Convergence Time Optimization for Federated Learning Over Wireless Networks

Mingzhe Chen, *Member, IEEE*, H. Vincent Poor, *Life Fellow, IEEE*,
Walid Saad, *Fellow, IEEE*, and Shuguang Cui, *Fellow, IEEE*

Abstract—In this paper, the convergence time of federated learning (FL), when deployed over a realistic wireless network, is studied. In particular, a wireless network is considered in which wireless users transmit their local FL models (trained using their locally collected data) to a base station (BS). The BS, acting as a central controller, generates a global FL model using the received local FL models and broadcasts it back to all users. Due to the limited number of resource blocks (RBs) in a wireless network, only a subset of users can be selected to transmit their local FL model parameters to the BS at each learning step. Moreover, since each user has unique training data samples, the BS prefers to include all local user FL models to generate a converged global FL model. Hence, the FL training loss and convergence time will be significantly affected by the user selection scheme. Therefore, it is necessary to design an appropriate user selection scheme that can select the users who can contribute toward improving the FL convergence speed more frequently. This joint learning, wireless resource allocation, and user selection problem is formulated as an optimization problem whose goal is to minimize the FL convergence time and the FL training loss. To solve this problem, a probabilistic user selection scheme is proposed such that the BS is connected to the users whose local FL models have significant effects on the global FL model with high probabilities. Given the user selection policy, the uplink RB allocation can be determined. To further reduce the FL convergence time, artificial neural networks (ANNs) are used to estimate the local FL models of the users that are not allocated any RBs for local FL model transmission at each given learning step, which enables the BS to improve the global model, the FL convergence speed, and the training loss. Simulation results show that the proposed approach can reduce the FL convergence time by up to 56% and improve the accuracy of identifying handwritten digits by up to 3%, compared to a standard FL algorithm.

Index Terms—Federated learning; wireless resource allocation, probabilistic user selection, artificial neural networks.

I. INTRODUCTION

TO TRAIN traditional machine learning algorithms for data analysis and inference, a central controller requires access to all users' training data samples. However, in wireless networks, it is impractical for wireless users to transmit their training data samples to such a central controller due to privacy concerns and the nature of limited wireless resources [2]. In order to train a machine learning model without collection of all of the users' training data samples, *federated learning* (FL) has been proposed in [3]. FL is, in essence, a distributed machine learning algorithm that enables users to collaboratively learn a shared machine learning model while keeping their collected data on their devices. For wireless communications, FL enables many applications such as Internet of Things (IoT) (e.g., ultra reliable low latency services), extended reality (XR), and autonomous vehicles. For example, in autonomous driving, FL enables vehicles to complete the environmental perception and object detection tasks in milliseconds. However, implementing FL and its applications over real-world wireless networks requires edge devices and the BS to repeatedly exchange their FL model parameters. Due to limited wireless resources such as bandwidth, in a wireless network, only a subset of devices can engage in FL. Meanwhile, FL model parameters that are transmitted from the users to a BS are subject to errors and delays caused by the wireless channel which affect the learning performance. For example, when an FL algorithm is implemented over a wireless network, its convergence time not only depends on the number of training steps, but also depends on the ML model parameter transmission time at each training step. Meanwhile, as the number of neurons in the ML model increases, the computation and communication delay of each device will significantly increase thus increasing the FL convergence time. In consequence, energy-constrained devices that are running time-sensitive applications (e.g. virtual reality and ultra-reliable low-latency communication) may not be able or want to wait to complete such a long time training process.

Mingzhe Chen is with the Shenzhen Research Institute of Big Data, Shenzhen 518172, China, and also with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544 USA (e-mail: mingzhec@princeton.edu).

H. Vincent Poor is with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544 USA (e-mail: poor@princeton.edu).

Walid Saad is with the Wireless@VT, Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA 24060 USA (e-mail: walids@vt.edu).

Shuguang Cui is with the Shenzhen Research Institute of Big Data and Future Network of Intelligence Institute (FNii), The Chinese University of Hong Kong, Shenzhen 518172, China (e-mail: shuguangcui@cuhk.edu.cn).

Therefore, it is necessary to consider the optimization of wireless networks to reduce FL training time and training loss.

Recently, the works in [4]–[15] have studied important problems related to the deployment of FL over wireless networks. In [4], the authors provided a new set of design principles for wireless communication and edge learning; however, they did not provide any mathematically rigorous results for minimizing the FL training loss over wireless networks. The authors in [5] proposed a blockchain-based FL scheme that enables edge devices to train FL without sending FL model parameters to a central controller. The works in [6] and [7] optimized the FL performance with communication constraints such as limited spectrum and computational resources. In [8] and [9], the authors investigated the tradeoff between the FL convergence time and users' energy consumption. The authors in [10] developed an echo state network based FL to predict locations and orientations of wireless virtual reality users so as to minimize their breaks in presence. The work in [11] developed a cell-free massive multiple-input multiple-output (MIMO) system for the implementation of FL over wireless networks. In [12], the authors defined a metric to evaluate the FL performance and optimized the batchsize selection and communication resource allocation for the acceleration of the FL training process. The work in [13] derived an analytical model to characterize the training loss of FL in wireless networks and evaluated the effectiveness of different scheduling policies. The authors in [14] studied the use of a relay network to construct a cooperative communication platform for supporting FL model transmission. In [15], the authors developed a federated learning based spiking neural network. However, most of these existing works [4]–[15] whose goal is to minimize the FL convergence time must sacrifice the training loss of the FL algorithm. For example, in [8] and [9], the authors sacrificed the training accuracy of FL to improve the convergence time. Moreover, most of these existing works [4]–[15] used random or fixed user selection methods for FL training, which may significantly increase the FL convergence time and the FL training loss. In addition, none of these existing works [4]–[15] considers the effect of the local FL models of the users that cannot connect to the BS due to limited wireless resources on the FL convergence time and the training loss. In [16], the authors proposed a lazily aggregate gradient (LAG) algorithm for optimizing user selection and local FL model transmission. However, the LAG algorithm in [16] is a deterministic user selection scheme [16] in which some users may not get an opportunity to send their local FL models to the BS thus increasing the training loss.

The main contribution of this work is a novel framework for jointly minimizing the FL convergence time and the FL training loss as captured by the learning loss function. Our key contributions include:

- We develop a realistic implementation of FL over a wireless network in which the users train their local FL models using their own data and transmit the trained local FL models to a base station (BS) over wireless links. The BS aggregates the received local FL models to generate a global FL model and send it back to the users. Since the number of resource blocks (RBs) that are used for FL model transmission is limited, the BS must select an appropriate set of users to perform FL at each learning step. Meanwhile, the BS must allocate its RBs to its users so as to improve the convergence speed. Hence, we formulate this joint user selection and RB allocation problem as an optimization problem whose goal is to minimize the number of iterations needed for FL convergence as well as the time duration of each learning step while optimizing the FL training loss, in terms of the accuracy.

- To solve this problem, we first propose a probabilistic user selection scheme in which users whose local FL models have large effects on the global FL model, will have high probabilities of connecting to the BS. The reason that optimizing user selection can improve FL convergence speed is that the global model contains less information from the local datasets of some users. In this case, increasing the connectivity probability of these users enables the global model to acquire more information from these users thus improving the FL convergence speed. Moreover, the proposed user selection scheme guarantees that every user has a non-zero chance of connecting to the BS and, hence, it enables the BS to learn from all of the users' training data samples so as to guarantee that the FL algorithm achieves the optimal training loss. Given the user selection scheme, the optimal RB allocation scheme can be determined.[1]

- To further reduce the FL convergence time, we propose the use of artificial neural networks (ANNs) to find a relationship among the users' local FL model parameters. Building this relationship enables the BS to estimate the local FL model parameters of the users that cannot transmit their local FL models to the BS due to the limited number of RBs at each given learning step. Hence, using ANNs, the BS can integrate more users' local FL model parameters to generate the global FL model and, hence, decrease the FL training loss and improve the convergence speed.

