Clustered Federated Learning in Internet of Things: Convergence Analysis and Resource Optimization

Bo Xu, Wenchao Xia, *Member, IEEE,* Haitao Zhao, *Senior Member, IEEE,* Yongxu Zhu, *Senior Member, IEEE,* Xinghua Sun, *Member, IEEE,* and Tony Q. S. Quek, *Fellow, IEEE*

***Abstract*—Federated learning (FL) framework enables user devices to collaboratively train a global model based on their local datasets without privacy leak. However, the training performance of FL is degraded when the data distributions of different devices are incongruent. Fueled by this issue, we consider a clustered FL (CFL) method where the devices are divided into several clusters according to their data distributions and are trained simultaneously. Convergence analysis is conducted, which shows that the clustered model performance depends on cosine similarity, device number per cluster, and device participation probability. Besides, to quantify the training performance, the utility of clustered model training is defined based on the analysis results. Then, aiming at optimizing the system utility, a joint problem of resource allocation and device clustering is formulated, which is solved by decoupling it into two sub-problems. First, given the results of device clustering, a low-complexity iterative algorithm based on the convex optimization theory is proposed to make the bandwidth allocation and the transmit power control. Then, according to the individual stability, a coalition formation algorithm is proposed for the device clustering. Finally, the real-data experiments on the classification tasks (e.g. MNIST, CIFAR-10, CIFAR-100) validate the results of convergence analysis and advantages of the proposed algorithm in terms of the test accuracy.**

***Index Terms*—Clustered federated learning, Internet of Things, resource allocation, coalition formation, convergence analysis**

This work is supported in part by the Science and Technology Innovation 2030-Major Project under Grant 2021ZD0140405, in part by the National Natural Science Foundation of China under Grants 92067201 and 62201285, 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 & Development Programme, in part by the Jiangsu Provincial Key Research and Development Program under Grant BE2020084-1, in part by the China Postdoctoral Science Foundation under Grant 2022M722669, in part by Royal Societys International Exchanges Scheme under Grant IEC\NSFC\211011, and in part by the Natural Science Research Start-up Foundation of Recruiting Talents of Nanjing University of Posts and Telecommunications under Grant NY223026. (*Corresponding authors: Haitao Zhao and Tony Q. S. Quek*)

B. Xu, W. Xia, and H. Zhao are with the Jiangsu Key Laboratory of Wireless Communications, Nanjing University of Posts and Telecommunications, and also with the Engineering Research Center of Health Service System Based on Ubiquitous Wireless Networks, Ministry of Education, Nanjing University of Posts and Telecommunications, Nanjing 210003, China, (e-mail: xubo@njupt.edu.cn, xiawenchao@njupt.edu.cn, zhaoht@njupt.edu.cn).

Y. Zhu is with Department of Electrical and Electronic Engineering, University of Warwick, Coventry CV4 8UW, United Kingdom (e-mail: yongxu.zhu@warwick.ac.uk).

X. Sun is with School of Electronics and Communication Engineering, Shenzhen Campus of Sun Yat-sen University, Shenzhen 518107, China (e-mail: sunxinghua@mail.sysu.edu.cn).

T. 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).

Parts of this work have been accepted by IEEE Globecom 2022 [1].

I. Introduction

Cisco's recent estimation has shown that more than 850 zettabytes of data will be generated each year in the Internet of Things (IoT) networks [2]. These valuable data can bring intelligent services such as smart home and smart city by leveraging machine learning techniques [3]. Conventional learning algorithms need to collect a large number of training samples from the devices and train the machine learning model in a central server [4, 5]. However, due to limited wireless resources in the wireless networks, uploading raw data to the central server can cause extreme transmission latency. Besides, since the data for the IoT applications may include some sensitive information, devices are unwilling to share their local training samples. To address the aforementioned issues, a novel distributed learning framework named federated learning (FL) [6] was recently proposed. In FL, each device can train a machine learning model on their local datasets, and then upload updated local model parameters such as model weights and gradients to a central server. After that, the central server performs the model aggregation and broadcasts the aggregated model parameters to the devices for next round of training. Since only the model parameters are exchanged between the central server and the devices, less wireless resource is utilized and privacy disclosure is mitigated.

However, when the local datasets of the devices are non-independent and identically distributed (non-i.i.d), the high statistical heterogeneity can decrease the accuracy of training [7]. Specially, since the central sever does not have the authority to access the local datasets of devices, some data preparation operations such as outlier detection and balancing are not work for FL [8]. To compensate the learning performance degradation caused by the statistical heterogeneity, novel FL algorithms have been proposed in recent works [9–12]. In [9, 10], some novel stochastic gradient methods for the local model updates were proposed. Besides, authors in [11, 12] improved the convergence rate by considering the "importance" of local updates. Apart from the data heterogeneity issue, training FL in IoT networks with limited computation and communication resources can increase the training latency and degrade the training efficiency. In this regard, recent works have shown that the training efficiency can be significantly improved by performing resource allocation [13–16] and device scheduling [17–19]. Besides, given fixed participating devices, the local computing power, the bandwidth, and the training latency budget were optimized in [13–15]. Combing the wireless resources with the training parameters, adaptive FL was

proposed in [16]. In terms of device scheduling, the authors in [17] analyzed the convergence rate of FL with different device scheduling schemes. Besides, aiming at minimizing the training latency without prior information, multi-armed bandit based online device scheduling methods was proposed in [18, 19]. Moreover, authors in [20–23] jointly performed the resource allocation and the device scheduling to reduce the training latency and improve the learning performance simultaneously.

Note that the most existing works are done under the assumption that there exists a global model that can fit well different data distributions of all devices, which may not hold especially when the devices are with incongruent data distributions. In particular, different devices have varying labels about the similar training samples. For instance, in a task of face recognition [24], assume that one half of the devices judge that people wearing glasses can improve the attraction, while the other half hold the opposite criteria. In this situation, one single global model will never be able to accurately predict the attractiveness of faces for all the devices at the same time. Hence, improving the test accuracy on the individual data distribution is an alternative metric of FL. To achieve this goal, instead of optimizing a consensus global model, Sattler et al. [25] first introduced the concept of clustered FL (CFL), which exploited the similarity of devices' local gradients and divided the devices into clusters based on the bipartition method. There have been some works on CFL [26–31]. Authors in [26] have shown the advantages of model accuracy in CFL under different cases of the incongruent data distribution. A clustering method based on the random initialization was proposed in [27]. Authors in [28] proposed a soft clustering method, which enabled the devices to share the overlapping clustered models. Moreover, the convergence performance of CFL was analyzed in [29]. Latest work [30] reduced the identification accuracy of CFL based visual classification tasks by dividing the devices into multiple groups under the similarities of the data distributions. Besides, in order to excessively consume devices with high computing capability and low remaining energy, an auction-based CFL framework was proposed in [31]. It is worth noting that the CFL does not need to change the communication protocol of the conventional FL. Nonetheless, the influences of limited wireless resources on CFL training performance has not been addressed yet. Although the authors in the above works can achieve higher accuracy of clustering and model testing in the congruent non-i.i.d FL setting, the performance can hardly be guaranteed for the resource constrained wireless CFL applications. Therefore, it is of significant importance to jointly consider the learning performance and the resource allocation in CFL.

In this paper, we consider improving the learning performance of CFL by performing device clustering and resource allocation. Based on the convergence analysis results, the performance of CFL with respect to the cosine similarity, the number of devices per cluster, and the device participation probability are obtained. Then, aiming at improving the learning performance of CFL, a joint resource allocation and device clustering problem is formulated, which can be decoupled into two sub-problems. For the sub-problem of resource allocation, an iterative algorithm based on the convex theory is proposed to optimize the bandwidth allocation and the transmit power control. Then, a coalition formation game model can be considered for device clustering. Motivated by the advantages of coalitional games in addressing various problems of wireless communications, e.g., device clustering [32] and pilot clustering [33], a number of players in coalitional games, i.e., devices in the context of this paper, who cooperatively form coalitions in order to improve the learning performance of clustered models. We further develop a coalition formation algorithm based on the individual stability [34]. Finally, the solution to the original problem can be achieved by iteratively solving the two sub-problems until convergence. Compared to the previous works [25–31], we jointly consider the CFL learning performance and the resource allocation with limited bandwidth resources, and the proposed problem is strictly formulated based on the analysis results in terms of the average cosine similarity and the average cluster size, thus we can improve the training efficiency of CFL in the resource constrained wireless networks. Besides, compared to existing FL works that consider resource allocation problem, one of this paper's method's advantage lies in a rigorous derivation of convergence and generalization, thus the interpretability and learning performance of the CFL is enhanced. In addition, compared with some simple personalized FL (PFL) methods, such as FedRep [35], the considered CFL algorithm can better cope with the problem of data inconsistency without learning the head of local model. Moreover, compared to the previous work in Globecom [1], we have added the convergence analysis for CFL, and proposed a low-complexity iterative algorithm for the resource allocation problem. Besides, in the simulation section, we have used more CFL methods, learning models, data distributions, and datasets for comparison. In general, this work analyzes the effects of the average cosine similarity and the average cluster size for CFL under wireless networks, and can provide a theoretical basis for the implementation of FL in multi-tasking scenarios.

The main contributions of this paper are summarized as follows.

- We consider a novel distributed learning algorithm named CFL to improve the learning performance, in which the devices are grouped into clusters, and several clustered models are trained simultaneously. Besides, we analyze the convergence performance and the generalization ability of CFL with respect to the cosine similarity, the number of devices per cluster, and the device participation probability.
- Aiming at improving the learning performance of CFL, the utility of clustered model training is defined based on the convergence analysis results. Then a joint problem of resource allocation and device clustering is formulated in order to maximize the average device utility. To address this issue, an iterative algorithm of bandwidth allocation and transmit power control is proposed. Besides, according to the individual stability, we develop a coalition formation algorithm for the device clustering.

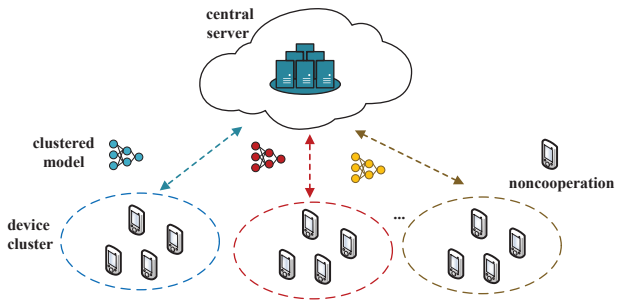


Fig. 1. The structure of CFL system.

TABLE I
SUMMARY OF MAIN NOTATIONS

Notation	Description
$\mathcal{K}$, K	Set of devices, size of $\mathcal{K}$
$\mathcal{S}$, S	Set of coalition indexes, number of coalitions
$\mathcal{W}$, $\boldsymbol{w}_s$,	Set of clustered models, clustered model of cluster s
Π, $\mathcal{V}_s$	Clustering strategy, device set of cluster s
$C_{k,k'}$	Cosine similarity between device k and device k'
ς_k, χ_k	Fluctuation of the computation capability, the maximum of the computation capability
τ_k^{c}	Average local computation latency of device k
τ_k^{u}	Average uploading latency of device k
U_s	Utility of coalition s
$u_k(\Pi)$	Utility of device k under coalition structure Π

- Simulation results on three popular datasets (e.g. MNIST, CIFAR-10 and CIFAR-100) are conducted and show that compared to the representative baselines, the proposed algorithm jointly considering the learning performance and the resource allocation can achieve higher test accuracy with limited wireless resources.

The remainder of this paper is organized as follows. Section II presents the system model and makes the convergence analysis. Then an optimization problem incorporating the bandwidth allocation, the transmit power control, and the device clustering is formulated. In section III, the problem is transformed into two sub-problems, and can be solved iteratively. In section IV, we present the simulation results, and in section V, we conclude the paper.

II. SYSTEM MODEL

In this section, we first introduce the training procedure, the training latency model, and the energy consumption model of CFL. Then an optimization problem is formulated with the results of the convergence analysis. The main notations are summarized in Table I.

