we can express $\mathbb{E}(F(\boldsymbol{g}_{t+1}))$ as (30), shown at the bottom of the page. Subtracting $\mathbb{E}(F(\boldsymbol{g}^*))$ from both sides of (30), we can attain $E(F(\boldsymbol{g}_{t+1}) - F(\boldsymbol{g}^*))$ in (31), shown at the bottom of the page. Because of assumption 2 and assumption 3, we have

$$\left\|\nabla F^{(s)}\left(\boldsymbol{g}_{s,t}\right)\right\|^2 \ge 2\mu\left(F^{(s)}\left(\boldsymbol{g}_{s,t}\right) - F\left(\boldsymbol{g}^*\right)\right). \quad (32)$$

$$\mathbb{E}\left(F^{(s)}\left(\boldsymbol{g}_{s,t+1}\right)\right) \le \mathbb{E}\left(F^{(s)}\left(\boldsymbol{g}_{s,t}\right)\right) - \frac{1}{2L}\left\|\nabla F^{(s)}\left(\boldsymbol{g}_{s,t}\right)\right\|^2 + \frac{1}{2L}\sum_{m\in\mathcal{S}}\frac{4}{K_m^{(S)}}\left(\sum_{i\in\mathcal{U}} K_i a_{i,m}\left(\frac{1-p_{s,m}}{1+\sum_{j\in\mathcal{S}_s}\left(1-p_{s,j}\right)} - \frac{1}{S}\left(1-q_i\left(a_{i,m}\right)\right)\right)\right)\left(\omega_1+\omega_2\left\|\nabla F^{(s)}\left(\boldsymbol{g}_{s,t}\right)\right\|^2\right)$$
$$= \mathbb{E}\left(F^{(s)}\left(\boldsymbol{g}_{s,t}\right)\right) + \frac{2\omega_1}{L}\sum_{m\in\mathcal{S}}\frac{1}{K_m^{(S)}}\left(\sum_{i\in\mathcal{U}} K_i a_{i,m}\left(\frac{1-p_{s,m}}{1+\sum_{j\in\mathcal{S}_s}\left(1-p_{s,j}\right)} - \frac{1}{S}\left(1-q_i\left(a_{i,m}\right)\right)\right)\right) - \frac{1}{2L}\left\|\nabla F^{(s)}\left(\boldsymbol{g}_{s,t}\right)\right\|^2\left(1-\omega_2\sum_{m\in\mathcal{S}}\frac{4}{K_m^{(S)}}\left(\sum_{i\in\mathcal{U}} K_i a_{i,m}\left(\frac{1-p_{s,m}}{1+\sum_{j\in\mathcal{S}_s}\left(1-p_{s,j}\right)} - \frac{1}{S}\left(1-q_i\left(a_{i,m}\right)\right)\right)\right)\right). \quad (29)$$

$$\mathbb{E}\left(F\left(\boldsymbol{g}_{t+1}\right)\right) = \sum_{s\in\mathcal{S}}\mathbb{E}\left(F^{(s)}\left(\boldsymbol{g}_{s,t+1}\right)\right) \le \mathbb{E}\left(F\left(\boldsymbol{g}_t\right)\right) + \frac{2\omega_1}{L}\sum_{s\in\mathcal{S}}\sum_{m\in\mathcal{S}}\frac{1}{K_m^{(S)}}\left(\sum_{i\in\mathcal{U}} K_i a_{i,m}\left(\frac{1-p_{s,m}}{1+\sum_{j\in\mathcal{S}_s}\left(1-p_{s,j}\right)} - \frac{1}{S}\left(1-q_i\left(a_{i,m}\right)\right)\right)\right) - \frac{1}{2L}\sum_{s\in\mathcal{S}}\left\|\nabla F^{(s)}\left(\boldsymbol{g}_{s,t}\right)\right\|^2\left(1-\omega_2\sum_{m\in\mathcal{S}}\frac{4}{K_m^{(S)}}\left(\sum_{i\in\mathcal{U}} K_i a_{i,m}\left(\frac{1-p_{s,m}}{1+\sum_{j\in\mathcal{S}_s}\left(1-p_{s,j}\right)} - \frac{1}{S}\left(1-q_i\left(a_{i,m}\right)\right)\right)\right)\right). \quad (30)$$