- We perform fundamental analysis on the expression of expected convergence rate of the proposed FL algorithm and we show that the FL training method (with full gradient descent or stochastic gradient descent [17]), RB allocation, user selection scheme, and the accuracy of predicting local FL models of users will significantly affect the convergence speed and training loss of FL.

Simulation results assess the various performance metrics of the proposed approach and show that the proposed FL approach can reduce the convergence time by up to 56% and improve the FL identification accuracy by up to 3%, compared to a standard FL algorithm. To our best knowledge, *this is the first work that studies the use of ANNs for the prediction of FL model parameters that are used in the training process so as to improve the FL convergence speed and training loss.*

---

[1]Optimizing RB allocation can improve the users' data rates thus decreasing the local FL model transmission delay. In consequence, we can optimize RB allocation to decrease the FL convergence time.
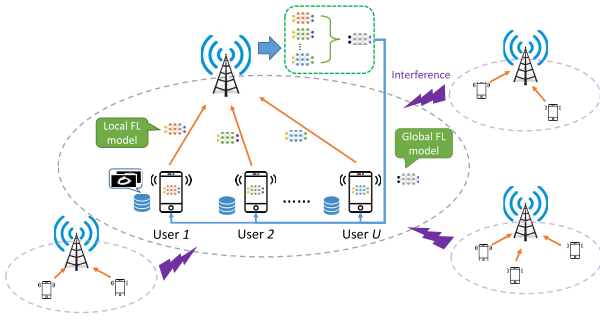
Fig. 1. The architecture of an FL algorithm that is being executed over a wireless network with multiple devices and a single base station.

The rest of this paper is organized as follows. The system model and problem formulation are described in Section II. The probabilistic user selection scheme, resource allocation scheme, and the use of ANNs for the prediction of users' local FL models are introduced in Section III. Simulation results are analyzed in Section IV. Conclusions are drawn in Section V.

II. SYSTEM MODEL AND PROBLEM FORMULATION

Consider a wireless network in which a set $\mathcal{U}$ of U users and one BS jointly execute an FL algorithm for data analysis and inference, as shown in Fig. 1. We assume that each user i collects K_i training data samples and each training data sample k consists of an input $\boldsymbol{x}_{ik} \in \mathbb{R}^{N_{\text{in}} \times 1}$ and its corresponding output $\boldsymbol{y}_{ik} \in \mathbb{R}^{N_{\text{out}} \times 1}$. We also assume that the data collected by the users follows the same distribution [18] or statistical heterogeneity [19]. The FL training process is done in a way to solve

$$\min_{\boldsymbol{g}} \frac{1}{K} \sum_{i=1}^{U} \sum_{k=1}^{K_i} f(\boldsymbol{g}, \boldsymbol{x}_{ik}, \boldsymbol{y}_{ik}), \qquad (1)$$

where $K = \sum_{i=1}^{U} K_i$ is total number of training data samples of all users, $\boldsymbol{g}$ is a vector that captures the FL model trained by K training data samples, and $f(\boldsymbol{g}, \boldsymbol{x}_{ik}, \boldsymbol{y}_{ik})$ is a loss function that captures the accuracy of the considered FL algorithm by building a relationship between an input vector $\boldsymbol{x}_{ik}$ and an output vector $\boldsymbol{y}_{ik}$. For different learning tasks, the definition of the loss functions can be different [20]. In particular, for our considered handwritten digit classification task, the cross entropy is $f(\boldsymbol{g}, \boldsymbol{x}_{ik}, \boldsymbol{y}_{ik}) = -\boldsymbol{y}_{ik} \log(\boldsymbol{x}_{ik}^{\mathrm{T}} \boldsymbol{g}) + (1 - \boldsymbol{y}_{ik}) \log(1 - \boldsymbol{x}_{ik}^{\mathrm{T}} \boldsymbol{g})$. To find the optimal vector $\boldsymbol{g}$, conventional centralized learning algorithms require the users to share their collected data with other users, which is impractical due to communication overhead and data privacy issues. To address these issues, FL can be used. The training procedure of FL proceeds as follows [3]:

a. The BS broadcasts the information that is used to initialize the learning model to each user. Then, each user initializes its learning model.

b. Each user trains its generated machine learning model using its collected data and sends the trained learning model parameters to the BS.

c. The BS integrates the received learning model parameters and broadcasts them back to all users.

d. Steps b. and c. are repeated until the optimal vector $\boldsymbol{g}$ is found.

From this FL training procedure, we observe that the users do not have to share their collected data with other users and the BS. In contrast, they only need to share their trained learning model parameters with the BS. Note that the implementation of steps b. and c. constitutes one iteration of training FL. Hereinafter, the FL model that is trained by each user using its collected data is called *local FL model* while the FL model that is generated by the BS is called *global FL model*. Since all of the FL model parameters are transmitted over wireless networks, we must consider the effect of wireless factors such as resource blocks, user selection, dynamic wireless channel, and transmission delay, on the FL training loss and convergence time. Next, we first mathematically formulate the training procedure of FL. Then, the transmission of FL model parameters over wireless links is introduced. Finally, the problem of minimzing the FL convergence time and the training loss is formulated. Table I provides a summary of the notations used hereinafter.

A. Training Procedure of Federated Learning

The local FL model of each user i at each iteration μ is defined as a vector $\boldsymbol{w}_{i,\mu} \in \mathbb{R}^{W \times 1}$, where W is the number of elements in $\boldsymbol{w}_{i,\mu}$. The update of the global FL model at iteration μ is given by [21]

$$\boldsymbol{g}(\boldsymbol{a}_\mu) = \sum_{i=1}^{U} \boldsymbol{w}_{i,\mu} \left(\frac{a_{i,\mu} K_i}{\sum_{i=1}^{U} a_{i,\mu} K_i} \right), \qquad (2)$$

where $\boldsymbol{g}(\boldsymbol{a}_\mu)$ is the global FL model at iteration μ, $\frac{a_{i,\mu} K_i}{\sum_{i=1}^{U} a_{i,\mu} K_i}$ is a scaling update weight of $\boldsymbol{w}_{i,\mu}$ and $\boldsymbol{a}_\mu = [a_{1,\mu}, \ldots, a_{U,\mu}]$ is a user association vector with $a_{i,\mu} \in \{0, 1\}$, $a_{i,\mu} = 1$ indicates that user i connects to the BS and user i must send its local FL model $\boldsymbol{w}_{i,\mu}$ to the BS at iteration μ, otherwise, we have $a_{i,\mu} = 0$. From (2), we see that the user association vector $\boldsymbol{a}_\mu$ will change at each iteration. This is due to the fact that, in wireless networks, the number of RBs is limited and the BS must select a suitable set of users to transmit their local FL models in each iteration.