TABLE II
COMMON LOSS FUNCTIONS FOR TRAINING

Model	Loss function
Linear regression	$\frac{1}{2}\|y_k - \boldsymbol{w}^{\mathrm{T}} x_k\|$
K-means	$\frac{1}{2}\min_i \|x_k - \boldsymbol{w}_i\|$ with $\boldsymbol{w}_i \triangleq [\boldsymbol{w}_1^{\mathrm{T}}, \boldsymbol{w}_2^{\mathrm{T}}, ...]^{\mathrm{T}}$
Squared-SVM	$\frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{\varrho}{2}\max\{0; 1 - y_k \boldsymbol{w}^{\mathrm{T}} x_k\}^2$ with constant ϱ
Neural network	Cross-entropy on cascaded transform [3]

A. Learning model

As shown in Fig. 1, consider a wireless network consisting of a set $\mathcal{K}$ of K devices and a central server. Each device k has a local dataset $\mathcal{D}_k = \{(x_{k,i}, y_{k,i})\}_{i=1}^{D_k}$, where $x_{k,i}$ denotes the i-th input training sample and $y_{k,i}$ is the labeled output of $x_{k,i}$. Hence the dataset for all devices is defined as $\mathcal{D} = \bigcup_{k\in\mathcal{K}} \mathcal{D}_k$. These devices are divided into different clusters and the cluster set is denoted as $\Pi = \{\mathcal{V}_1, \mathcal{V}_2, ..., \mathcal{V}_{|\Pi|}\}$, where $\mathcal{S} = \{1, 2, ..., |\Pi|\}$ is the index set of clusters, and $\mathcal{V}_s$ is the s-th cluster of the devices with $\mathcal{V}_s \cap \mathcal{V}_{s'} = \emptyset$ for $s \neq s'$. Let $a_{k,s} \in \{0, 1\}$ denote the clustering strategy of device k. Specifically, if device $k \in \mathcal{V}_s$, we have $a_{k,s} = 1$, otherwise, $a_{k,s} = 0$. The clustered model of cluster s is denoted as $\boldsymbol{w}_s$, and the set of clustered models is denoted as $\mathcal{W} = \{\boldsymbol{w}_1, \boldsymbol{w}_2, ..., \boldsymbol{w}_{|\Pi|}\}$.

Then the objective function of CFL is expressed as [29]

$$\min_{\mathcal{W}} \sum_{s\in\mathcal{S}} \frac{\sum_{k\in\mathcal{K}} a_{k,s} D_k F_k(\boldsymbol{w}_s)}{\sum_{k\in\mathcal{K}} a_{k,s} D_k}, \tag{1}$$

where $F_k(\boldsymbol{w}_s) = \sum_{\{x_{k,i}, y_{k,i}\}\in\mathcal{D}_k} l(\boldsymbol{w}_s, x_{k,i}, y_{k,i})$ and $l(\boldsymbol{w}_s, x_{k,i}, y_{k,i})$ captures the error of the model parameter $\boldsymbol{w}_l$ on the training data pair $\{x_{k,i}, y_{k,i}\}$. Some examples of popular loss functions are summarized in Table II.

The CFL training includes the following steps.

1) At the beginning of training, given a global model $\boldsymbol{w}$, the devices first perform local updates and evaluate the cosine similarity. In particular, the cosine similarity between device k and device k' is denoted as [25]

$$C_{k,k'}(\boldsymbol{w}) = \frac{< \nabla F_k(\boldsymbol{w}), \nabla F_{k'}(\boldsymbol{w}) >}{\|\nabla F_k(\boldsymbol{w})\| \|\nabla F_{k'}(\boldsymbol{w})\|}, \tag{2}$$

where $\nabla F_k(\boldsymbol{w})$ is the local gradient of device k on the initialized model parameter $\boldsymbol{w}$. Based on the obtained cosine similarity, the central server evaluates the cluster structure Π and initializes the clustered model set $\mathcal{W}$. We assume that all the local gradients can be received by the center server successfully at the step of clustering. Then the initialized clustered model of cluster s is given by

$$\boldsymbol{w}_s^{(1)} = \boldsymbol{w} - \eta \frac{\sum_{k\in\mathcal{K}} a_{k,s} D_k \nabla F_k(\boldsymbol{w})}{\sum_{k\in\mathcal{K}} a_{k,s} D_k}, \tag{3}$$

where $\eta > 0$ is the learning rate.

2) In training round r, each device $k \in \mathcal{V}_s$, $\forall s \in \mathcal{S}$, receives the clustered model $\boldsymbol{w}_s^{(r)}$ from the central server, and then evaluates its local gradient $\nabla F_k(\boldsymbol{w}_s^{(r)})$ by applying the gradient descent algorithm on its local dataset.

Algorithm 1 The iteration procedure of CFL algorithm.

Initialize the model parameter $\boldsymbol{w}$ as a random vector.
Each device $k \in \mathcal{K}$ evaluates and uploads its local gradient $\nabla F_k(\boldsymbol{w})$ to the central server to evaluate the cosine similarity and obtain the cluster structure Π with initialized clustered model set $\mathcal{W}$.
for $r = 1, 2, ..., R$ **do**
  Devices receive the corresponding clustered models from the central server.
  for cluster $s = 1, 2, ..., |\Pi|$ in parallel **do**
    If $|\mathcal{V}_s|$=1
      The device in $\mathcal{V}_s$ trains its clustered model locally without cooperation.
    else
      Each device $k \in \mathcal{V}_s$ evaluates and transmits its local gradient $\nabla F_k(\boldsymbol{w}_s^{(r)})$ to the central server.
      Central server performs the clustered model aggregation to evaluate the updated clustered model $\boldsymbol{w}_s^{(r+1)}$.
    end if
  end for
end for

3) If $|\mathcal{V}_s| > 1$, $\forall s \in \mathcal{S}$, each device $k \in \mathcal{V}_s$ uploads their evaluated gradients to the central server for model aggregation. Besides, if $|\mathcal{V}_s| = 1$, $\forall s \in \mathcal{S}$, each device $k \in \mathcal{V}_s$ can train its clustered model locally without cooperation. Note that due to limited computation and communication resources, some devices may fail to finish local training or upload its local gradient to the central server. Hence, we have $q_k^{(r)} = 1$ to indicate that device k can participate in training successfully and $q_k^{(r)} = 0$ otherwise. Then the central server updates each clustered model $\boldsymbol{w}_s^{(r+1)}$ as

$$\boldsymbol{w}_s^{(r+1)} = \boldsymbol{w}_s^{(r)} - \eta \frac{\sum_{k\in\mathcal{K}} a_{k,s} q_k^{(r)} D_k \nabla F_k(\boldsymbol{w}_s^{(r)})}{\sum_{k\in\mathcal{K}} a_{k,s} q_k^{(r)} D_k}. \quad (4)$$

4) Steps 2) and 3) are repeated until convergence.

For a more clear description, we provide the iteration procedure of CFL in **Algorithm 1**.

B. Latency Model

According the CFL training process, we find that the training latency contains two main parts. The first part is the local computation latency and local gradient uploading latency of devices. The second part is the clustered model aggregation latency and the clustered model downlink transmission latency of central server. Since the central server is commonly rich in computation resource and typically has high transmit power compared to the devices, the second part is ignored in this work.

Due to the randomness of the local computation capability, we adapt the shifted exponential distribution to characterize the probability distribution of computation latency τ_k^{c} for device k to perform local updates in each training round [21], i.e.,

$$\mathbb{P}(\tau_k^{\mathrm{c}} \le \psi_k) = \begin{cases} 1 - e^{-\frac{\varsigma_k(\psi_k - D_k\chi_k)}{D_k}}, & \text{if } \psi_k \ge D_k\chi_k, \\ 0, & \text{otherwise}, \end{cases} \quad (5)$$

where ψ_k is the time reserved for device k to perform local updates, $\varsigma_k > 0$ and $\chi_k > 0$ are the constants that indicate the fluctuation and the maximum of the computation capability, respectively, and $D_k\chi_k$ is the minimal time consumed by device k to perform local updates.

The spectrum resource is divided into K orthogonal radio access channels for the devices, and each device can access to at most one channel. Then the average uplink rate from device k to the central server can be written as $r_k = b_k \log_2(1 + \frac{|h_k|^2 p_k}{N_0})$, where b_k is the bandwidth allocated for device k, p_k is the transmit power of device k, $|h_k|^2$ is the average channel gain between device k and central server, and N_0 is the background noise. We assume that the devices in the same cluster can share the bandwidth resources, then the total bandwidth allocated to each cluster is set as B, and the bandwidth allocation strategy of cluster s satisfies $\sum_{k\in\mathcal{V}_s} b_k \le B$. Besides, since a device $k \in \mathcal{V}_s$ with $|\mathcal{V}_s| = 1$ can train the clustered model locally without uploading its local gradient, the time consumed for each device k to upload its local gradient is calculated as

$$\tau_k^{\mathrm{u}} = \begin{cases} \dfrac{M}{r_k}, & \text{if } k \in \mathcal{V}_s \text{ and } |\mathcal{V}_s| > 1, \forall s \in \mathcal{S}, \\ 0, & \text{otherwise}, \end{cases} \quad (6)$$

where M is the size of local gradient in bits.

In addition, due to the synchronous model aggregation of training, the training latency budget for all clusters can be defined as $\tau^{\max}$, and the time reserved for device k to perform local updates per round can be evaluated as $\psi_k = \tau^{\max} - \tau_k^{\mathrm{u}}$. Then, according to the definition of $q_k^{(r)}$, the average probability for device k to successfully participate in training in each round r satisfies

$$\mathbb{P}\left(\tau_k^{\mathrm{c}} \le \psi_k\right) = \mathbb{E}\left(q_k^{(r)}\right). \quad (7)$$

C. Energy Consumption Model

The energy consumption of each device k consists of the energy consumed by local updates and the energy for local model uploading, i.e.,

$$e_k = e_k^{\mathrm{c}} + p_k \tau_k^{\mathrm{u}}, \quad (8)$$

where e_k^{c} is the average energy consumption for local training, which is determined by the frequency of the CPU clock, the size of training samples, and the number of CPU cycles required for per bit data [36].

D. Convergence Analysis

Note that the cosine similarity and participation probability, which determine the clustered model performance, and depend

on the device clustering, bandwidth allocation, and transmit power control strategies. In order to quantitatively analyze the influence of the three variables on convergence performance of the CFL, we first introduce the following assumptions and then give a theorem. Besides, to facilitate the analysis, we introduce the following assumptions.

Assumption 1. *Local gradient $\nabla F_k(\boldsymbol{w})$ is Lipschitz continuous with respect to $\boldsymbol{w}$, i.e., $\|\nabla F_k(\boldsymbol{w}) - \nabla F_k(\boldsymbol{w}')\| \leq L\|\boldsymbol{w} - \boldsymbol{w}'\|$, $\forall \boldsymbol{w}, \boldsymbol{w}'$, where L is a positive constant.*

Assumption 2. *Local loss $F_k(\boldsymbol{w})$ is strongly convex with respect to $\boldsymbol{w}$, i.e., $F_k(\boldsymbol{w}) \geq F_k(\boldsymbol{w}') + (\boldsymbol{w} - \boldsymbol{w}')^T \nabla F_k(\boldsymbol{w}') + \frac{\mu}{2}\|\boldsymbol{w} - \boldsymbol{w}'\|^2$, $\forall \boldsymbol{w}, \boldsymbol{w}'$, where μ is a positive constant.*

Assumption 3. *Training loss of cluster s is defined as $\bar{F}_s(\boldsymbol{w})$ which is twice-continuously differentiable, i.e., $\mu \boldsymbol{I} \leq \nabla^2 \bar{F}_s(\boldsymbol{w}) \leq L\boldsymbol{I}$. Besides, the local gradient $\nabla F_k(\boldsymbol{w})$ satisfies $\zeta_1^2 \leq \|\nabla F_k(\boldsymbol{w})\|^2 \leq \zeta_2^2$ with $\xi_1, \xi_2 > 0$, $\forall \boldsymbol{w}$.*

Above standard assumptions have been widely used in the convergence analysis of FL [20, 37, 38], which can be used to describe the characters of the loss functions and the gradients. This is because these assumptions are satisfied by many popular learning models, such as the least-squared SVM and the linear regression. It is worth noting that our analytical results also work well for some popular loss functions which do not satisfy these assumptions. Hence we can derive that these assumptions can contribute to reliable analytical results.