$$\mathbb{E}\left(F\left(\boldsymbol{g}_{t+1}\right) - F\left(\boldsymbol{g}^*\right)\right) \le \mathbb{E}\left(F\left(\boldsymbol{g}_t\right) - F\left(\boldsymbol{g}^*\right)\right) + \frac{2\omega_1}{L}\sum_{s\in\mathcal{S}}\sum_{m\in\mathcal{S}}\frac{1}{K_m^{(S)}}\left(\sum_{i\in\mathcal{U}} K_i a_{i,m}\left(\frac{1-p_{s,m}}{1+\sum_{j\in\mathcal{S}_s}\left(1-p_{s,j}\right)} - \frac{1}{S}\left(1-q_i\left(a_{i,m}\right)\right)\right)\right) - \frac{1}{2L}\sum_{s\in\mathcal{S}}\left\|\nabla F^{(s)}\left(\boldsymbol{g}_{s,t}\right)\right\|^2\left(1-\omega_2\sum_{m\in\mathcal{S}}\frac{4}{K_m^{(S)}}\left(\sum_{i\in\mathcal{U}} K_i a_{i,m}\left(\frac{1-p_{s,m}}{1+\sum_{j\in\mathcal{S}_s}\left(1-p_{s,j}\right)} - \frac{1}{S}\left(1-q_i\left(a_{i,m}\right)\right)\right)\right)\right). \quad (31)$$

$$\mathbb{E}\left(F\left(\boldsymbol{g}_{t+1}\right) - F\left(\boldsymbol{g}^*\right)\right) \le \frac{2\omega_1}{L}\sum_{s\in\mathcal{S}}\sum_{m\in\mathcal{S}}\frac{1}{K_m^{(S)}}\left(\sum_{i\in\mathcal{U}} K_i a_{i,m}\left(\frac{1-p_{s,m}}{1+\sum_{j\in\mathcal{S}_s}\left(1-p_{s,j}\right)} - \frac{1}{S}\left(1-q_i\left(a_{i,m}\right)\right)\right)\right) + A\mathbb{E}\left(F\left(\boldsymbol{g}_t\right) - F\left(\boldsymbol{g}^*\right)\right). \quad (33)$$

$$A = \left(1 - \frac{\mu}{L} + \frac{4\omega_2\mu}{L}\sum_{s\in\mathcal{S}}\sum_{m\in\mathcal{S}}\frac{1}{K_m^{(S)}}\left(\sum_{i\in\mathcal{U}} K_i a_{i,m}\left(\frac{1-p_{s,m}}{1+\sum_{j\in\mathcal{S}_s}\left(1-p_{s,j}\right)} - \frac{1}{S}\left(1-q_i\left(a_{i,m}\right)\right)\right)\right)\right). \quad (34)$$

$$\begin{aligned}
&\mathbb{E}\big(F(\boldsymbol{g}_{t+1}) - F(\boldsymbol{g}^*)\big) \\
&\quad \le \frac{2\omega_1}{L} \sum_{s\in\mathcal{S}} \sum_{m\in\mathcal{S}} \frac{1}{K_m^{(S)}} \left(\sum\nolimits_{i\in\mathcal{U}} K_i a_{i,m} \left(\frac{1-p_{s,m}}{1+\sum_{j\in\mathcal{S}_s}(1-p_{s,j})} - \frac{1}{S}\big(1-q_i(a_{i,m})\big) \right) \right) \sum_{k=0}^{t-1} A^k + A^t \mathbb{E}\big(F(\boldsymbol{g}_0) - F(\boldsymbol{g}^*)\big) \\
&\quad = \frac{2\omega_1}{L} \sum_{s\in\mathcal{S}} \sum_{m\in\mathcal{S}} \frac{1}{K_m^{(S)}} \left(\sum\nolimits_{i\in\mathcal{U}} K_i a_{i,m} \left(\frac{1-p_{s,m}}{1+\sum_{j\in\mathcal{S}_s}(1-p_{s,j})} - \frac{1}{S}\big(1-q_i(a_{i,m})\big) \right) \right) \frac{1-A^t}{1-A} + A^t \mathbb{E}\big(F(\boldsymbol{g}_0) - F(\boldsymbol{g}^*)\big). \quad (35)
\end{aligned}$$

Plugging (32) into (31), we can further conduct (33) and (34), shown at the bottom of the previous page. In a recursion manner, (35), shown at the top of the page, can be derived from (33), which give the result in Theorem 1.