The update of the local FL model $\boldsymbol{w}_{i,\mu}$ depends on the used training method. In our model, the full gradient descent method [18] is used to update the local FL model. In Section III-D, we revisit our analysis when FL is executed with a stochastic gradient descent method to update the local FL model. Using the full gradient descent method, the update of local FL model can be given by

$$\boldsymbol{w}_{i,\mu+1} = \boldsymbol{g}(\boldsymbol{a}_\mu) - \frac{\lambda}{K_i} \sum_{k=1}^{K_i} \nabla f(\boldsymbol{g}(\boldsymbol{a}_\mu), \boldsymbol{x}_{ik}, \boldsymbol{y}_{ik}), \qquad (3)$$

where λ is the learning rate and $\nabla f(\boldsymbol{g}(\boldsymbol{a}_\mu), \boldsymbol{x}_{ik}, \boldsymbol{y}_{ik})$ is the gradient of $f(\boldsymbol{g}(\boldsymbol{a}_\mu), \boldsymbol{x}_{ik}, \boldsymbol{y}_{ik})$ with respect to $\boldsymbol{g}(\boldsymbol{a}_\mu)$. Based

TABLE I

LIST OF NOTATIONS

Notation	Description	Notation	Description
$\boldsymbol{g}$	Global FL model	K	Total number of training data samples of all users
$\boldsymbol{a}_\mu$	User selection vector at iteration μ	K_i	Number of training data samples of each user i
λ	Learning rate	$p_{i,\mu}$	Probability of user i connecting to the BS at iteration μ
R	Number of RBs	$\boldsymbol{w}_{i,\mu}$	Local FL model of user i at iteration μ
γ	Prediction error requirement	$\boldsymbol{r}_{i,\mu}$	RB allocation vector of user i at iteration μ
B	Bandwidth of each RB	$E_{i,\mu}$	Prediction error of user i's local FL model at iteration μ
γ_R	Number of users considered in (10)	$\boldsymbol{R}_\mu$	RB allocation matrix at iteration μ
c_i^{D}	Downlink data rate of each user i	$\boldsymbol{R}$	RB allocation matrix from iteration 1 to T
I_n	Interference over RB n	$\boldsymbol{A}$	User selection matrix from iteration 1 to T
B^{D}	Downlink bandwidth	$\hat{\boldsymbol{w}}_{i,\mu}$	Predicted local FL model of user i at iteration μ
Z	Data size of local or global FL model	i^*	The user always connects to the BS
Ω_μ	Convergence indicator at iteration μ	T	Maximum number of iterations

on (2) and (3), the BS and the users can update their FL models so as to find the optimal global FL model that solves problem (1).

B. Transmission Model

We assume that an orthogonal frequency-division multiple access (OFDMA) technique is adopted for local FL model transmission from the users to the BS. In our model, we assume that the total number of RBs that the BS can allocate to its users is R, and each user can occupy only one RB. The uplink rate of user i that is transmitting its local FL model parameters to the BS at iteration μ is given by [22]

$$c_i^{\mathrm{U}}(\boldsymbol{r}_{i,\mu}) = \sum_{n=1}^{R} r_{in,\mu} B \log_2 \left(1 + \frac{P h_i}{I_n + B N_0} \right), \qquad (4)$$

where $\boldsymbol{r}_{i,\mu} = [r_{i1,\mu}, \ldots, r_{iR,\mu}]$ is an RB allocation vector with $r_{in,\mu} \in \{0,1\}$; $r_{in,\mu} = 1$ indicates that RB n is allocated to user i at iteration μ, otherwise, we have $r_{in,\mu} = 0$, and P is the transmit power of user i, which is assumed to be equal for all users. h_i is the channel gain between user i and the BS, N_0 is the noise power spectral density, and I_n is the interference caused by the users that connect to other BSs using the same RB. Since the interference caused by the users using the same RBs may significantly affect the transmission delay and the time needed for FL to converge, we must consider this interference in (4).

Similarly, the downlink data rate of the BS when transmitting the global FL model parameters to each user i is given by [22]

$$c_i^{\mathrm{D}} = B^{\mathrm{D}} \log_2 \left(1 + \frac{P_B h_i}{B^{\mathrm{D}} N_0} \right), \qquad (5)$$

where B^{D} is the bandwidth that the BS used to broadcast the global FL model to each user i and P_B is the transmit power of the BS.

Since the number of elements in the local FL model $\boldsymbol{w}_{i,\mu}$ is similar to that of the global FL model $\boldsymbol{g}(\boldsymbol{a}_\mu)$, the data size of the local FL model $\boldsymbol{w}_{i,\mu}$ is equal to the data size of the global FL model $\boldsymbol{g}(\boldsymbol{a}_\mu)$. Here, the data size of the local FL model is defined as the number of bits that the user requires to transmit the local FL model vector $\boldsymbol{w}_{i,\mu}$ to the BS. Let Z be the data size of a global FL model or local FL model. The

transmission delay between user i and the BS over both uplink and downlink at iteration μ will be

$$l_i^{\mathrm{U}}(\boldsymbol{r}_{i,\mu}) = \frac{Z}{c_i^{\mathrm{U}}(\boldsymbol{r}_{i,\mu})}, \qquad (6)$$

$$l_i^{\mathrm{D}} = \frac{Z}{c_i^{\mathrm{D}}}. \qquad (7)$$

The time that the users and the BS require to jointly complete an update of their respective local and global FL models at iteration μ is given by

$$t_\mu(\boldsymbol{a}_\mu, \boldsymbol{R}_\mu) = \max_{i \in \mathcal{U}} a_{i,\mu} \left(l_i^{\mathrm{U}}(\boldsymbol{r}_{i,\mu}) + l_i^{\mathrm{D}} \right), \qquad (8)$$

where $\boldsymbol{R}_\mu = [\boldsymbol{r}_{1,\mu}, \ldots, \boldsymbol{r}_{U,\mu}]$. When $a_{i,\mu} = 0$, $a_{i,\mu}(l_i^{\mathrm{U}}(\boldsymbol{r}_{i,\mu}) + l_i^{\mathrm{D}}) = 0$. Here, $a_{i,\mu} = 0$ implies that user i will not send its local FL model to the BS at iteration μ, and hence, user i will not cause any delay at iteration μ, $(a_{i,\mu}(l_i^{\mathrm{U}}(\boldsymbol{r}_{i,\mu}) + l_i^{\mathrm{D}}) = 0)$. When $a_{i,\mu} = 1$, then user i will transmit its local FL model to the BS and the transmission delay will be $l_i^{\mathrm{U}}(\boldsymbol{r}_{i,\mu}) + l_i^{\mathrm{D}}$. Hence, (8) is essentially the worst-case transmission delay among all selected users.

C. Problem Formulation

Having defined the system model, the next step is to introduce a joint RB allocation and user selection scheme to minimize the time that the users and the BS need in order to complete the FL training process. This optimization problem is formulated as follows:

$$\min_{\boldsymbol{A}, \boldsymbol{R}} \sum_{\mu=1}^{T} t_\mu(\boldsymbol{a}_\mu, \boldsymbol{R}_\mu) \Omega_\mu \qquad (9)$$

$$\text{s. t. } a_{i,\mu}, r_{in,\mu}, \Omega_\mu \in \{0,1\}, \quad \forall i \in \mathcal{U}, n = 1, \ldots, R, \tag{9a}$$

$$\sum_{i \in \mathcal{U}} r_{i,n} \leq 1, \ \forall n = 1, \ldots, R, \tag{9b}$$