Theorem 1. *Denote by $\boldsymbol{w}_s^*$ the optimal learning model for the devices in $\mathcal{V}_s$. Then, we have*

$$\begin{aligned} &\mathbb{E}\left(\bar{F}_s(\boldsymbol{w}_s^{(r+1)}) - \bar{F}_s(\boldsymbol{w}_s^*)\right) \\ &\leq \left(1 - \frac{\mu}{L}\right)^{r+1} \mathbb{E}\left(\bar{F}_s(\boldsymbol{w}_s^{(1)}) - \bar{F}_s(\boldsymbol{w}_s^*)\right) \\ &+ \frac{L(A_{1,s}^2 - (1 - \frac{\mu}{L})^{r+1} A_{1,s}^2)}{\mu}, \end{aligned} \tag{9}$$

where

$$A_{1,s} = \max_r \frac{\sum_{k\in\mathcal{K}} a_{k,s} D_k \sum_{k'\in\mathcal{K}} a_{k',s} D_{k'} \sqrt{2\xi_2^2 - 2C_{k,k'}(\boldsymbol{w}_s^{(r)})\xi_1^2}}{\sum_{k\in\mathcal{K}} a_{k,s} q_k^{(r)} D_k \sum_{k'\in\mathcal{K}} a_{k',s} D_{k'}}. \tag{10}$$

Proof: See Appendix A for reference.

In addition, note that the types of data distribution are unknown in advance and devices can perform local updates without collaboration in order to minimize the local loss in problem (1). However, due to the limited number of training samples carried by the devices, the trained local models only have limited generalization ability with poor test accuracy [20, 39]. To jointly consider the generalization ability in our analysis, we further define the global training loss function as $F(\cdot)$, and investigate the performance of $F(\cdot)$ under the obtained clustered model $\boldsymbol{w}_s^{(r+1)}$.

Theorem 2. *The upper bound of $\mathbb{E}\left(F(\boldsymbol{w}_s^{(r+1)}) - F(\boldsymbol{w}_s^*)\right)$ is given by*

$$\begin{aligned} &\mathbb{E}\left(F(\boldsymbol{w}_s^{(r+1)}) - F(\boldsymbol{w}_s^*)\right) \leq \left(1 - \frac{\mu}{L}\right)^{r+1} \mathbb{E}\left(F(\boldsymbol{w}_s^{(1)}) - \bar{F}_s(\boldsymbol{w}_s^*)\right) \\ &+ \frac{L(A_{2,s} - (1 - \frac{\mu}{L})^{r+1} A_{2,s})}{\mu}, \end{aligned} \tag{11}$$

where

$$A_{2,s} = \frac{4\xi_2^2 \left(\sum_{k\in\mathcal{K}} D_k - \sum_{k\in\mathcal{K}} a_{k,s} D_k \mathbb{E}\left(q_k^{(r)}\right)\right)}{\sum_{k\in\mathcal{K}} D_k}. \tag{12}$$

Proof: See Appendix B for reference.

According to Theorem 1, we find that a gap $\frac{L(A_{1,s}^2 - (1-\frac{\mu}{L})^{r+1} A_{1,s}^2)}{\mu}$ exists between $\bar{F}_s(\boldsymbol{w}_s^{(r+1)})$ and $\bar{F}_s(\boldsymbol{w}_s^*)$, which relies on the cosine similarity among the devices. This gap decreases as the data distributions of the devices in the same cluster are more congruent. Specially, due to limited communication and computation resources, the central server collects gradient information from all devices will cause a large latency and it is impractical to calculate the cosine similarity in each round. Hence, we perform the device clustering before training and calculate the cosine similarity $C_{k,k'}(\boldsymbol{w})$ to approximate the cosine similarity $C_{k,k'}(\boldsymbol{w}_s^{(r)})$. The average cosine similarity of each cluster can be used to approximate the cosine similarity during the training process, i.e.,

$$\bar{A}_{1,s} = \frac{\sum_{k\in\mathcal{K}} a_{k,s} D_k \sum_{k'\in\mathcal{K}} a_{k',s} D_{k'} C_{k,k'}(\boldsymbol{w})}{\sum_{k\in\mathcal{K}} a_{k,s} D_k \sum_{k'\in\mathcal{K}} a_{k',s} D_{k'}}. \tag{13}$$

In addition, from Theorem 2, it is also observed a gap $\frac{L(A_{2,s} - (1-\frac{\mu}{L})^{r+1} A_{2,s})}{\mu}$ exists between $F(\boldsymbol{w}_s^{(r+1)})$ and $F(\boldsymbol{w}_s^*)$, which relies on the clustering strategy. In particular, for cluster $\mathcal{V}_s$, as the number of devices or the device participation probability increases, the gap between $F(\boldsymbol{w}_s^{(r+1)})$ and $F(\boldsymbol{w}_s^*)$ decreases. Hence, the generalization ability of clustered model $\boldsymbol{w}_s$ can be denoted as

$$\bar{A}_{2,s} = \sum_{k\in\mathcal{K}} a_{k,s} D_k \mathbb{E}\left(q_k^{(r)}\right). \tag{14}$$

Motivated by Theorem 1 and Theorem 2, we can improve the learning performance by grouping the devices with congruent data distribution into the same cluster and enable more devices participate into the training process. Thus, taking into the two factors, we define the utility of cluster s as

$$U_s = \rho \bar{A}_{1,s} + (1 - \rho)\bar{A}_{2,s}, \tag{15}$$

where ρ is the weight to balance the cosine similarity and the generalization ability.

In addition, considering the devices in the same cluster share the clustered model, the utility of device k is defined as

$$u_k(\Pi) = \sum_{s\in\mathcal{S}} a_{k,s} U_s. \tag{16}$$

where $|\Pi|$ represents the cluster structure. Denote $N(|\Pi|)$ as the number of devices that train the local models without cooperation and denote $|\Pi|^{\max}$ as the maximal number of clusters that can occupy the bandwidth resources. Accordingly, the

Algorithm 2 Iterative algorithm for resource allocation.

Initialize: device clustering strategy Π.
for $s = 1, 2, ..., |\Pi|$ **do**
  repeat:
  With given $\mathcal{P}_s$, obtain the solution of $\mathcal{B}_s$ according to (20).
  With given $\mathcal{B}_s$, obtain the solution of $\mathcal{P}_s$ according to (21).
  until Objective value of problem (19) converges.
  if $\sum_{k\in\mathcal{V}_s} b_k^{\min} > B$ **do**
    Cluster s is not feasible for training.
  end if
end for

number of clusters with cooperative training can be evaluated as $|\Pi| - N(|\Pi|)$. Due to the scarcity of bandwidth resource, the number of clusters occupying bandwidth resources is limited, i.e.,

$$|\Pi| - N(|\Pi|) \leq |\Pi|^{\max}. \tag{17}$$

E. Problem Formulation

Define $\mathcal{B} = \{\mathcal{B}_1, \mathcal{B}_2, ..., \mathcal{B}_{|\Pi|}\}$ with $\mathcal{B}_s = \{b_k | k \in \mathcal{V}_s\}$ and $\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2, ..., \mathcal{P}_{|\Pi|}\}$ with $\mathcal{P}_s = \{p_k | k \in \mathcal{V}_s\}$. We aim to maximize the average utility of devices by jointly optimizing resource allocation and device clustering, which is formulated as

$$\max_{\mathcal{B},\mathcal{P},\Pi} \frac{1}{K} \sum_{k\in\mathcal{K}} u_k(\Pi) \tag{18a}$$

$$\text{s.t.} \sum_{k\in\mathcal{V}_s} b_k \leq B, \forall s \in \mathcal{S}, \tag{18b}$$

$$b_k \geq 0, \forall k \in \mathcal{K}, \tag{18c}$$

$$e_k \leq e_k^{\max}, \forall k \in \mathcal{K}, \tag{18d}$$

$$0 \leq p_k \leq p_k^{\max}, \forall k \in \mathcal{K}, \tag{18e}$$

$$\mathbb{E}\left(q_k^{(r)}\right) \geq 0, \forall k \in \mathcal{K}, \tag{18f}$$

$$\sum_{s\in\mathcal{S}} a_{k,s} = 1, \forall k \in \mathcal{K}, \tag{18g}$$

$$a_{k,s} \in \{0, 1\}, \forall k \in \mathcal{K}, s \in \mathcal{S}, \tag{18h}$$

$$|\Pi| - N \leq |\Pi|^{\max}, \tag{18i}$$

where $e_k^{\max}$ and $p_k^{\max}$ are the transmit power budget and the energy consumption budget of device k, respectively. Note that problem (18) is a mixed-integer nolinear programming (MINLP) problem and is hard to solve. To find solutions to problem (18), we can decompose it into two sub-problems with separated objectives: a resource allocation problem including bandwidth allocation and transmit power control with fixed device clustering, and a device clustering problem given the results of resource allocation.

III. Resource Allocation and Device Clustering

A. Resource Allocation

Given Π, the resource allocation problem is equivalent to

$$\max_{\mathcal{B},\mathcal{P}} (1-\rho) \sum_{s\in\mathcal{S}} \sum_{k\in\mathcal{K}} a_{k,s} D_k$$

$$\max\left\{1 - e^{-\frac{\varsigma_k}{D_k}\left(\tau^{\max} - \frac{M}{b_k \log_2(1+\frac{|h_k|^2 p_k}{N_0})} - D_k\chi_k\right)}, 0\right\} \tag{19a}$$

$$\text{s.t.} \quad (18b) - (18f). \tag{19b}$$

From problem (19), given the training latency budget $\tau^{\max}$, the participation probability of some bandwidth resource allocated devices can be less than 0. To avoid this situation and improve the bandwidth utilization, we can solve problem (19) by using the follow theorem.

Theorem 3. *For cluster $\mathcal{V}_s$, $\forall s \in \mathcal{S}$, if $|\mathcal{V}_s| > 1$, the optimal solution $\{b_k^*, p_k^*\}$, $\forall k \in \mathcal{V}_s$, of problem* (19) *is*

$$b_k^* = \max\left\{b_k^{min}, \frac{\varsigma_k M}{2D_k \log_2\left(1 + \frac{|h_k|^2 p_k^*}{N_0}\right) \mathcal{W}\left(\sqrt{\frac{\nu_s^* \varsigma_k M e^{\frac{\varsigma_k}{D_k}(\tau^{max} - D_k\chi_k)}}{4D_k^2(1-\rho)a_{k,s}\log_2\left(1+\frac{|h_k|^2 p_k^*}{N_0}\right)}}\right)}\right\}, \tag{20}$$

and

$$p_k^* = \min\{p_k(b_k^*), p_k^{max}\} \tag{21}$$

where $\mathcal{W}(\cdot)$ is the Lambert $\mathcal{W}$ function, ν_s^* is evaluated according to the constraint $\sum_{k\in\mathcal{K}} a_{k,s} b_k^* = B$,

$$b_k^{min} = \frac{M}{(\tau^{max} - D_k\chi_k)\log_2(1 + \frac{|h_k|^2 p_k}{N_0})} \tag{22}$$

and $p_k(b_k^*)$ is the solution of

$$\frac{p_k M}{b_k^* \log_2(1 + \frac{|h_k|^2 p_k}{N_0})} = e_k^{max} - e_k^c \tag{23}$$

Specially, if $\sum_{k\in\mathcal{K}} a_{k,s} b_k^{min} > B$, we can infer that the cluster $\mathcal{V}_s$ is not feasible for training.

Proof: See Appendix C for reference.