REFERENCES

[1] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial Internet of Things: Challenges, opportunities, and directions," *IEEE Trans. Ind. Inform.*, vol. 14, no. 11, pp. 4724–4734, Nov. 2018.

[2] J. H. Park, "Advances in future Internet and the Industrial Internet of Things," *Symmetry*, vol. 11, no. 2, p. 244, 2019.

[3] W. Sun, J. Liu, and Y. Yue, "AI-enhanced offloading in edge computing: When machine learning meets Industrial IoT," *IEEE Netw.*, vol. 33, no. 5, pp. 68–74, Sept./Oct. 2019.

[4] F. Foukalas and A. Tziouvaras, "Edge artificial intelligence for Industrial Internet of Things applications: An industrial edge intelligence solution," *IEEE Ind. Electron. Mag.*, vol. 15, no. 2, pp. 28–36, Jun. 2021.

[5] J. Wang, C. Jiang, K. Zhang, X. Hou, Y. Ren, and Y. Qian, "Distributed Q-learning aided heterogeneous network association for energy-efficient IIoT," *IEEE Trans. Ind. Informat.*, vol. 16, no. 4, pp. 2756–2764, Apr. 2020.

[6] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arca, "Communication efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2017, pp. 273–1282.

[7] K. Guo, Z. Chen, H. H. Yang, and T. Q. S. Quek, "Dynamic scheduling for heterogeneous federated learning in private 5G edge networks," *IEEE J. Select. Topics Signal Process.*, vol. 16, no. 1, pp. 26–40, Jan. 2022.

[8] W. Y. B. Lim et al., "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 2031–2063, 3rd Quart., 2020.

[9] J. Konecny, H. B. McMahan, D. Ramage, and P. Richtarik, "Federated optimization: Distributed machine learning for on-device intelligence," 2016, *arXiv:1610.02527v1*.

[10] Y. Sun, J. Shao, Y. Mao, J. H. Wang, and J. Zhang, "Semi-decentralized federated edge learning for fast convergence on non-IID data," in *Proc. IEEE Wireless. Commun. Netw. Conf. (WCNC)*, 2022, pp. 1898–1903.

[11] C. Shi et al., "Ultra-low latency cloud-fog computing for Industrial Internet of Things," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, 2018, pp. 1–6.

[12] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Low-latency federated learning and blockchain for edge association in digital twin empowered 6G networks," *IEEE Trans. Ind. Informat.*, vol. 17, no. 7, pp. 5098–5107, Jul. 2021.

[13] X. Wang, J. Zhang, C. Chen, J. He, Y. Ma, and X. Guan, "Trust-AoI-aware codesign of scheduling and control for edge-enabled IIoT systems," *IEEE Trans. Ind. Informat.*, vol. 20, no. 2, pp. 2833–2842, Feb. 2024.

[14] P. Kumar, R. Kumar, A. Kumar, A. A. Franklin, S. Garg, and S. Singh, "Blockchain and deep learning for secure communication in digital twin empowered Industrial IoT network," *IEEE Trans. Netw. Sci. Eng.*, vol. 10, no. 5, pp. 2802–2813, Oct. 2023.

[15] J. Xu, B. Yang, Y. Liu, C. Chen, and X. Guan, "Joint task offloading and resource allocation for multihop Industrial Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 21, pp. 22022–22033, Nov. 2022.

[16] W. Xia, T. Q. S. Quek, K. Guo, W. Wen, H. H. Yang, and H. Zhu, "Multi-armed bandit-based client scheduling for federated learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 11, pp. 7108–7123, Nov. 2020.

[17] L. Wang, W. Wang, and B. Li, "CMFL: Mitigating communication overhead for federated learning," in *Proc. IEEE Int. Conf. Distrib. Comput. Syst.*, Dallas, TX, USA, 2019, pp. 954–964.

[18] Y. J. Cho, J. Wang, and G. Joshi, "Client selection in federated learning: Convergence analysis and power-of-choice selection strategies," 2020, *arXiv:2010.01243*.

[19] W. Shi, S. Zhou, Z. Niu, M. Jiang, and L. Geng, "Joint device scheduling and resource allocation for latency constrained wireless federated learning," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 453–467, Jan. 2021.