$$\sum_{n=1}^{R} r_{in,\mu} = a_{i,\mu}, \ \forall i \in \mathcal{U}, \tag{9c}$$

where $\boldsymbol{A} = [\boldsymbol{a}_1, \ldots, \boldsymbol{a}_\mu, \ldots, \boldsymbol{a}_T]$ is a user selection matrix for all iterations, $\boldsymbol{R} = [\boldsymbol{R}_1, \ldots, \boldsymbol{R}_\mu, \ldots, \boldsymbol{R}_T]$ is an RB allocation matrix for all users at all iterations, and T is a constant, which is large enough to guarantee the convergence of FL. In other words, the number of iterations that the FL algorithm
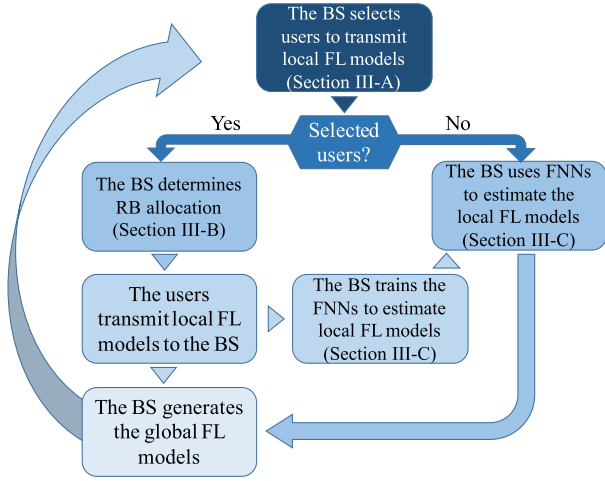
Fig. 2.   The training procedure of the proposed FL.

requires to converge will not be larger than T. In (9), $\Omega_\mu = 1$ implies that the FL algorithm does not converge, otherwise, we have $\Omega_\mu = 0$, (9a) and (9b) imply that each user can only occupy one RB, and (9c) implies that all RBs must be allocated to the users that are associated with the BS. From (9), we see that the time used for the update of the local and global FL models, t_μ, depends on the user selection vector $\boldsymbol{a}_\mu$ and RB allocation matrix $\boldsymbol{R}_\mu$. Meanwhile, as shown in (3), the total number of iterations that the FL algorithm needs in order to converge depends on the user selection vector $\boldsymbol{a}_\mu$. In consequence, the time duration of each FL training iteration and the number of iterations needed for the FL algorithm to converge are dependent. Moreover, given the global model $\boldsymbol{g}(\boldsymbol{a}_\mu)$ at iteration μ, the BS cannot calculate the number of iterations that the FL algorithm needs to converge since all of the training data samples are located at the users' devices. Hence, problem (9) is challenging to solve.

III. MINIMIZATION OF CONVERGENCE TIME OF FEDERATED LEARNING

To solve problem (9), we first need to determine the user association at each iteration. Given the user selection vector, the optimal RB allocation scheme can be derived. To further improve the convergence speed of FL, ANNs are introduced to estimate the local FL models of the users that are not allocated any RBs for transmitting their local model parameters at each given learning step. Fig. 2 summarizes the training procedure of the proposed FL.

A. Gradient Based User Association Scheme

To predict the local FL model of each user, the BS needs to use the local FL model of a given user as an ANN input, as will be explained in Subsection III-C. Hence, in the proposed user association scheme, one user must be selected to connect with the BS during all training iterations. To determine the user that connects to the BS during the entire training process, we first assume that the distance d_i between user i and the BS satisfies

$d_1 \leq d_2 \leq \ldots \leq d_U$. Hence, user i^* that always connects to the BS can be found from

$$i^* = \arg\max_{i \in \mathcal{U}} \sum_{k=1}^{K_i} \nabla f\left(\boldsymbol{g}\left(\boldsymbol{a}_{\mu-1}\right), \boldsymbol{x}_{ik}, \boldsymbol{y}_{ik}\right), \tag{10}$$

$$\text{s. t. } d_1 \leq d_i \leq d_{\gamma_R}, \quad \forall i \in \mathcal{U}, \tag{10a}$$

where γ_R is the number of users considered in (10) with $1 \leq \gamma_R \leq U$. As γ_R increases, the number of users considered in (10) increases. Hence, the transmission delay of user i^* may increase thus increasing the time used to complete one FL iteration. However, as γ_R increases, the value of $\max_{i \in \mathcal{U}} \sum_{k=1}^{K_i} \nabla f\left(\boldsymbol{g}\left(\boldsymbol{a}_{\mu-1}\right), \boldsymbol{x}_{ik}, \boldsymbol{y}_{ik}\right)$ may increase, and, thus, the number of iterations required for FL to converge decreases. Here, user i^* is determined at the first iteration.

From (3), we can observe that, at each iteration μ, the global FL model $\boldsymbol{g}(\boldsymbol{a}_{\mu-1})$ will change $\lambda \sum_{k=1}^{K_i} \nabla f\left(\boldsymbol{g}\left(\boldsymbol{a}_{\mu-1}\right), \boldsymbol{x}_{ik}, \boldsymbol{y}_{ik}\right)$ due to the local FL model of a given user i. We define a vector $\boldsymbol{e}_{i,\mu} = \lambda \sum_{k=1}^{K_i} \nabla f\left(\boldsymbol{g}\left(\boldsymbol{a}_{\mu-1}\right), \boldsymbol{x}_{ik}, \boldsymbol{y}_{ik}\right)$ as the change in the global FL model due to user i's local FL model. To enable the BS to predict each user's local FL model at each learning step, each user must have a chance of connecting to the BS so as to provide the training data samples (local FL model parameters) to the BS for training ANNs. Therefore, a probabilistic user association scheme is developed, which is given by

$$p_{i,\mu} = \begin{cases} \dfrac{\|\boldsymbol{e}_{i,\mu}\|}{\sum\limits_{i=1, i \neq i^*}^{U} \|\boldsymbol{e}_{i,\mu}\|}, & \text{if } i \neq i^*, \\ 1, & \text{if } i = i^*, \end{cases} \tag{11}$$

where $p_{i,\mu}$ represents the probability of user i connecting to the BS at iteration μ, and $\|\boldsymbol{e}_{i,\mu}\|$ is the norm of vector $\boldsymbol{e}_{i,\mu}$. From (11), we can see that, as $\|\boldsymbol{e}_{i,\mu}\|$ increases, the probability of associating user i with the BS increases. In consequence, the probability that the BS uses user i's local FL model to generate the global FL model increases. Hence, using the proposed user association scheme in (11), the BS has a high probability of connecting to users whose local FL models significantly affect the global FL model, thus improving the FL convergence speed. From (11), we also see that user i^* will always connect to the BS so as to provide information for the prediction of other users' local FL models. Based on (11), the user association scheme at each iteration can be determined. To calculate $p_{i,\mu}$ in (11), the BS only needs to know $\|\boldsymbol{e}_{i,\mu}\|$ of each user i without requiring the exact training data information. In fact, $\|\boldsymbol{e}_{i,\mu}\|$ can be directly calculated by user i and each user i needs to transmit only a scalar $\|\boldsymbol{e}_{i,\mu}\|$ to the BS.