Note that ν_s^* and $p_k(b_k^*)$ can be found with the bisection method, and the time complexity of which is $\mathcal{O}(\log_2(1/\varsigma_1))$ and $\mathcal{O}(\log_2(1/\varsigma_2))$, respectively, where $\varsigma_1 > 0$ and $\varsigma_2 > 0$ are the accuracy that can be obtained using the bisection method. Since the objective function of the problem (19) is convex with respect to the optimization variables which are bounded and continuous, we can notice that the proposed algorithm yields a non-decreasing sequence of the objective value. Therefore, the resource allocation problem based on the proposed iterative algorithm is guaranteed to converge. Besides, The detailed steps of iterative algorithm are provided in **Algorithm 2**, and the complexity of which is $\mathcal{O}\left(IK\left(\log_2(1/\varsigma_1) + \log_2(1/\varsigma_2)\right)\right)$, where I is the number of iterations. Our algorithm can be performed with lower complexity for two main reasons. On the one hand, in each iteration step, the low complexity is reflected in that our iterative algorithm is based on the closed-form

solution, and bandwidth and power values can be obtained quickly with a logarithmic complexity in each iteration. On the other hand, given accuracy ς_3 and ς_4, compared with some classical algorithms, such as gradient descent algorithm with L-smooth and μ-strongly convex, and newtonian method with matrix evaluation complexity ι, whose complexity are evaluated as $\mathcal{O}(\frac{LK}{\mu}\log(1/\varsigma_3))$ and $\mathcal{O}(\iota K\log(1/\varsigma_4))$, respectively, However, gradient descent algorithm needs a proper setting of the step size $\eta = \mathcal{O}(\frac{1}{L})$, and the convergence rate is variational under different smooth and convex assumptions with different optimization variables. Newtonian method is more dependent on the initial value setting and requires updating a K-dimensional matrix in each iteration, which is costly to calculate in computation. Specially, since the proposed algorithm is based on the closed-form solution, in a normal case, the proposed algorithm can achieve lower complexity without other limitations. In addition, compared with other heuristic optimization methods, such as discrete optimization and random walk, the proposed algorithm is obviously to have lower complexity with exact convergence policy. Moreover, if the proposed iterative algorithm is performed completed independently in each cluster, the complexity can be evaluated as $\mathcal{O}\left(I\left(\log_2(1/\varsigma_1)+\log_2(1/\varsigma_2)\right)\right)$ with $I << K$. Hence, under the derived closed-form solution, the proposed algorithm has lower complexity.

B. Device Clustering

Given the results of resource allocation, let $u_k^*(\Pi)$ denote the utility of device k under the cluster structure Π, then the clustering problem can be formulated as

$$\max_{\Pi} \frac{1}{K}\sum_{k\in\mathcal{K}} u_k^*(\Pi) \tag{24a}$$

$$\text{s.t. (18g)} - \text{(18i)}, \tag{24b}$$

Theorem 4. *Problem* (24) *is NP-hard by nature.*

Proof: We conduct the proof through a polynomial-time reduction from the $0-1$ knapsack problem, which is known to be NP-hard [40]. Given a knapsack with capacity W, the goal of the $0-1$ knapsack problem is to maximize the total value, in which each item can be use 0 or 1 time. In particular, the value for the item k to join in knapsack s can be defined as $V_{k,s}$, and the capacity of each item is defined as C_k. We further use $X_{k,s}=1$ to denote that item k is in knapsack s, and $X_{k,s}=0$ otherwise. The strategy of the $0-1$ knapsack problem is defined as $\mathcal{X}$, then the $0-1$ knapsack problem can be defined as

$$\max \sum_{k\in\mathcal{K}} X_{k,s}V_{k,s} \tag{25a}$$

$$\text{s.t. } X_{k,s}\in 0,1, \forall k\in\mathcal{K}, s\in\mathcal{S}, \tag{25b}$$

$$\sum_{k\in\mathcal{K}} X_{k,s}C_k \le M, \forall k\in\mathcal{K}, s\in\mathcal{S}. \tag{25c}$$

We then treat the item, the capacity, and the value as the device, the bandwidth, and the device utility. Consequently, we can obtain a special case for device clustering problem where only one device cluster exists and such a transformation is obviously in polynomial time. Therefore, if we can easily obtain an optimal solution to the $0-1$ knapsack problem instance, an optimal solution to the device clustering instance yields, which reaches a contradiction as device clustering problem is NP-hard. Hence we can complete the proof.

Algorithm 3 Coalition formation algorithm for device clustering.

Initialize $i=0$, $\Pi^{(0)}=\{\{1\},\{2\},...,\{K\}\}$.
Repeat:
for $k=1,2,...,K$ **do**
  Device k finds acceptable coalitions $\Pi_k \subseteq \Pi^{(i)}\cup\varnothing$ that can strictly improve its utility and accepts the deviation.
  If $|\Pi_k|>0$ **do**
    Device k leaves its current coalition and joins a coalition $\mathcal{V}^*\in\Pi_k$ that can achieve the largest utility.
  end if
end for
Update the coalition structure $\Pi^{(i+1)}=\Pi^{(i)}$.
Update the coalition structure index $i=i+1$.
Until there exists no device deviation is admissible.
if the number of coalitions occupying bandwidth resources $|\Pi^{(i)}|-N(\Pi^{(i)})>|\Pi|^{\max}$ **do**
  According to **Algorithm 2**, $|\Pi^{(i)}|-|\Pi|^{\max}$ coalitions with lower average utility are split into multiple singleton coalitions.
end if
Find the optimal clustered model:
Given the cluster structure Π, devices train the clustered models according to **Algorithm 1**.
Devices download all clustered models from the central server.
Each device selects the clustered model with the best learning performance.

To reduce the complexity, rather than using a traversal method, we introduce the coalitional game theory [33, 41], in which each cluster can be seen as a coalition and the devices need to select the coalition they belong to. Besides, in the considered coalitional game, each device's behavior follows the maximization of its utility and achieving the stable coalition structure. We consider the individual stability [34], and use three elements to describe the coalition formation: 1) device deviation, 2) admission of deviation, and 3) individual stability checking.

Definition 1. *(Device Deviation) A device $k\in\mathcal{V}_s$ leaves coalition $\mathcal{V}_s$ and join coalition $\mathcal{V}_{s'}$, $s\neq s'$ and $\mathcal{V}_{s'}\in\Pi\cup\varnothing$. Then the coalition structure Π changes to Π'. We can denote this change as $\Pi \overset{k,s,s'}{\rightarrow} \Pi'$.*

According to the concept of individual stability [34], a device deviation can be admitted only if a device can strictly improve its utility and ensure that the devices in the coalition it joins do not reduce their utility. Such a deviation requirement is based on the fact that a device that wants to join a coalition will share the bandwidth resources of the others, thus must asking for the permission of deviation.

Definition 2. *(Admission of Deviation) A deviation of device*

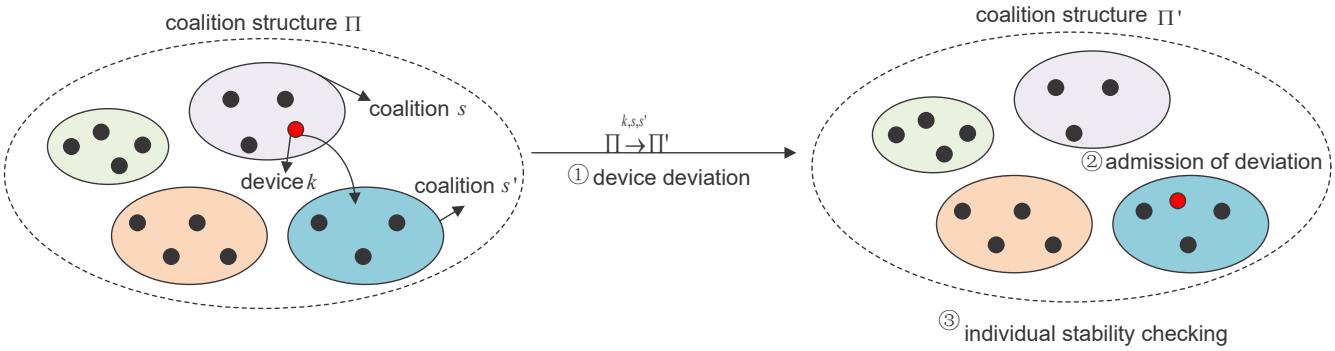


Fig. 2. A sketch map of the proposed coalition formation algorithm.

$k \in \mathcal{V}_s$ to join coalition $\mathcal{V}_{s'}$, $s \neq s'$, i.e., $\Pi \overset{k,s,s'}{\rightarrow} \Pi'$, is admissible only if

$$u_k^*(\Pi) > u_k^*(\Pi'), \tag{26}$$

and

$$u_{k'}^*(\Pi') > u_{k'}^*(\Pi), \forall k' \in \mathcal{V}_{s'}. \tag{27}$$

Then, to ensure the convergence of the proposed coalition formation algorithm, we need to check the individual stability.

Definition 3. *(Individual Stability Checking) If there exists no device deviation is admissible, we can derive that the coalition structure Π is individually stable.*

In **Algorithm 3**, we provide the implementation of coalition formation. The initialized coalition structure is denoted as $\Pi^{(0)}$, which can be the form of singleton coalitions. In line 3, each device k is selected to check if a deviation is profitable, then according to the utility of the current coalition structure Π, device k evaluates its utility if it can joint other coalitions $\Pi_k \subseteq \Pi \cup \varnothing$, where Π_k includes the coalitions that can strictly improve its utility and accept the deviation. In line 5, if $|\Pi_k| > 0$, a coalition $\mathcal{V}^* \in \Pi_k$ that can achieve the largest utility is selected by device k. Iterations terminate when no deviations of devices take place anymore. Specially, in line 13, according to **Algorithm 2**, since the maximal number of clusters that occupy bandwidth resources is limited, we need to check the constraint (18i) when the coalition formulation converges. Note that the objective function is to maximize the average utility of devices, thus we can simply dismantle coalitions with lower average utility into multiple singleton coalitions. In addition, after performing the clustered model training, each device downloads all clustered models from the central server and selects the clustered model with the best learning performance for utilization. The complexity of **Algorithm 3** largely depends on the number of iterations which can be denoted as G. Combined with the resource allocation in each iteration, then the computational complexity of our proposed algorithm is $\mathcal{O}\left(GIK\left(\log_2(1/\varsigma_1) + G\log_2(1/\varsigma_2)\right)\right)$. Since the central server has the powerful computation capability to solve the problem (19) and (24), the overhead of problem solving can be negligible.

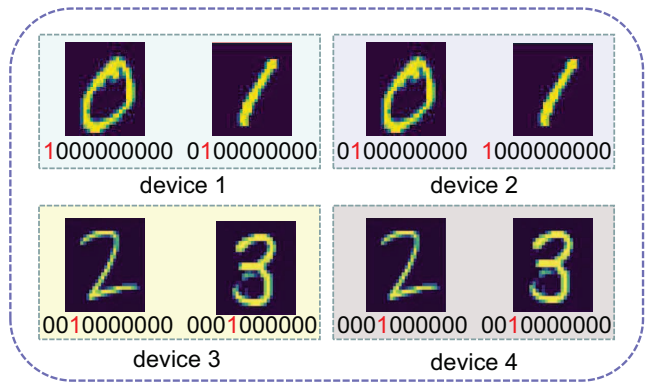


Fig. 3. A sketch map of the incongruent data distribution.

TABLE III
EXPERIMENTAL DATA SETUP

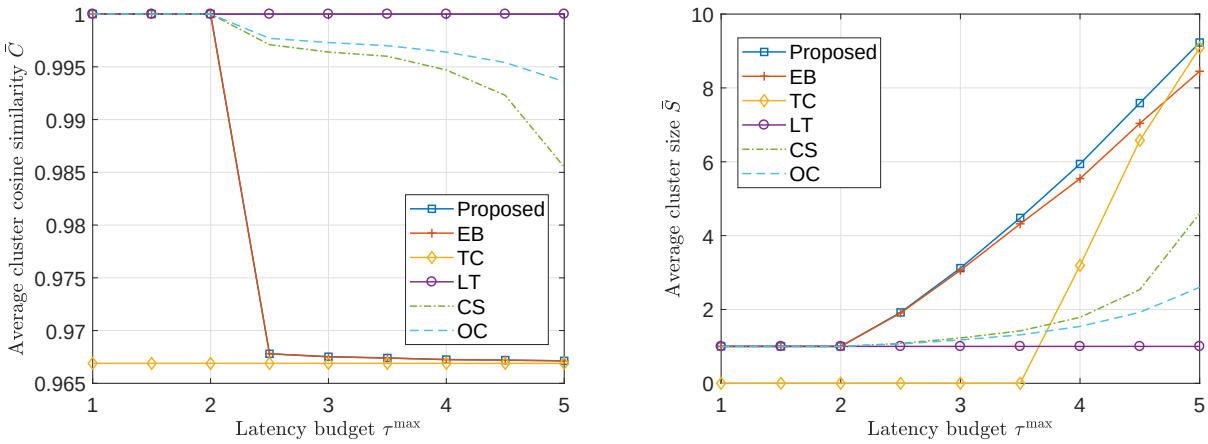
Parameter	Value
K	600
B	400 KHz
N_0	-107 dBm
$\tau^{\max}$	4 seconds
$\lvert\Pi\rvert^{\max}$	70
Training samples per device	1000
Types of labels per device	10
Incongruent training samples per device	200
Types of incongruent labels per device	2

IV. SIMULATION RESULTS

In this section, we conduct experiments to validate the theoretical analysis and test the performance of the proposed algorithm. All simulations are carried out with tensorflow 2.0 on a personal computer with NVIDIA GeForce RTX 2080 Ti.