[20] J. Ren, Y. He, D. Wen, G. Yu, K. Huang, and D. Guo, "Scheduling for cellular federated edge learning with importance and channel awareness," *IEEE Trans. Wireless Commun.*, vol. 19, no. 11, pp. 7690–7703, Nov. 2020.

[21] T. Huang, W. Lin, W. Wu, L. He, K. Li, and A. Zomaya, "An efficiency-boosting client selection scheme for federated learning with fairness guarantee," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 7, pp. 1552–1564, Jul. 2021.

[22] 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.

[23] N. Yoshida, T. Nishio, M. Morikura, and K. Yamamoto, "MAB-based client selection for federated learning with uncertain resources in mobile networks," in *Proc. IEEE Global Commun. Conf. Workshops*, Taipei, Taiwan, 2020, pp. 1–6.

[24] B. Xu, W. Xia, J. Zhang, T. Q. S. Quek, and H. Zhu, "Online client scheduling for fast federated learning," *IEEE Wireless Commun. Lett.*, vol. 10, no. 7, pp. 1434–1438, Jul. 2021.

[25] S. Luo, X. Chen, Q. Wu, Z. Zhou, and S. Yu, "HFEL: Joint edge association and resource allocation for cost-efficient hierarchical federated edge learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 10, pp. 6535–6548, Oct. 2020.

[26] B. Xu, W. Xia, W. Wen, P. Liu, H. Zhao, and H. Zhu, "Adaptive hierarchical federated learning over wireless networks," *IEEE Trans. Veh. Technol.*, vol. 71, no. 2, pp. 2070–2083, Feb. 2022.

[27] N. Mhaisen, A. A. Abdellatif, A. Mohamed, A. Erbad, and M. Guizani, "Optimal user-edge assignment in hierarchical federated learning based on statistical properties and network topology constraints," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 1, pp. 55–66, Jan./Feb. 2022.

[28] T. Zhao, F. Li, and L. He, "DRL-based joint resource allocation and device orchestration for hierarchical federated learning in NOMA-enabled Industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 19, no. 6, pp. 7468–7479, Jun. 2023.

[29] H. Saadat, M. S. Allahham, A. A. Abdellatif, A. Erbad, and A. Mohamed, "RL-assisted energy-aware user-edge association for IoT-based hierarchical federated learning," in *Proc. Int. Wireless Commun. Mob. Comput. (IWCMC)*, 2022, pp. 548–553.

[30] B. Xu, W. Xia, J. Zhang, X. Sun, and H. Zhu, "Dynamic client association for energy-aware hierarchical federated learning," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, 2021, pp. 1–6.

[31] X. Wei, J. Liu, X. Shi, and Y. Wang, "Participant selection for hierarchical federated learning in edge clouds," in *Proc. IEEE Int. Conf. Netw. Arch. Storage (NAS)*, 2022, pp. 1–8.

[32] W. Y. B. Lim, J. S. Ng, Z. Xiong, D. Niyato, C. Miao, and D. I. Kim, "Dynamic edge association and resource allocation in self-organizing hierarchical federated learning networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3640–3653, Dec. 2021.

[33] Y. Xi, A. Burr, J. Wei, and D. Grace, "A general upper bound to evaluate packet error rate over quasi-static fading channels," *IEEE Trans. Wireless Commun.*, vol. 10, no. 5, pp. 1373–1377, May 2011.

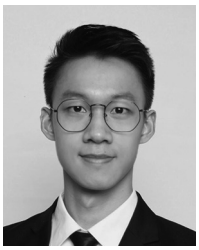
[34] S. U. Stich, "Local SGD converges fast and communicates little," 2018, *arXiv:1805.09767*.

[35] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, "Energy efficient federated learning over wireless communication networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1935–1949, Mar. 2021.
[36] H. H. Yang, Z. Liu, T. Q. S. Quek, and H. V. Poor, "Scheduling policies for federated learning in wireless networks," *IEEE Tans. Commun.*, vol. 68, no. 1, pp. 317–333, Jan. 2020.
[37] Z. Gu, E. Rothberg, and R. Bixby. "Gurobi optimizer reference manual." Gurobi. 2016. [Online]. Available: http://www.gurobi.com

Haitao Zhao (Senior Member, IEEE) received the M.S. and Ph.D. degrees (Hons.) in signal and information processing from Nanjing University of Posts and Telecommunications, Nanjing, China, in 2008 and 2011, respectively.