B. Optimal RB Allocation Scheme

Given the user association scheme at each iteration μ, problem (9) at iteration μ can be simplified as follows:

$$\min_{\boldsymbol{R}_\mu} t_\mu\left(\boldsymbol{R}_\mu\right) = \min_{\boldsymbol{R}_\mu} \max_{i \in \mathcal{U}} a_{i,\mu} \left(l_i^{\mathrm{U}}\left(\boldsymbol{r}_{i,\mu}\right) + l_i^{\mathrm{D}}\right) \tag{12}$$

$$\text{s. t. } r_{in,\mu} \in \{0,1\}, \quad \forall i \in \mathcal{U}, n = 1, \dots, R, \tag{12a}$$

$$\sum_{i \in \mathcal{U}} r_{in,\mu} \le 1, \ \forall n = 1, \dots, R, \tag{12b}$$

$$\sum_{n=1}^{R} r_{in,\mu} = a_{i,\mu}, \ \forall i \in \mathcal{U}. \tag{12c}$$

We assume that there exists a variable m that satisfies $a_{i,\mu}\left(l_i^{\mathrm{U}}\left(\boldsymbol{r}_{i,\mu}\right) + l_i^{\mathrm{D}}\right) \le m$. Problem (12) can be simplified as follows:

$$\min_{\boldsymbol{R}_\mu} m \tag{13}$$

$$\text{s. t. (12a), (12b), and (12c),} \tag{13a}$$

$$m \ge a_{i,\mu}\left(l_i^{\mathrm{U}}\left(\boldsymbol{r}_{i,\mu}\right) + l_i^{\mathrm{D}}\right), \forall i \in \mathcal{U}. \tag{13b}$$

Since (13b) is nonlinear, we must transform it into a linear constraint. We first assume that $l_{in,\mu}^{\mathrm{U}} = \frac{Z}{B\log_2\left(1 + \frac{Ph_i}{I_n + BN_0}\right)}$, which represents the delay of user i transmitting the local FL model over RB n at iteration μ. Then, we have $l_i^{\mathrm{U}}\left(\boldsymbol{r}_{i,\mu}\right) = \sum_{n=1}^{R} r_{in,\mu} l_{in,\mu}^{\mathrm{U}}$. Hence, problem (13) can be rewritten as follows:

$$\min_{\boldsymbol{R}_\mu} m \tag{14}$$

$$\text{s. t. (12a), (12b), and (12c),} \tag{14a}$$

$$m \ge a_{i,\mu}\left(\sum_{n=1}^{R} r_{in,\mu} l_{in,\mu}^{\mathrm{U}} + l_i^{\mathrm{D}}\right), \quad \forall i \in \mathcal{U}. \tag{14b}$$

Problem (14) is equivalent to (13) and is an integer linear programming problem, which can be solved by known optimization algorithms such as interior-point methods [23].

C. Prediction of the Local FL Models

The previous subsections determine the users that are associated with the BS at each iteration and minimize their transmission delay by optimizing the RB allocation. Next, we introduce an ANN-based algorithm to predict the local FL model parameters of the users that are not allocated any RBs for local FL model transmission at each given learning step. In particular, ANNs are used to build a relationship between the local FL models of different users. Since the fully-connected multilayer perceptrons (MLPs) in ANNs are good at function fitting tasks and finding a relationship among different users' local FL models is a regression task, we prefer to use MLPs instead of other neural networks such as recurrent neural networks. Next, we first introduce the architecture of our FNN-based algorithm. Then, we explain how to implement this algorithm to predict the local FL model parameters of the users at each given learning step.

Our FNN-based prediction algorithm consists of three components: a) input, b) a single hidden layer, and c) output, which is defined as follows:

- *Input*: The input of the FNN that is used for the prediction of user j's local FL model is a vector $\boldsymbol{w}_{i^*,\mu}$, which represents the local FL model of user i^*. As we mentioned in Subsection III-A, user i^* will always connect with the BS so as to provide the input information for the MLP to predict the local FL models of other users.
- *Output*: The output of the FNN for the prediction of user j's local FL model is a vector $\boldsymbol{o} = \boldsymbol{w}_{i^*,\mu} - \boldsymbol{w}_{j,\mu}$, which represents the difference between user i^*'s local FL model and user j's local FL model. Based on the prediction output $\boldsymbol{o}$ and user i^*'s local FL model, we can obtain the local FL model of user j, i.e., $\hat{\boldsymbol{w}}_{j,\mu} = \boldsymbol{w}_{i^*,\mu} - \boldsymbol{o}$ with $\hat{\boldsymbol{w}}_{j,\mu}$ being the predicted user j's local FL model.
- *A Single Hidden Layer*: The hidden layer of an FNN allows it to learn nonlinear relationships between input vector $\boldsymbol{w}_{i^*,\mu}$ and output vector $\boldsymbol{o}$. Mathematically, a single hidden layer consists of N neurons. The weight matrix that represents the connection strength between the input vector and the neurons in the hidden layer is $\boldsymbol{v}^{\mathrm{in}} \in \mathbb{R}^{N \times W}$. Meanwhile, the weight matrix that captures the strengths of the connections between the neurons in the hidden layer and the output vector is $\boldsymbol{v}^{\mathrm{out}} \in \mathbb{N}^{W \times N}$.

Given the components of the MLP, next, we introduce the use of the MLP to predict each user's local FL model. The states of the neurons in the hidden layer are given by

$$\boldsymbol{\vartheta} = \sigma\left(\boldsymbol{v}^{\mathrm{in}} \boldsymbol{w}_{i^*,\mu} + \boldsymbol{b}_\vartheta\right), \tag{15}$$

where $\sigma\left(\boldsymbol{x}\right) = \frac{2}{1 + \exp(-2\boldsymbol{x})} - 1$ and $\boldsymbol{b}_\vartheta \in \mathbb{R}^{N \times 1}$ is the bias. Given the neuron states, we can obtain the output of the FNN, as follows:

$$\boldsymbol{o} = \boldsymbol{v}^{\mathrm{out}} \boldsymbol{\vartheta} + \boldsymbol{b}_o, \tag{16}$$

where $\boldsymbol{b}_o \in \mathbb{R}^{W \times 1}$ is a vector of bias. Based on (16), we can calculate the predicted local FL model of each user j at each iteration μ, i.e., $\hat{\boldsymbol{w}}_{j,\mu} = \boldsymbol{w}_{i^*,\mu} - \boldsymbol{o}$. To enable the FNN to predict each user's local FL model, the FNN must be trained by the online gradient descent method [24].

Given the prediction of the users' FL models, the update of the global FL model can be rewritten by

$$\boldsymbol{g}\left(\boldsymbol{a}_\mu\right) = \frac{\sum\limits_{i=1}^{U} K_i a_{i,\mu} \boldsymbol{w}_{i,\mu} + \sum\limits_{i=1}^{U} K_i \left(1 - a_{i,\mu}\right) \hat{\boldsymbol{w}}_{i,\mu} \mathbb{1}_{\{E_{i,\mu} \le \gamma\}}}{\sum\limits_{i=1}^{U} K_i a_{i,\mu} + \sum\limits_{i=1}^{U} K_i \left(1 - a_{i,\mu}\right) \mathbb{1}_{\{E_{i,\mu} \le \gamma\}}}, \tag{17}$$

where $\sum\limits_{i=1}^{U} K_i a_{i,\mu} \boldsymbol{w}_{i,\mu}$ is the sum of the local FL models of the users that connect to the BS at iteration μ and $\sum\limits_{i=1}^{U} K_i \left(1 - a_{i,\mu}\right) \hat{\boldsymbol{w}}_{i,\mu} \mathbb{1}_{\{E_{i,\mu} \le \gamma\}}$ is the sum of the predicted local FL models of the users that are not associated with the BS at iteration μ, $E_{i,\mu} = \frac{1}{2W} \|\hat{\boldsymbol{w}}_{i,\mu} - \boldsymbol{w}_{i,\mu}\|^2$ is the prediction error at iteration μ, and γ is the prediction requirement. In (17), when the prediction accuracy of the FNN cannot meet the prediction requirement (i.e., $E_{i,\mu} > \gamma$), the BS will not use the prediction result for updating its global FL model. From (17), we can also observe that, using MLPs, the BS can include

Algorithm 1 Proposed FL Over Wireless Network

Init: Local FL model of each user i, $\boldsymbol{w}_i$, FNN model for the prediction of local FL model, $\boldsymbol{v}^{\text{in}}, \boldsymbol{v}^{\text{out}}, \boldsymbol{b}_\vartheta, \boldsymbol{b}_o$, user i^* that always connects to the BS.