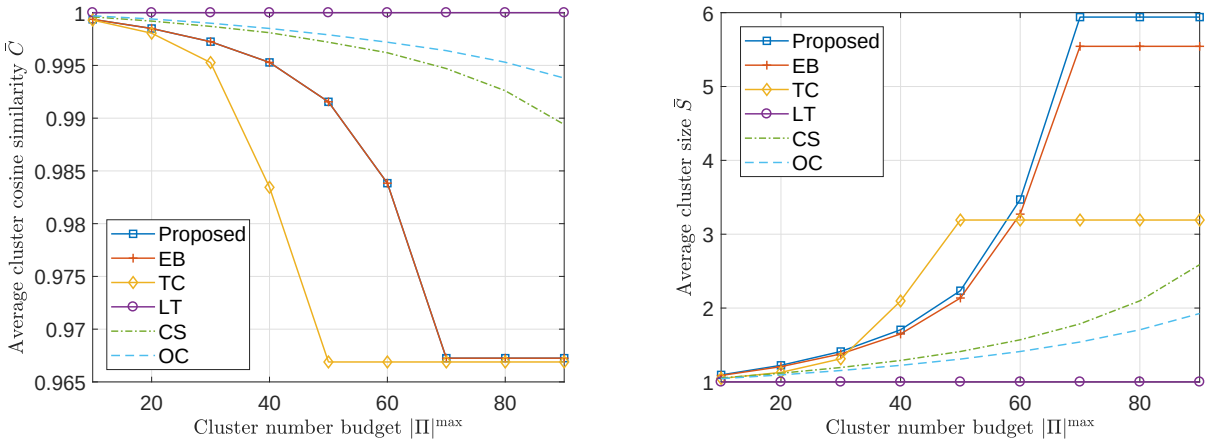
A. Experiment Settings

1) Network Settings: Unless otherwise specified, consider a network of $K = 600$ devices which are uniformly deployed in a disc with a radius of 500 m, and a central server at the center of the disc. The path loss model is set as $L[\text{dB}] = 128.1 + 37.6\log_{10} d_{[\text{km}]}$, and the standard deviation of the log-normal shadowing fading is 8 dB [42]. We also set the bandwidth $B = 400$ KHz and the background noise



(a) (b)

Fig. 4. Comparison among different algorithms as the latency budget $\tau^{\max}$ varies. (a) the average cosine similarity $\bar{C}$. (b) the average device size $\bar{S}$.



(a) (b)

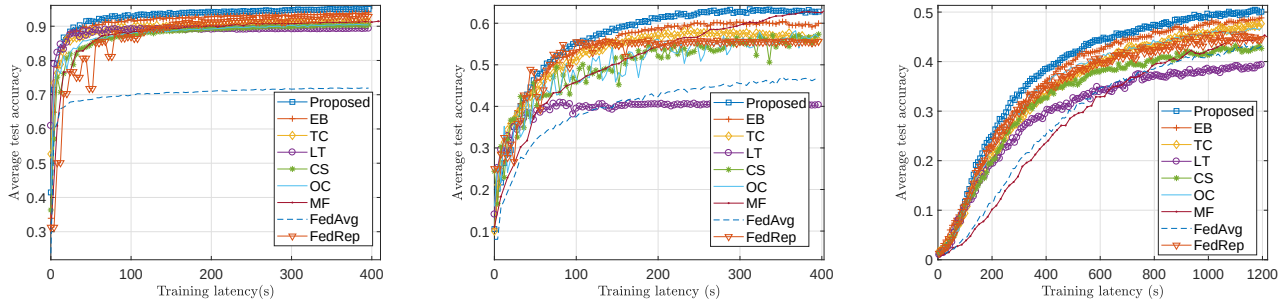
Fig. 5. Comparison among different algorithms as the cluster number budget $|\Pi|^{\max}$ varies. (a) the average cosine similarity $\bar{C}$. (b) the average cluster size $\bar{S}$.

$N_0 = -107$ dBm. For each device k, the parameters of the computation latency model are set as $\chi_k = 0.1$ ms/samples and $\varsigma_k = \frac{1}{\chi_k}$, respectively [21]. We set an equal maximum transmit power $p_1^{\max} = p_2^{\max} =, ..., = p_K^{\max} = 10$ dBm, an equal energy consumption of local training $e_1^{\mathrm{c}} = e_2^{\mathrm{c}}, ..., = e_K^{\mathrm{c}} = 0.4$ J, and an equal energy consumption budget $e_1^{\max} = e_2^{\max}, ..., = e_K^{\max} = 0.5$ J. The training latency budget is $\tau^{\max} = 4$ s, the maximal number of the bandwidth occupying clusters is $|\Pi|^{\max} = 70$, and the weight parameter is $\rho = 0.99995$.

2) Learning Settings: To evaluate the performance of the proposed algorithm, we consider a classification task using the MNIST dataset with 60,000 training samples and 10,000 testing samples of 10 types of digits [43]. A standard multilayer perceptron model is adopted for training, which has one hidden layer of 128 hidden nodes and finally a output layer. The batch size is 32, the optimizer is SGD, and the learning rate is 0.05. To simulate an incongruent data distribution, each device has 1000 training samples with 10 types of labels, in which 200 training samples with 2 types of labels can be randomly selected and modified by swapping [25]. A sketch map of the incongruent data distribution is shown in Fig.3. For device 1, the data samples "0" and "1" are labeled as "1000000000" and "0100000000", respectively. However, for device 2, the data samples "0" and "1" are re-labeled as "0100000000" and "1000000000". Similarly, the labels of the same data samples in device 3 and device 4 can be modified by swapping.

It is worth noting that the similar operations as the training samples are done on the test samples of each device. Besides, to investigate the test accuracy on the individual data distribution, different from the settings in the conventional FL, the test samples of each device in the considered CFL is with the same assignment labels of the training samples [25]. The experimental data setup are summarized in Table III.

3) Baseline Settings: Five baseline algorithms are introduced for comparison as follows. This first one (labelled as **EB** algorithm) is introduced where Algorithm 3 is adopted for device clustering and equal bandwidth allocation is adopted. The second one (labelled as **TC** algorithm) divides the devices with congruent data distributions into the same cluster [29]. The third one (labelled as **LT** algorithm) assumes all devices



(a) (b) (c)

Fig. 6. Comparison of the average test accuracy among different algorithms. (a) the average test accuracy under the MNIST dataset. (b) the average test accuracy under the CIFAR-10 dataset. (c) the average test accuracy under the CIFAR-100 dataset.

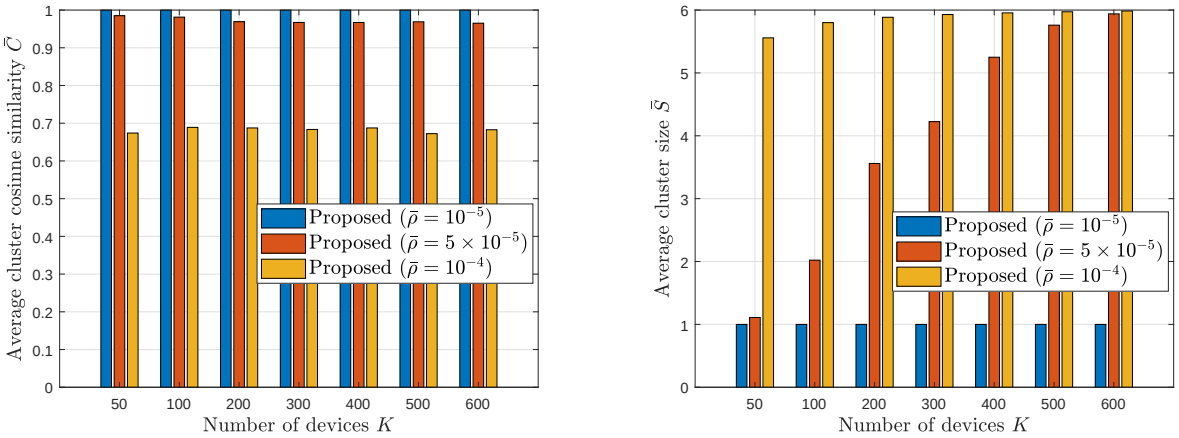
train their clustered models without cooperation. The fourth one (labelled as **CS** algorithm) assumes 400 newcome devices are added to the existing clusters [26]. The fifth one (labelled as **OC** algorithm) assumes each device can be repeatedly added into multiple clusters [28].

B. Comparison of Different Algorithms

In Fig. 4, we first compare the performance of the proposed algorithm with the baseline algorithms as the latency budget $\tau^{\max}$ varies. Fig. 4(a) and 4(b) show the trends of the average cosine similarity $\bar{C} = \frac{\sum_{s\in\mathcal{S}} \bar{A}_{1,s}}{|\Pi|}$ and the average cluster size $\bar{S} = \frac{\sum_{s\in\mathcal{S}} \bar{A}_{2,s}}{|\Pi| \sum_{k\in\mathcal{K}} D_k}$, respectively. As shown in Fig. 4(a), the average cosine similarity of the proposed algorithm decreases as the latency budget increases. This is because given more time for training, devices prefer to collaborate with others, thus increase the variability of data distribution. It is interesting to find that the cosine similarity obtained by the proposed algorithm and the EB algorithm is almost the same and is closed to that of the TC algorithm. This fact suggests that the device clustering is mainly based on the data distribution. In Fig. 4(b), we can observe that the average cluster size of the proposed algorithm increases with more latency budget due to the fact that a larger latency budget enables each cluster to include more devices and increase the participation probability. Specially, given a smaller latency budget, i.e, $\tau^{\max} \leq 2$, the average cluster size is equal to 1 due to the limited time reserved for local training and model transmission. Besides, as the latency budget increases, i.e, $\tau^{\max} \geq 5$, the performance of the TC and MT algorithm is close to that of the proposed algorithm. This is because under a larger latency budget, the proposed algorithm can seek to let all devices with congruent data distribution be grouped into the same clusters. Besides, although the average cosine similarity of the proposed algorithm is lower than that of the LT, CS and OC algorithms, a larger average cluster size can be obtained. In general, the proposed algorithm can achieve larger cluster size than the benchmarks. This is because the proposed algorithm jointly optimizes the resources and the device clustering, while the bandwidth is fixed in the EB algorithm, and the number of devices per cluster is not optimized in the other baselines. In addition, we investigate the percentage of the same distribution devices per cluster. For instance, in cluster s, if the data distribution of device $k \in \mathcal{V}_s$ is congruent with device $k' \in \mathcal{V}_s$, we can derive $d_{k,k'} = 1$, and $d_{k,k'} = 0$ otherwise. Then the percentage of the same distribution devices in cluster s can be defined as $z_s = \frac{\sum_{i,j\in\mathcal{V}_s, i\neq j} d_{i,j}}{|\mathcal{V}_s|(|\mathcal{V}_s|-1)}$. Specially, in the proposed algorithm, we can obtain $z_s = 1, \forall s \in \mathcal{S}$, which can prove that the devices with consistent data distribution are divided into the same cluster.