He is currently the Dean of the School of Internet of Things, Nanjing University of Posts and Telecommunications. From May 2019 to November 2019, he was a Visiting Scholar with the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong. His current research interests include wireless multimedia modeling, ubiquitous wireless communication, and the Internet of Things.

Dr. Zhao received the Second Prize of Jiangsu Science and Technology Award 2019 and the Second Prize of the Technology Invention Award of the Chinese Communication Society in 2017 and 2019.



Yuhao Tan received the B.S. degree in communication engineering from Nanjing University of Posts and Telecommunications, Nanjing, China, in 2021.

He is currently a graduate student with Nanjing University of Posts and Telecommunications.

Mr. Tan is currently with the Faculty of Jiangsu Key Laboratory of Wireless Communications, College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications. His research interests include mobile edge computing and distributed learning.

Kun Guo (Member, IEEE) received the B.E. degree in telecommunications engineering and the Ph.D. degree in communication and information systems from Xidian University, Xi'an, China, in 2012 and 2019, respectively.

From 2019 to 2021, she was a Postdoctoral Research Fellow with the Singapore University of Technology and Design, Singapore. She is currently a Zijiang Young Scholar with the School of Communications and Electronics Engineering, East China Normal University, Shanghai, China. Her research interests include wireless edge computing and intelligence, as well as non-terrestrial networks.

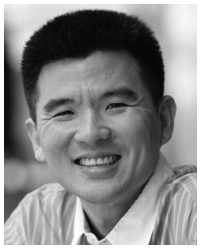
Wenchao Xia (Member, IEEE) received the B.S. degree in communication engineering and the Ph.D. degree in communication and information systems from Nanjing University of Posts and Telecommunications, Nanjing, China, in 2014 and 2019, respectively.

From 2019 to 2020, he was a Postdoctoral Research Fellow with Singapore University of Technology and Design, Singapore. He is currently with the Faculty of the Jiangsu Key Laboratory of Wireless Communications, College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications. His research interests include edge intelligence and multiantenna communications.

Dr. Xia was a recipient of the IEEE Globecom Best Paper Award in 2016 and the IEEE JC&S Best Paper Award in 2022. He serves as an Associate Editor for the *IET Electronics Letters*.

Bo Xu (Member, IEEE) received the B.S. degree in communication engineering and the Ph.D. degree in communication and information systems from Nanjing University of Posts and Telecommunications, Nanjing, China, in 2018 and 2022, respectively.

He is currently with the Faculty of the Jiangsu Key Laboratory of Wireless Communications, College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications. His research interests include mobile edge computing, big data, and distributed learning.



Tony Q. S. Quek (Fellow, IEEE) received the B.E. and M.E. degrees in electrical and electronics engineering from Tokyo Institute of Technology, Tokyo, Japan, in 1998 and 2000, respectively, and the Ph.D. degree in electrical engineering and computer science from Massachusetts Institute of Technology, Cambridge, MA, USA, in 2008.

He is currently the Cheng Tsang Man Chair Professor with Singapore University of Technology and Design (SUTD), Singapore, and the ST Engineering Distinguished Professor. He also serves as the Director of the Future Communications Research and Development Programme, the Head of ISTD Pillar, and the Deputy Director of the SUTD-ZJU IDEA, Hangzhou, China. His current research topics include wireless communications and networking, network intelligence, nonterrestrial networks, open radio access network, and 6G.

Dr. Quek was honored with the 2008 Philip Yeo Prize for Outstanding Achievement in Research, the 2012 IEEE William R. Bennett Prize, the 2015 SUTD Outstanding Education Awards—Excellence in Research, the 2016 IEEE Signal Processing Society Young Author Best Paper Award, the 2017 CTTC Early Achievement Award, the 2017 IEEE ComSoc AP Outstanding Paper Award, the 2020 IEEE Communications Society Young Author Best Paper Award, the 2020 IEEE Stephen O. Rice Prize, the 2020 Nokia Visiting Professor, and the 2022 IEEE Signal Processing Society Best Paper Award. He is currently serving as an Area Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS. He has been actively involved in organizing and chairing sessions, and has served as a member of the Technical Program Committee as well as the symposium chair in a number of international conferences. He is a Fellow of the Academy of Engineering Singapore.