1: **for** iteration μ **do**
2:     Each user i trains its local FL model to obtain $\boldsymbol{w}_{i,\mu}$.
3:     Each user i calculates the change of gradient of the local FL model $\boldsymbol{e}_{i,\mu}$ and sends $|\boldsymbol{e}_{i,\mu}|$ to the BS.
4:     The BS calculates $p_{i,\mu}$ using (11) and determines $\boldsymbol{a}_\mu$.
5:     The BS determines $\boldsymbol{R}_\mu$ in (12).
6:     The selected users ($a_{i,\mu} = 1$) transmit their local FL models to the BS based on $\boldsymbol{R}_\mu$ and $\boldsymbol{a}_\mu$.
7:     The BS uses MLPs to estimate the local FL models of the users who are not associated with the BS ($a_{i,\mu} = 0$).
8:     The BS calculates the global FL model $\boldsymbol{g}$ using (17).
9:     The BS uses collected local FL models to train the MLPs.
10: **end for**

---

additional local FL models to generate the global FL model so as to decrease the FL training loss and improve the FL convergence speed. (17) is used to generate the global FL model in Step c. of the FL training procedure specified in Section II.

The proposed FL algorithm that jointly minimizes the FL convergence time FL training loss is shown in Algorithm 1. From Algorithm 1, we can see that the user selection and RB allocation are optimized at each FL iteration and, hence problem (9) is solved at each FL iteration.

D. Convergence, Implementation, and Complexity Analysis

1) Convergence Analysis: Next, we analyze the convergence of the proposed FL algorithm. We first assume that $F(\boldsymbol{g}_\mu) = \frac{1}{K} \sum_{i=1}^{U} \sum_{k=1}^{K_i} f(\boldsymbol{g}_\mu, \boldsymbol{x}_{ik}, y_{ik})$ and $F_i(\boldsymbol{g}_\mu) = \frac{1}{K_i} \sum_{k=1}^{K_i} f(\boldsymbol{g}_\mu, \boldsymbol{x}_{ik}, y_{ik})$ where $\boldsymbol{g}_\mu$ is short for $\boldsymbol{g}(\boldsymbol{a}_\mu)$. We also assume that $\|\nabla F_i(\boldsymbol{g}_\mu)\|^2 \leq \zeta_\mu^1 + \zeta_\mu^2 \|\nabla F(\boldsymbol{g}_\mu)\|^2$ and $\|\nabla \hat{F}_i(\boldsymbol{g}_\mu)\| = \varsigma_\mu^1 + \varsigma_\mu^2 \|\nabla F(\boldsymbol{g}_\mu)\|$ where $\|\nabla \hat{F}_i(\boldsymbol{g}_\mu)\|$ is the gradient deviation that is caused by the prediction inaccuracy of user i's local FL model. We also assume that $\boldsymbol{g}^*$ is the optimal global FL model that is achieved by the FL algorithm that can collect all users' local FL models at each iteration. The convergence of the proposed FL algorithm at each iteration μ is given by the following theorem.

Theorem 1: Given the optimal global FL model $\boldsymbol{g}^*$, the gradient deviation caused by the prediction inaccuracy of users' local FL models, $\|\nabla \hat{F}_i(\boldsymbol{g}_\mu)\|^2$, and the learning rate $\lambda = \frac{1}{L}$, the upper bound of $\mathbb{E}\left[F(\boldsymbol{g}_{\mu+1}) - F(\boldsymbol{g}^*)\right]$ can be given by

$$\mathbb{E}\left[F(\boldsymbol{g}_{\mu+1}) - F(\boldsymbol{g}^*)\right] \leq \varpi_\mu^1 + \varpi_\mu^2 \mathbb{E}\left(F(\boldsymbol{g}_\mu) - F(\boldsymbol{g}^*)\right). \tag{18}$$

where $\varpi_\mu^1 = \frac{\zeta_\mu^1 (9K - 8\mathbb{E}(A))}{2LK}$ determines the global FL model at convergence and the convergence speed with $\mathbb{E}(A) = \sum_{i=1}^{U} K_i (1 - p_{i,\mu}) \mathbb{1}_{\{E_{i,\mu} \leq \gamma\}} + \sum_{i=1}^{U} K_i p_{i,\mu}$ and $\varpi_\mu^2 =$

$\left(1 - \frac{\vartheta}{L} + \frac{\vartheta \zeta_\mu^2 (9K - 8\mathbb{E}(A))}{LK}\right)$ also affects the convergence speed.

Proof: See Appendix A. $\qquad\square$

From Theorem 1, we can see that as ϖ_μ^1 decreases, the gap between $\boldsymbol{g}_\mu$ and $\boldsymbol{g}^*$ decreases. In particular, as $\varpi_\mu^1 = 0$, the proposed FL algorithm will converge to the optimal FL model $\boldsymbol{g}^*$ and achieve the optimal training loss. Hereinafter, the gap between the global FL model that the proposed FL algorithm converges to, $\boldsymbol{g}_\mu$, and the optimal global FL model $\boldsymbol{g}^*$ is referred as the convergence accuracy. ϖ_μ^1 is used to capture the convergence accuracy. From Theorem 1, we can also see that ϖ_μ^1 and ϖ_μ^2 jointly determine the convergence speed. This is because as ϖ_μ^1 and ϖ_μ^2 decrease, the value of $\mathbb{E}\left[F(\boldsymbol{g}_{\mu+1}) - F(\boldsymbol{g}^*)\right]$ will be smaller than the value of $\mathbb{E}\left[F(\boldsymbol{g}_\mu) - F(\boldsymbol{g}^*)\right]$. Hence, as ϖ_μ^1 and ϖ_μ^2 decrease, the speed of $\boldsymbol{g}_{\mu+1}$ converging to $\boldsymbol{g}^*$ increases. Theorem 1 shows that the prediction of local FL models affects both convergence speed and convergence accuracy. In particular, as the prediction errors of local FL models $E_{i,\mu}$ is smaller than γ, then the predicted local FL models can be used to update the global FL model hence improving the convergence speed and accuracy of the proposed FL algorithm. From Theorem 1, we can also see that the number of training data samples K_i and the connection probability $p_{i,\mu}$ of each user i, also affect the convergence speed and accuracy.

Based on Theorem 1, we have the following observations.