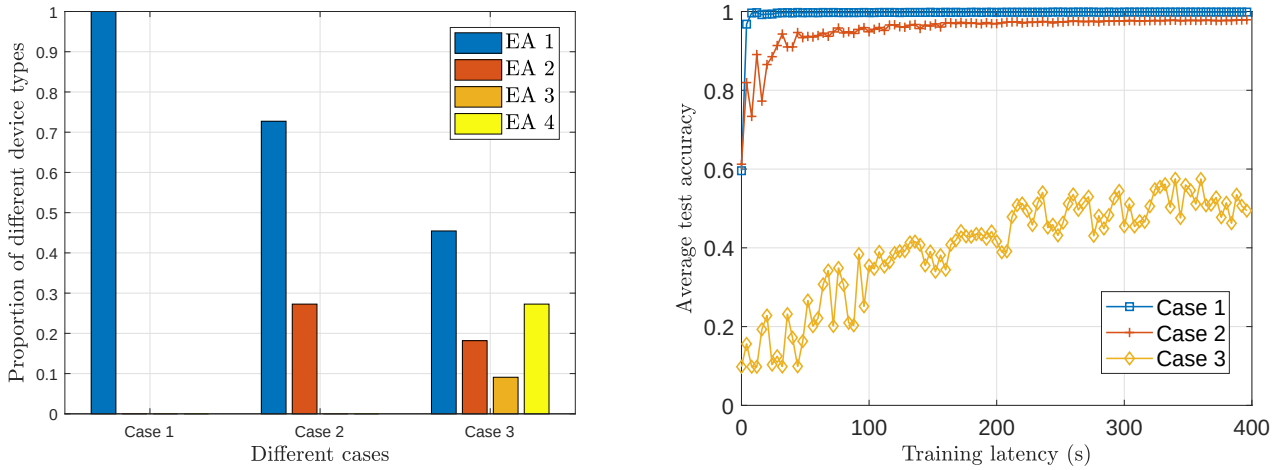
Fig. 5(a) and 5(b) show the trends of the average cosine similarity $\bar{C}$ and the average cluster size $\bar{S}$ as the cluster number budget $|\Pi|^{\max}$ varies. We can see that both the average cosine similarity and the average cluster size of the proposed algorithm increase as the cluster number budget increases, then remain unchanged. This suggests that the utility obtained by local training, i.e., devices in the LT algorithm, is lower than the cooperative training, and the devices prefer to formulate the clusters with other devices. Although the proposed algorithm outperforms the benchmarks, the advantages are not significant especially under a smaller number budget. This is because given a smaller cluster number budget, i.e., $|\Pi|^{\max} \leq 50$, the cosine similarity and the cluster size of devices with local training play a major role in the calculation of the average values thus the benefits derived from the cluster formulation are not significant.

In Fig. 6, we compare the test accuracy of the proposed algorithm with baseline algorithms. For generality, aside from the MNIST dataset, the CIFAR-10 dataset and the CIFAR-100 dataset are also considered for training [44]. CIFAR-10 has 50,000 training samples and 10,000 testing samples of 10 types of images, and CIFAR-100 has 50,000 training samples and 10,000 testing samples of 100 types of images. A CNN model is utilized for CIFAR-10, which has two 3x3 convolution layers with respective 32 and 32 channels, also followed with 2x2 max pooling, two fully connected layers with 384 and 192 units, and finally a softmax output layer. The batch size is 32, the optimizer is SGD, and the learning rate is set to 0.01. ResNet-18 is utilized for CIFAR-100 [45], which uses 3×3 filters with stride and pad of 1, and the average pooling layer contains 1×1 filter, and one fully connected layer, with a final softmax layer. The batch size is 64, the optimizer is Adam, and the learning rate is set to 0.02. Besides, to further demonstrate the advantages of the proposed algorithm, we add extra three baseline algorithms for



(a) (b)

Fig. 7. Performance of the proposed algorithm under different weight parameter $\bar{\rho}$ and device number K. (a) the average cosine similarity $\bar{C}$. (b) the average cluster size $\bar{S}$.



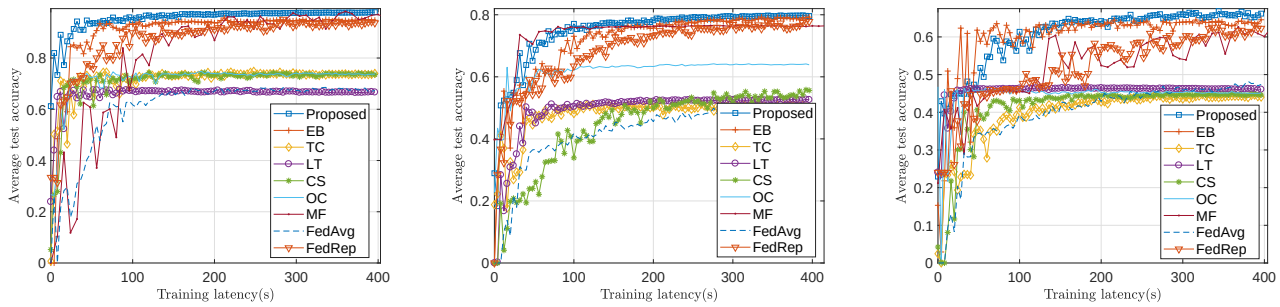
(a) (b)

Fig. 8. Comparison of the average test accuracy among different algorithms with the default setting. (a) Average clustering accuracy. (b) Average test accuracy.

comparison. The first one (labelled as **MF** algorithm) assumes the full participation of devices, in which the latency budget is increased to let all device participate in training successfully. The second one is the conventional **FedAvg** algorithm, in which all devices form a grand cluster, and all bandwidth resources in the system can be utilized for training. The third one is **FedRep** algorithm [35], in which devices are selected to perform personalized local gradient-based updates to solve for its optimal model head, and once the local updates with respect to the head finish, the device participates in the aggregation by sending its locally-updated representation to the server. Results shows that the proposed algorithm can achieve higher test accuracy with lower training latency compared to the baseline algorithms on the both two datasets. We can also observe that even though the convolutional layers in CIFAR-10 and CIFAR-100 are not convex, the proposed algorithm can also improve the learning performance. The LT and the FedAvg algorithms have the worst performing, because they do not jointly consider the convergence performance and the generalization ability of training. Besides, compared to the EB, TC, CS, OC, and MF algorithms, the proposed algorithm achieves a balance between the cosine similarity and the cluster size, thus improving the training efficiency. In summary, the advantage of the proposed algorithm is twofold: firstly, the proposed algorithm addresses the issue of statistical heterogeneity by grouping devices with congruent data distribution into the same clusters. Besides, with limited wireless resources, the number of devices per cluster and the participation probability are optimized simultaneously to improve the generalization ability of the trained clustered models. Moreover, as shown in Table IV, to have a more intuitive understanding of the tradeoff between the average cosine similarity and the average cluster size, we present a table under default parameters and MNIST dataset. Considering the limited bandwidth in each cluster and the differences of data distribution among devices, when we increase the number of

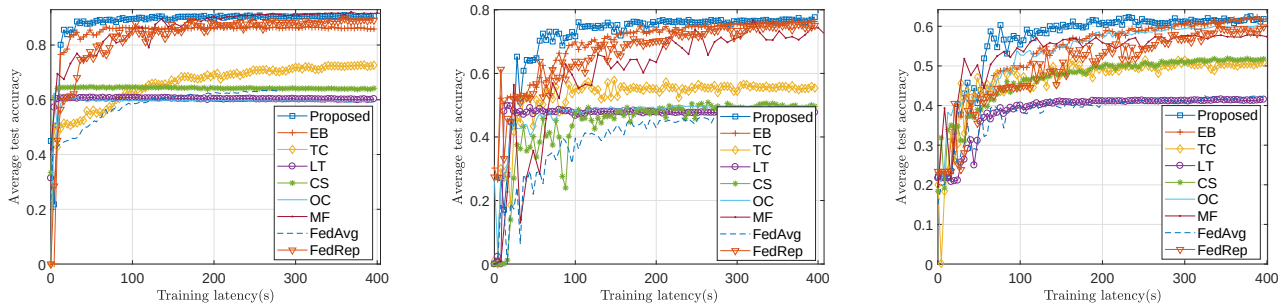
TABLE IV
ILLUSTRATE THE TRADEOFF BETWEEN THE AVERAGE COSINE SIMILARITY AND THE AVERAGE CLUSTER SIZE.

algorithm	average cosine similarity	average cluster size	maximal test accuracy
Proposed	0.9673	5.9400	0.9505
EB	0.9673	5.5440	0.9391
TC	0.9668	3.1920	0.9127
LT	1	1	0.8916
CS	0.9947	1.7859	0.9040
OC	0.9964	1.5403	0.8937
FedAvg	0.5842	600	0.7193



(a) (b) (c)

Fig. 9. Comparison of the average test accuracy among different algorithms under a more general data partition method without label swapping. (a) the average test accuracy under the MNIST dataset. (b) the average test accuracy under the CIFAR-10 dataset. (c) the average test accuracy under the CIFAR-100 dataset.



(a) (b) (c)

Fig. 10. Comparison of the average test accuracy among different algorithms under a more general data partition method with label swapping. (a) the average test accuracy under the MNIST dataset. (b) the average test accuracy under the CIFAR-10 dataset. (c) the average test accuracy under the CIFAR-100 dataset.

devices in the cluster, the similarity will inevitably be affected. The most intuitive case is to compare the LT algorithm with the proposed algorithm. Although the LT algorithm can obtain a similarity of 1, the number of devices in a single cluster is only one, so the learning performance obtained is lower than the proposed algorithm. A similar phenomenon can also be seen in the proposed algorithm and other comparison algorithms. For instance, although the proposed algorithm has lower cosine similarity than the LT, CS, and OC algorithms, the proposed algorithm can achieve larger cluster size with higher learning performance. However, in a extreme case, the FedAvg algorithm has the largest cluster size with the minimal similarity, and the learning performance is lower than the proposed algorithm. Hence, we can clearly see that the tradeoff between the average cosine similarity and the average cluster size does exist.

C. Effects of the weighted parameter

In Fig. 7, to have a deeper understanding of how the proposed algorithm works, we further investigate performance of the proposed algorithm as the weight parameter $\bar{\rho} = 1 - \rho$ and the device number K vary. In particular, Figs. 7(a)-7(b) show the trends of average cluster cosine similarity $\bar{C}$ and the average cluster size $\bar{S}$, respectively. In Fig. 7(a), we can observe that the average cluster cosine similarity of the proposed algorithm decreases as the weight parameter $\bar{\rho}$ increases. It is because increasing the weight of the generalization ability enables the devices to achieve higher utility by increasing the cluster size. In particular, given a smaller weight parameter $\bar{\rho}$, the proposed algorithm emphasizes the cosine similarity among the devices and ensures the devices with similar data distribution to be grouped into the same cluster. On the contrary, given a larger weight parameter $\bar{\rho}$, since the

effects of the cosine similarity are almost negligible and the devices focus on the number of devices per cluster and the participation probability, the average cluster cosine similarity $\bar{C}$, is almost random. In Fig. 7(b), results show that the average cluster size $\bar{S}$ of the proposed algorithm is increased and trends to be unchanged as $\bar{\rho}$ or K increases. It is because the utility can be improved by including more devices or increasing the participation probability of the existing devices. This observation is also aligned with the previous discussions of CFL.

D. Performance under different data partition methods

Our proposed algorithm, while showing higher learning performance, is based on a data distribution in [25] that considered the swapping of labels. Hence, it is important to design a more generalized data partitioning method by overlapping the labels among the devices to demonstrate the validity and generality of the proposed clustering method. We jointly consider and advance the data partitioning methods used in references [26, 28] where each device has only a subset of labels, which from different distribution types overlap without any label swapping. We can observe that there are significant differences in cosine similarity among the devices for the following four examples. 1) devices with the same data distributions. 2) data distributions with the subset relationship. 3) data distributions have disjointed parts. 4) data distributions are un-intersected. According to the clustering results, in each cluster s, we first select $|\tilde{\mathcal{V}}_s^{\mathrm{c1}}|$ devices which have label distributions with the most occurrences, as a benchmark. We further denote the number of remaining devices which belong to example 2, example 3, and example 4 as $|\tilde{\mathcal{V}}_s^{\mathrm{c2}}|$, $|\tilde{\mathcal{V}}_s^{\mathrm{c3}}|$, and $|\tilde{\mathcal{V}}_s^{\mathrm{c4}}|$, respectively, then the accuracy of clustering is defined as the proportion in each example, i.e., EA1 $= \frac{|\tilde{\mathcal{V}}_s^{\mathrm{c1}}|}{|\mathcal{V}_s|}$, EA2 $= \frac{|\tilde{\mathcal{V}}_s^{\mathrm{c2}}|}{|\mathcal{V}_s|}$, EA3 $= \frac{|\tilde{\mathcal{V}}_s^{\mathrm{c3}}|}{|\mathcal{V}_s|}$, EA4 $= \frac{|\tilde{\mathcal{V}}_s^{\mathrm{c4}}|}{|\mathcal{V}_s|}$. In addition, according to the previous simulation results, we notice that the weighted parameter in the optimization problem is critical to the clustering results. Hence, given the generalizable data partitioning, we evaluate the weighted parameter $\bar{\rho}$ for the three datasets, which can be divided into three cases. Case 1: a smaller weight parameter $10^{-6} < \bar{\rho} < 2 \times 10^{-5}$. Case 2: a moderate weight parameter $4 \times 10^{-5} < \bar{\rho} < 7 \times 10^{-5}$. Case 3: a larger weight parameter $9 \times 10^{-5} < \bar{\rho} < 2 \times 10^{-4}$. As shown in Fig. 8(a), we can notice that with a smaller or moderate weight parameter, all devices with inconsistent data distributions can be well distinguished. Besides, in Fig. 8(b), based on the MNIST dataset, the test accuracy is influenced by the clustering results, and setting a proper weight parameter can achieve higher test accuracy. In addition, as shown in Fig. 9 and Fig. 10, we further add experiments of all baselines under the proposed general data partition method with and without label swapping, and we adapt a moderate weight parameter for training. Similar to Fig. 6, our proposed algorithm can achieve higher test accuracy with lower training latency. Specially, it is worth nothing that we can achieve higher accuracy compared with Fig.6 under the case that each device has only a subset of label types. This is because the classifier is less difficult to train when there are fewer label types. In addition, for the proposed method, the influence of label swapping is not significant, because our method does not group devices with the conflicting data distribution into the same cluster. Therefore, for each cluster, devices with similar data distributions still participate in the same task of training.