Corollary 1: If the BS cannot predict the local FL models of the users that are not allocated any RBs, the upper bound of $\mathbb{E}\left[F(\boldsymbol{g}_{\mu+1}) - F(\boldsymbol{g}^*)\right]$ can be given by

$$\begin{aligned} &\mathbb{E}\left[F(\boldsymbol{g}_{\mu+1}) - F(\boldsymbol{g}^*)\right] \\ &\leq \frac{\zeta_\mu^1 \left(9K - 8\sum_{i=1}^{U} K_i p_{i,\mu}\right)}{2LK} \\ &+ \left(1 - \frac{\vartheta}{L} + \frac{\vartheta \zeta_\mu^2 \left(9K - 8\sum_{i=1}^{U} K_i p_{i,\mu}\right)}{LK}\right) \mathbb{E}\left(F(\boldsymbol{g}_\mu)\right) \\ &- F(\boldsymbol{g}^*)). \end{aligned} \tag{19}$$

Proof: If the BS cannot predict the users' local FL models, we have $\mathbb{1}_{\{E_{i,\mu} \leq \gamma\}} = 0$ and hence $\mathbb{E}(A) = \sum_{i=1}^{U} K_i p_{i,\mu}$. Substituting $\mathbb{E}(A) = \sum_{i=1}^{U} K_i p_{i,\mu}$ into (18), we can obtain (19). This completes the proof. $\qquad\square$

From Corollary 1 and Theorem 1, we can see that $\varpi_\mu^1 \leq \frac{\zeta_\mu^1 \left(9K - 8\sum_{i=1}^{U} K_i p_{i,\mu}\right)}{2LK}$ and $\varpi_\mu^2 \leq 1 - \frac{\vartheta}{L} + \frac{\vartheta \zeta_\mu^2 \left(9K - 8\sum_{i=1}^{U} K_i p_{i,\mu}\right)}{LK}$, which implies that the prediction of local FL models can improve the convergence speed and convergence accuracy. Hence, this is the lower bound of the expected convergence of the proposed FL algorithm. Corollary 1 also shows that there exists a gap between $\boldsymbol{g}_\mu$ and $\boldsymbol{g}^*$ at convergence. This gap is caused by the probabilistic user association.

Corollary 2: If the prediction accuracy of each user's local FL model satisfies $E_{i,\mu} \leq \gamma$, the upper bound of

$\mathbb{E}\left[F\left(\boldsymbol{g}_{\mu+1}\right)-F\left(\boldsymbol{g}^*\right)\right]$ can be given by

$$\mathbb{E}\left[F\left(\boldsymbol{g}_{\mu+1}\right)-F\left(\boldsymbol{g}^*\right)\right]$$
$$\leq \frac{\zeta_\mu^1}{2L} + \left(1 - \frac{\vartheta}{L} + \frac{\vartheta\zeta_\mu^2}{L}\right)\mathbb{E}\left(F\left(\boldsymbol{g}_\mu\right)-F\left(\boldsymbol{g}^*\right)\right). \quad (20)$$

Proof: If $E_{i,\mu} \leq \gamma$, we have $\mathbb{1}_{\{E_{i,\mu}\leq\gamma\}} = 1$ and hence $\mathbb{E}(A) = K$. Substituting $\mathbb{E}(A) = K$ into (18), (20) can be obtained. This completes the proof. $\quad\square$

From Corollary 2, we can see that if $E_{i,\mu} \leq \gamma$, the convergence accuracy only depends on ζ_μ^1 and ζ_μ^2. Hence, this is an upper bound of the expected convergence of the proposed FL algorithm.

Theorem 1 derives the convergence accuracy and rate for the proposed FL algorithm that uses full gradient descent. Next, we derive the convergence accuracy and rate of the proposed FL algorithm when it uses a *stochastic gradient descent (SGD)* [25] method to update the local FL models. In particular, using the SGD update method, the update of local FL model at each user in (3) can be rewritten by

$$\boldsymbol{w}_{i,\mu+1} = \boldsymbol{g}\left(\boldsymbol{a}_\mu\right) - \lambda \sum_{k\in\mathcal{K}_{i,\mu}} \nabla f\left(\boldsymbol{g}\left(\boldsymbol{a}_\mu\right),\boldsymbol{x}_{ik},\boldsymbol{y}_{ik}\right), \quad (21)$$

where $\mathcal{K}_{i,\mu}$ is the subset of M training data samples selected from user i's training dataset $\mathcal{K}_i$. Hereinafter, we use $\nabla f_{ik}\left(\boldsymbol{g}_\mu\right)$ to represent $\nabla f\left(\boldsymbol{g}\left(\boldsymbol{a}_\mu\right),\boldsymbol{x}_{ik},\boldsymbol{y}_{ik}\right)$. The users and the BS will exchange FL model parameters once the users implement one SGD update in (21). Given (21), the convergence accuracy and rate of the proposed FL scheme when it uses the SGD update method is given in the following theorem.

Theorem 2: Given the optimal global FL model $\boldsymbol{g}^*$, the learning rate $\lambda = \frac{1}{L}$, and the subset of training data samples used to update each user i's local FL model at iteration μ, $\mathcal{K}_{i,\mu}$, the upper bound of $\mathbb{E}\left[F\left(\boldsymbol{g}_{\mu+1}\right)-F\left(\boldsymbol{g}^*\right)\right]$ for the proposed FL algorithm with SGD can be given by

$$\mathbb{E}\left[F\left(\boldsymbol{g}_{\mu+1}\right)-F\left(\boldsymbol{g}^*\right)\right] \leq \psi_\mu^1 + \psi_\mu^2\mathbb{E}\left(F\left(\boldsymbol{g}_\mu\right)-F\left(\boldsymbol{g}^*\right)\right). \quad (22)$$

where $\psi_\mu^1 = \frac{\zeta_\mu^1\left(9K-8\mathbb{E}\left(A^{\mathrm{S}}\right)\right)}{2LK}$ and $\psi_\mu^2 = \left(1 - \frac{\vartheta}{L} + \frac{\vartheta\zeta_\mu^2\left(9K-8\mathbb{E}\left(A^{\mathrm{S}}\right)\right)}{LK}\right)$ with $\mathbb{E}\left(A^{\mathrm{S}}\right) = \sum_{i=1}^{U} K_i\left(1-p_{i,\mu}\right)\mathbb{1}_{\{E_{i,\mu}\leq\gamma\}} + \sum_{i=1,i\neq i^*}^{U}\sum_{k\in\mathcal{K}_{i,\mu}} p_{i,\mu} + K_{i^*}$

Proof: See Appendix B. $\quad\square$

From Theorem 2, we can see that, in order to guarantee the convergence of the proposed FL with SGD, ψ_μ^2 must be smaller than 1 (i.e., $\psi_\mu^2 < 1$). Since ψ_μ^2 depends on the predicted local FL models, the local FL model of user i^*, user selection probability $p_{i,\mu}$, and the size of training dataset $\mathcal{K}_{i,\mu}$, we can adaptively adjust these parameters so as to guarantee the convergence of the proposed FL with SGD. For instance, we can adjust $p_{i,\mu}$ so as to maximize $\mathbb{E}\left(A^{\mathrm{S}}\right)$ hence decreasing ψ_μ^2.

2) Implementation Analysis: With regards to the implementation of the proposed algorithm, the BS must: a) determine the user selection policy, b) use an optimization algorithm to find the optimal RB allocation for each user, and c) use MLPs to predict the users' local FL models. To determine the user selection policy, the BS requires $\|\boldsymbol{e}_{i,\mu}\|$ of each user i at each iteration μ. Hence, each user must transmit $\|\boldsymbol{e}_{i,\mu}\|$ to the BS at each iteration. Since $\|\boldsymbol{e}_{i,\mu}\|$ is a scalar, the data size of which is much smaller than the data size of the local FL models that the users must transmit to the BS during each iteration. In consequence, we can ignore the overhead of each user transmitting $\|\boldsymbol{e}_{i,\mu}\|$ to the BS. To use an optimization algorithm for optimizing RB allocation, the BS needs to calculate the total transmission delay $l_i^{\mathrm{U}}\left(\boldsymbol{r}_{i,\mu}\right)+l_i^{\mathrm{D}}$ of each user i over each RB. The BS can use channel estimation methods to learn the signal-to-interference-plus-noise ratio over each RB so as to calculate $l_i^{\mathrm{U}}\left(\boldsymbol{r}_{i,\mu}\right)+l_i^{\mathrm{D}}$. To train the MLPs that are used for the predictions of users' local FL models, the BS will use the local FL models that are transmitted from the users who have RBs. These local FL models are originally used for the update of the global FL model and, hence, the BS does not require any additional information for training MLPs. Although our analytical results and the proposed solutions are considered for a network that consists of only one BS, one can easily extend them to a network that consists of multiple BSs by considering user association with different BSs.