V. Conclusion

In this paper, the convergence analysis of CFL in terms of the cosine similarity, the number of devices per cluster, and the device participation probability was conducted. Besides, based on the obtained analysis results, an optimization problem incorporating the bandwidth allocation, the transmit power control, and the device clustering was proposed, aiming at maximizing the learning performance of the CFL. The joint problem was decoupled into two sub-problems and solved iteratively. In particularly, given fixed results of device clustering, an iterative algorithm based on the convex optimization theory was proposed for bandwidth allocation and transmit power control. Besides according to the individual stability, we developed a distributed coalition formation algorithm for device clustering. The simulation results have shown that the proposed algorithm, compared to state of the art benchmarks, can achieve higher test accuracy with limited wireless resources.

Appendix

A. Proof of *Theorem 1*

Based on the second-order Taylor expansion of $\bar{F}_s(\boldsymbol{w}_s^{(r+1)})$ we can derive that

$$\begin{aligned}
&\bar{F}_s(\boldsymbol{w}_s^{(r+1)}) = \bar{F}_s(\boldsymbol{w}_s^{(r)}) + (\boldsymbol{w}_s^{(r+1)} - \boldsymbol{w}_s^{(r)})^{\mathrm{T}} \nabla \bar{F}_s(\boldsymbol{w}_s^{(r)}) \\
&+ \frac{1}{2}(\boldsymbol{w}_s^{(r+1)} - \boldsymbol{w}_s^{(r)})^{\mathrm{T}} \nabla^2 \bar{F}_s(\boldsymbol{w}_s^{(r)})(\boldsymbol{w}_s^{(r+1)} - \boldsymbol{w}_s^{(r)}) \\
&\leq \bar{F}_s(\boldsymbol{w}_s^{(r)}) + (\boldsymbol{w}_s^{(r+1)} - \boldsymbol{w}_s^{(r)})^{\mathrm{T}} \nabla \bar{F}_s(\boldsymbol{w}_s^{(r)}) \\
&+ \frac{L}{2}\|\boldsymbol{w}_s^{(r+1)} - \boldsymbol{w}_s^{(r)}\|^2. \qquad (28)
\end{aligned}$$

Given the learning rate $\eta = \frac{1}{L}$, the expected loss $\mathbb{E}(\bar{F}_s(\boldsymbol{w}_s^{(r+1)}))$ can be expressed as

$$\begin{aligned}
&\mathbb{E}(\bar{F}_s(\boldsymbol{w}_s^{(r+1)})) \leq \mathbb{E}(\bar{F}_s(\boldsymbol{w}_s^{(r)})) - \frac{1}{2L}\|\nabla \bar{F}_s(\boldsymbol{w}_s^{(r)})\|^2 \\
&+ \frac{1}{2L}\mathbb{E}\left\|\nabla \bar{F}_s(\boldsymbol{w}_s^{(r)}) - \frac{\sum_{k\in\mathcal{K}} a_{k,s} q_k^{(r)} D_k \nabla F_k(\boldsymbol{w}_s^{(r)})}{\sum_{k\in\mathcal{K}} a_{k,s} q_k^{(r)} D_k}\right\|^2. \qquad (29)
\end{aligned}$$

From the perspective of cosine similarity, we have

$$\begin{aligned}
&\left\|\nabla \bar{F}_s(\boldsymbol{w}_s^{(r)}) - \frac{\sum_{k\in\mathcal{K}} a_{k,s} q_k^{(r)} D_k \nabla F_k(\boldsymbol{w}_s^{(r)})}{\sum_{k\in\mathcal{K}} a_{k,s} q_k^{(r)} D_k}\right\|^2 = \\
&\left\|\frac{\sum_{k\in\mathcal{K}} a_{k,s} q_k^{(r)} D_k (\nabla \bar{F}_s(\boldsymbol{w}_s^{(r)}) - \nabla F_k(\boldsymbol{w}_s^{(r)}))}{\sum_{k\in\mathcal{K}} a_{k,s} q_k^{(r)} D_k}\right\|^2 = \\
&\left\|\frac{\sum_{k\in\mathcal{K}} a_{k,s} q_k^{(r)} D_k \sum_{k'\in\mathcal{K}} a_{k',s} D_{k'} (\nabla F_{k'}(\boldsymbol{w}_s^{(r)}) - \nabla F_k(\boldsymbol{w}_s^{(r)}))}{\sum_{k\in\mathcal{K}} a_{k,s} q_k^{(r)} D_k \sum_{k'\in\mathcal{K}} a_{k',s} D_{k'}}\right\|^2. \qquad (30)
\end{aligned}$$

Let

$$C_{k,k'}(\boldsymbol{w}_s^{(r)}) = \frac{< \nabla F_k(\boldsymbol{w}_s^{(r)}), \nabla F_{k'}(\boldsymbol{w}_s^{(r)}) >}{\|\nabla F_k(\boldsymbol{w}_s^{(r)})\| \|\nabla F_{k'}(\boldsymbol{w}_s^{(r)})\|}, \qquad (31)$$

then we have

$$\begin{aligned}&\|\nabla F_{k'}(\boldsymbol{w}_s^{(r)})-\nabla F_k(\boldsymbol{w}_s^{(r)})\|^2\\&=\|\nabla F_{k'}(\boldsymbol{w}_s^{(r)})\|^2+\|\nabla F_k(\boldsymbol{w}_s^{(r)})\|^2\\&-2\left\langle\nabla F_{k'}(\boldsymbol{w}_s^{(r)}),\nabla F_k(\boldsymbol{w}_s^{(r)})\right\rangle\\&\leq 2\xi_2^2-2C_{k,k'}(\boldsymbol{w}_s^{(r)})\xi_1^2,\end{aligned}\tag{32}$$

and

$$\begin{aligned}&\left\|\nabla\bar{F}_s(\boldsymbol{w}_s^{(r)})-\frac{\sum_{k\in\mathcal{K}}a_{k,s}q_k^{(r)}D_k\nabla F_k(\boldsymbol{w}_s^{(r)})}{\sum_{k\in\mathcal{K}}a_{k,s}q_k^{(r)}D_k}\right\|^2\leq\\&\left\|\frac{\sum_{k\in\mathcal{K}}a_{k,s}D_k\sum_{k'\in\mathcal{K}}a_{k',s}D_{k'}\sqrt{2\xi_2^2-2C_{k,k'}(\boldsymbol{w}_s^{(r)})\xi_1^2}}{\sum_{k\in\mathcal{K}}a_{k,s}q_k^{(r)}D_k\sum_{k'\in\mathcal{K}}a_{k',s}D_{k'}}\right\|^2.\end{aligned}\tag{33}$$

By substituting (33) into (29), we can obtain

$$\begin{aligned}&\mathbb{E}(\bar{F}_s(\boldsymbol{w}_s^{(r+1)}))\leq\mathbb{E}(\bar{F}_s(\boldsymbol{w}_s^{(r)}))-\frac{1}{2L}\|\nabla\bar{F}_s(\boldsymbol{w}_s^{(r)})\|^2\\&+\underbrace{\mathbb{E}\left\|\frac{\sum_{k\in\mathcal{K}}a_{k,s}D_k\sum_{k'\in\mathcal{K}}a_{k',s}D_{k'}\sqrt{2\xi_2^2-2C_{k,k'}(\boldsymbol{w}_s^{(r)})\xi_1^2}}{\sum_{k\in\mathcal{K}}a_{k,s}q_kD_k\sum_{k'\in\mathcal{K}}a_{k',s}D_{k'}}\right\|^2}_{\left(A_{1,s}^{(r)}\right)^2}.\end{aligned}\tag{34}$$

According to assumptions 2 and 3, we further have [46]

$$\|\nabla\bar{F}_s(\boldsymbol{w}_s^{(r)})\|^2\geq 2\mu\left(\bar{F}_s(\boldsymbol{w}_s^{(r)})-\bar{F}_s(\boldsymbol{w}_s^*)\right),\tag{35}$$

where $\boldsymbol{w}_s^*$ is the optimal clustered model of cluster s. Substituting (35) into (34), we have

$$\begin{aligned}&\mathbb{E}\left(\bar{F}_s(\boldsymbol{w}_s^{(r+1)})-\bar{F}_s(\boldsymbol{w}_s^*)\right)\\&\leq\left(1-\frac{\mu}{L}\right)\mathbb{E}\left(\bar{F}_s(\boldsymbol{w}_s^{(r)})-\bar{F}_s(\boldsymbol{w}_s^*)\right)+\left(A_{1,s}^{(r)}\right)^2.\end{aligned}\tag{36}$$

Let $A_{1,s}=\max_r A_{1,s}^{(r)}$. Then, applying (36) recursively, we can complete the proof.

B. Proof of ***Theorem 2***

From the perspective of participation probability, we have [20]

$$\begin{aligned}&\left\|\nabla F(\boldsymbol{w}_s^{(r)})-\frac{\sum_{k\in\mathcal{K}}a_{k,s}q_k^{(r)}D_k\nabla F_k(\boldsymbol{w}_s^{(r)})}{\sum_{k\in\mathcal{K}}a_{k,s}q_k^{(r)}D_k}\right\|^2\\&\leq\underbrace{\frac{4\xi_2^2\left(\sum_{k\in\mathcal{K}}D_k-\sum_{k\in\mathcal{K}}a_{k,s}q_k^{(r)}D_k\right)}{\sum_{k\in\mathcal{K}}D_k}}_{A_{2,s}^{(r)}}.\end{aligned}\tag{37}$$

Then, substituting (37) into (29), and applying recursively, we can complete the proof.

C. Proof of ***Theorem 4***

Given device set $\mathcal{V}_s$, since the objective function of problem (19) is convex, we can solve it by using the Karush-Kuhn-Tucker (KKT) conditions, and the corresponding Lagrange function is

$$\begin{aligned}&\mathcal{L}(\mathcal{B},\nu)=\nu_s\left(\sum_{k\in\mathcal{V}_s}b_k-B\right)-\\&(1-\rho)\sum_{k\in\mathcal{V}_s}D_k\left(1-e^{-\frac{\varsigma_k}{D_k}\left(\tau^{\max}-\frac{M}{b_k\log_2(1+\frac{|h_k|^2p_k}{N_0})}-D_k\chi_k\right)}\right),\end{aligned}\tag{38}$$

where $\nu_s>0$ is the Largrange multiplier. The first order of (38) with respect to b_k, $\forall k\in\mathcal{V}_s$, is

$$\begin{aligned}&\frac{\partial\mathcal{L}(\mathcal{B},\nu_s)}{\partial b_k}=\nu_s\\&-\frac{(1-\rho)a_{k,s}\varsigma_kM}{b_k^2\log_2(1+\frac{|h_k|^2p_k}{N_0})}e^{-\frac{\varsigma_k}{D_k}\left(\tau^{\max}-\frac{M}{b_k\log_2(1+\frac{|h_k|^2p_k}{N_0})}-D_k\chi_k\right)}.\end{aligned}\tag{39}$$

By solving (39), we can obtain the optimal solution b_k^*. Besides, increasing the transmit power p_k can also increase the objective function of (19), thus we can choose the maximal p_k that satisfies the energy consumption budget. Then we complete the proof.

References

[1] W. Xia, B. Xu, H. Zhao, Y. Zhu, X. Sun, and T. Q. Quek, "Optimization of clustering strategy and resource allocation for clustered federated learning," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Rio de Janeiro, Brazil, Dec 2022, pp. 3077–3082.

[2] "Cisco global cloud index: Forecast and methodology." [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/ service-provider/global-cloud-index-gci/white-paper-c11-738085.html

[3] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. Cambridge, MA: MIT Press, 2016.

[4] H. Cao, S. Wu, G. S. Aujla, Q. Wang, L. Yang, and H. Zhu, "Dynamic embedding and quality of service-driven adjustment for cloud networks," *IEEE Trans. Indus. Infor.*, vol. 16, no. 2, pp. 1406–1416, Feb. 2019.

[5] W. Xia, G. Zheng, Y. Zhu, J. Zhang, J. Wang, and A. P. Petropulu, "A deep learning framework for optimization of MISO downlink beamforming," *IEEE Trans. Commun.*, vol. 68, no. 3, pp. 1866–1880, Mar. 2020.

[6] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS)*, Fort Lauderdale, FL, USA, Apr. 2017, pp. 1–10.

[7] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-iid data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3400–3413, Nov. 2020.

[8] U. Michelucci, *Applied Deep Learning*. Springer, 2018.

[9] J. Sun, T. Chen, G. B. Giannakis, Q. Yang, and Z. Yang, "Lazily aggregated quantized gradient innovation for communication-efficient federated learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 4, pp. 2031–2044, Apr. 2022.

[10] T. Chen, Y. Sun, and W. Yin, "Communication-adaptive stochastic gradients for distributed learning," *IEEE Trans. Signal Process.*, vol. 69, no. 1, pp. 4637–4651, Jul. 2021.

[11] M. M. Amiri, D. Gunduz, S. R. Kulkarni, and H. V. Poor, "Update aware device scheduling for federated learning at the wireless edge," in *Proc. IEEE Int. Symp. Inf. Theor. Proc. (ISIT)*, Los Angeles, CA, USA, Jun. 2020, pp. 2598–2603.

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

[13] 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.

[14] V.-D. Nguyen, S. K. Sharma, T. X. Vu, S. Chatzinotas, and B. Ottersten, "Efficient federated learning algorithm for resource allocation in wireless iot networks," *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2276–2288, Mar. 2021.

[15] J. Yao and N. Ansari, "Enhancing federated learning in fog-aided iot by CPU frequency and wireless power control," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3438–3445, Mar. 2021.

[16] C. T. Dinh, N. H. Tran, M. N. Nguyen, C. S. Hong, W. Bao, A. Y. Zomaya, and V. Gramoli, "Federated learning over wireless networks: Convergence analysis and resource allocation," *IEEE/ACM Trans. Networking.*, vol. 29, no. 1, pp. 398–409, Feb. 2020.

[17] H. H. Yang, Z. Liu, T. Q. Quek, and H. V. Poor, "Scheduling policies for federated learning in wireless networks," *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 317–333, Jan 2019.

[18] 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, Jul. 2020.

[19] B. Xu, W. Xia, J. Zhang, T. Q. Quek, and H. Zhu, "Online client scheduling for fast federated learning," *IEEE Wireless Commun. Lett.*, vol. 10, no. 7, pp. 1434–1438, Jul. 2020.

[20] 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. Wirel. Commun.*, vol. 20, no. 1, pp. 269–283, Jan. 2021.

[21] W. Shi, S. Zhou, Z. Niu, M. Jiang, and L. Geng, "Joint device scheduling and resource allocation for latency constrained wireless federated learning," *IEEE Wirel. Commun.*, vol. 20, no. 1, pp. 453–467, Sep. 2020.

[22] Y. He, J. Ren, G. Yu, and J. Yuan, "Importance-aware data selection and resource allocation in federated edge learning system," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 13 593–13 605, Nov. 2020.

[23] H. Cao, S. Garg, G. Kaddoum, S. Singh, and M. S. Hossain, "Softwarized resource management and allocation with autonomous awareness for 6g-enabled cooperative intelligent transportation systems," *IEEE Trans. Intell. Trans. Syst.*, vol. 23, no. 12, pp. 24 662–24 671, Dec. 2022.

[24] S. Pan, J. Wu, X. Zhu, C. Zhang, and P. S. Yu, "Joint structure feature exploration and regularization for multi-task graph classification," *IEEE Trans. Knowl. Data. Eng.*, vol. 28, no. 3, pp. 715–728, Mar. 2016.

[25] F. Sattler, K.-R. Mller, and W. Samek, "Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 8, pp. 3710–3722, Aug. 2020.

[26] M. Duan, D. Liu, X. Ji, Y. Wu, L. Liang, X. Chen, and Y. Tan, "Flexible clustered federated learning for client-level data distribution shift," *IEEE Trans. Parallel Distrib Syst.*, vol. 33, no. 11, pp. 2661–2674, Dec. 2022.

[27] G. Long, M. Xie, T. Shen, T. Zhou, X. Wang, and J. Jiang, "Multi-center federated learning: Clients clustering for better personalization," *WWW*, vol. 26, no. 1, pp. 481–500, Jan. 2023.

[28] C. Li, G. Li, and P. K. Varshney, "Federated learning with soft clustering," in *IEEE Internet Things J.*, vol. 9, no. 15. IEEE, May 2021, pp. 7773–7782.

[29] A. Ghosh, J. Chung, D. Yin, and K. Ramchandran, "An efficient framework for clustered federated learning," in *Proc. 34th Adv. neural inf. proces. syst. (NIPS)*, Virtual Conference, Dec. 2020, pp. 1–6.

[30] X.-X. Wei and H. Huang, "Edge devices clustering for federated visual classification: A feature norm based framework," *IEEE Trans. Image Process*, vol. 32, pp. 995–1010, Jan. 2023.

[31] R. Lu, W. Zhang, Y. Wang, Q. Li, X. Zhong, H. Yang, and D. Wang, "Auction-based cluster federated learning in mobile edge computing systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 34, no. 4, pp. 1145–1158, Apr. 2023.

[32] Q.-V. Pham, H. T. Nguyen, Z. Han, and W.-J. Hwang, "Coalitional games for computation offloading in noma-enabled multi-access edge computing," *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 1982–1993, Nov. 2019.

[33] R. Mochaourab, E. Björnson, and M. Bengtsson, "Adaptive pilot clustering in heterogeneous massive mimo networks," *IEEE Trans. Wirel. Commun.*, vol. 15, no. 8, pp. 5555–5568, Aug. 2016.

[34] A. Bogomolnaia and M. O. Jackson, "The stability of hedonic coalition structures," *Game. Economi. Behav.*, vol. 38, no. 2, pp. 201–230, 2002.

[35] L. Collins, H. Hassani, A. Mokhtari, and S. Shakkottai, "Exploiting shared representations for personalized federated learning," in *PMLR*, 2021, pp. 2089–2099.

[36] Y. Pan, C. Pan, Z. Yang, and M. Chen, "Resource allocation for d2d communications underlaying a noma-based cellular network," *IEEE Wireless Commun. Lett.*, vol. 7, no. 1, pp. 130–133, Feb. 2018.

[37] 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, Mar. 2019.

[38] D. Liu and O. Simeone, "Privacy for free: Wireless federated learning via uncoded transmission with adaptive power control," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 170–185, Nov. 2020.

[39] J. Ren, G. Yu, and G. Ding, "Accelerating DNN training in wireless federated edge learning system," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 219–232, Nov. 2020.

[40] D. Pisinger, "An exact algorithm for large multiple knapsack problems," *Eur. J. Oper. Res.*, vol. 114, no. 3, pp. 528–541, 1999.

[41] R. Mochaourab and E. A. Jorswieck, "Coalitional games in miso interference channels: Epsilon-core and coalition structure stable set," *IEEE Trans. Signal Process.*, vol. 62, no. 24, pp. 6507–6520, Dec. 2014.

[42] T. X. Tran and D. Pompili, "Joint task offloading and resource allocation for multi-server mobile-edge computing networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 856–868, Nov. 2018.

[43] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE.*, vol. 86, no. 11, pp. 2278–2324, 1998.

[44] A. Krizhevsky, V. Nair, and G. Hinton, "The cifar-10 dataset," 2014. [Online]. Available: http://www. cs. toronto. edu/kriz/cifar. html

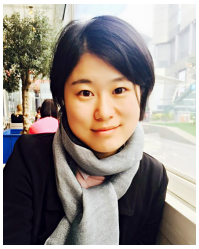
[45] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE CVPR*, Las Vegas, NV, USA, Jun 2016, pp. 770–778.

[46] B. Stephen and V. Lieven, *Convex Optimization*. Cambridge, MA: MIT Press, 2004.

Bo Xu received his B.S. degree in communication engineering and Ph.D. degree in communication and information systems from Nanjing University of Posts and Telecommunications in 2018 and 2022. 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.

Wenchao Xia (Member, IEEE) received his B.S. degree in communication engineering and Ph.D. degree in communication and information systems from Nanjing University of Posts and Telecommunications in 2014 and 2019, respectively. From 2019 to 2020, he was a postdoctoral research fellow at Singapore University of Technology and Design. 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 current research interests include edge intelligence, edge computing, cloud radio access networks, and massive MIMO. He was a recipient of the Best Paper Award at IEEE GLOBECOM 2016.

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 in 2008 and 2011, respectively. Currently, he is a professor of the School of Communication and Information Engineering in Nanjing University of Posts and Telecommunications. From May 2019 to November 2019, he was a Visiting Scholar with the Department of Electronic and lnformation Engineering, The Hong Kong Polytechnic University, Hong Kong China. His current research interests include wireless multi-media 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 lnvention Award of the Chinese Communication Society in 2017 and 2019.



Yongxu Zhu (PI, IEEE Senior Member) received her Ph.D. degree at the University College London (U.K.) in 2017 and then took postdoctoral positions at Loughborough University (U.K.). She has been with the Department of Engineering at the London South Bank University (U.K.) as Senior Lecturer since Jan. 2022 to Oct. 2022, and Lecturer in 2019-2021. Now she is the Assistant Professor in University of Warwick from Nov. 2022. Her research interests include B5G/6G, heterogeneous networks, Non-terrestrial network, and physical-layer security. She also serves as an Editor for IEEE Wireless Communications Letters and IEEE Transactions on Wireless Communications.

Xinghua Sun (Member, IEEE) received the PhD degree from City University of Hong Kong (CityU) in 2013. In 2010, he was a visiting student with INRIA, France. In 2013, he was a postdoctoral fellow at CityU. From 2015 to 2016, he was a postdoctoral fellow at University of British Columbia, Canada. From July to Aug. 2019, he was a visiting scholar at Singapore University of Technology and Design. From 2014 to 2018, he was an associate professor with Nanjing University of Posts and Telecommunications. Since 2018, he has been an associate professor with Sun Yat-sen University. Dr. Sun served as the Technical Program Committee Member for numerous IEEE conferences. His research interests are in the area of stochastic modeling of wireless networks and machine learning for networking.

Tony Q.S. Quek (Fellow, IEEE) received the B.E. and M.E. degrees in electrical and electronics engineering from the Tokyo Institute of Technology in 1998 and 2000, respectively, and the Ph.D. degree in electrical engineering and computer science from the Massachusetts Institute of Technology in 2008. Currently, he is the Cheng Tsang Man Chair Professor with Singapore University of Technology and Design (SUTD) and ST Engineering Distinguished Professor. He also serves as the Director of the Future Communications R&D Programme, the Head of ISTD Pillar, and the Deputy Director of the SUTD-ZJU IDEA. His current research topics include wireless communications and networking, network intelligence, non-terrestrial networks, open radio access network, and 6G.

Dr. Quek has been actively involved in organizing and chairing sessions, and has served as a member of the Technical Program Committee as well as symposium chairs in a number of international conferences. He is currently serving as an Area Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS.

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 a Fellow of IEEE and a Fellow of the Academy of Engineering Singapore.