3) Complexity Analysis: With regards to the complexity of the proposed algorithm, we first analyze the complexity of the interior-point method that is used to find the optimal RB allocation for each user. Let L_{O} be the number of iterations until the interior-point method converges. The complexity of the interior-point method is $\mathcal{O}\left(L_{\mathrm{O}}UR\right)$ [26]. Therefore, the complexity of the interior-point method, which depends on both the number of RBs and users, is linear. The complexity of training an MLP depends on the training data samples and the number of users. Since an MLP is trained by the BS which has enough computational resource, the training overhead can be ignored.

IV. SIMULATION RESULTS AND ANALYSIS

For our simulations, we consider a circular network area having a radius $r = 500$ m with one BS at its center servicing $U = 15$ uniformly distributed users. The value of the inter-cell interference at each RB is randomly selected from $\left[10^{-4}, 0.01\right]$. The FL algorithm is used for two learning tasks: a) the function fitting task where an FL algorithm is used for regression and b) the classification task where an FL algorithm is used to identify the handwritten digits from 0 to 9. For the classification task, each user trains an MLP using the MNIST dataset [27]. The size of neuron weight matrices are 784×50 and 50×10. The BS also implements an MLP for each user to predict its local FL model parameters. The MLPs are generated based on the MATLAB machine learning toolbox [28]. 1,000 handwritten digits are used to test the trained FL algorithms. The other parameters used in simulations are listed in Table I. For comparison purposes, we use two baselines: a) an FL algorithm that uses the
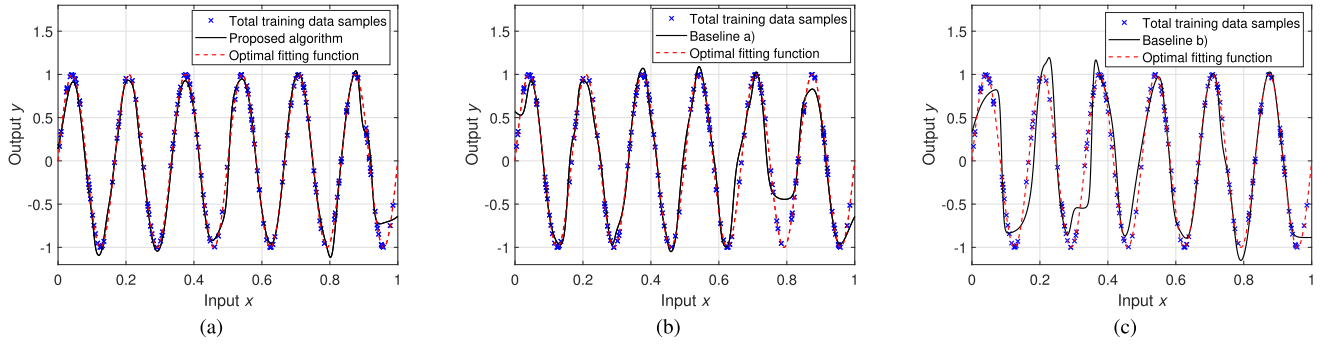
Fig. 3.   An example of implementing FL for regression.

TABLE II
SYSTEM PARAMETERS

Parameter	Value	Parameter	Value
α	2	N_0	-174 dBm/Hz
P	1 W	B	1 MHz
R	5	B^{D}	20 MHz
N	5	P_{B}	1 W
N_{out}	10	K_i	500
N_{in}	784	γ	0.01
W	5000	γ_{R}	5

proposed user association policy without the prediction of users' local FL models at each given learning step and b) a standard FL algorithm in [18] that randomly determines user selection and resource allocation without using MLP to estimate the local FL model parameters of each user at each given learning step. Hereinafter, *the proposed FL refers to the proposed FL algorithm that uses stochastic gradient descent method to train the local FL models*. The simulation results are averaged over a large number of independent runs.

In Fig. 3, we show an example of implementing FL for regression. In this example, FL is used to approximate the function $y = \sin(2\pi x)$ where $x \in [0, 1]$ is the FL input and y is the FL output. 15 users jointly implemented the considered FL algorithms and each user has 12 training data samples. To perform this learning task, each user implements a function fitting neural network [29] based FL algorithm. In Fig. 3, the total training data samples are the training data samples of all users ($15 \times 12 = 180$). The optimal fitting function is the target function that the FL algorithm is approximating. From Fig. 3, we can see that the proposed FL algorithm approximates the optimal fitting function better than baselines a) and b). This is because the proposed FL algorithm uses a probabilistic user selection scheme to select the users that transmit their local FL models to the BS at each FL iteration which improves the learning speed. Meanwhile, at each given learning step, the proposed FL algorithm uses MLPs to estimate the local FL models of the users that are not allocated any RBs for their local model parameter transmission so as to include more local FL models to generate the global FL model.

Fig. 4 shows an example of implementing the considered FL algorithms for the classification of handwritten digits from 0 to 9. From Fig. 4, we can see that, for a total of 25 handwritten digits, the proposed FL algorithm can correctly identify 22
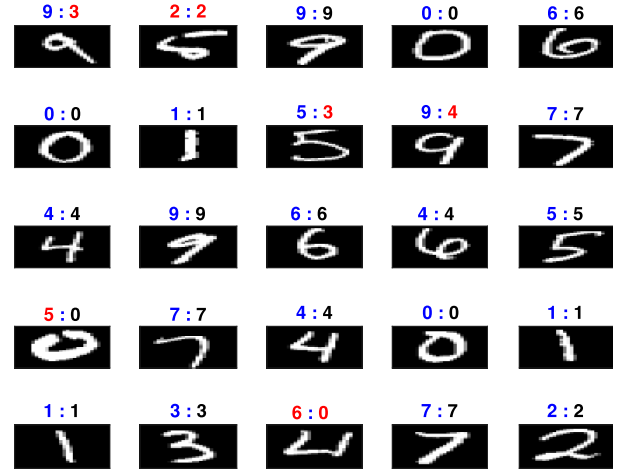

Fig. 4.   An example of implementing FL for classification of handwritten digits. In this figure, the blue digits that are displayed above the handwritten digits are the classification results of the proposed FL algorithm. The black digits that are displayed below the handwritten digits are the classification results of the standard FL algorithm. Meanwhile, the red digits represents the classification errors from all considered FL algorithms.

digits while the standard FL algorithm can identify 20 digits. This is because, at each given learning step, the proposed FL algorithm can use MLPs to estimate users' local FL model parameters. Hence, even though the number of RBs is limited, by employing our proposed FL algorithm, the BS can exploit all users' local FL models to generate the global FL model and, thus improving FL convergence time and decreasing FL training loss.

In Fig. 5, we show how the FL identification accuracy changes as time elapses. From this figure, we can see that, as time elapses, the FL identification accuracy of all considered algorithms increases. This is because the local FL models and the global FL model are trained by the users and the BS as time elapses. From Fig. 5(a), we can see that the proposed FL algorithm can reduce the number of iterations needed for convergence, by, respectively, up to 9% and 14% compared to baselines a) and b). The 9% gain stems from the fact that the proposed FL algorithm uses MLPs to estimate the local FL model parameters of the users that are not allocated any RBs for local FL model transmission at each given learning step. The 14% gain stems from the fact that the proposed FL algorithm uses the proposed probabilistic user selection scheme to
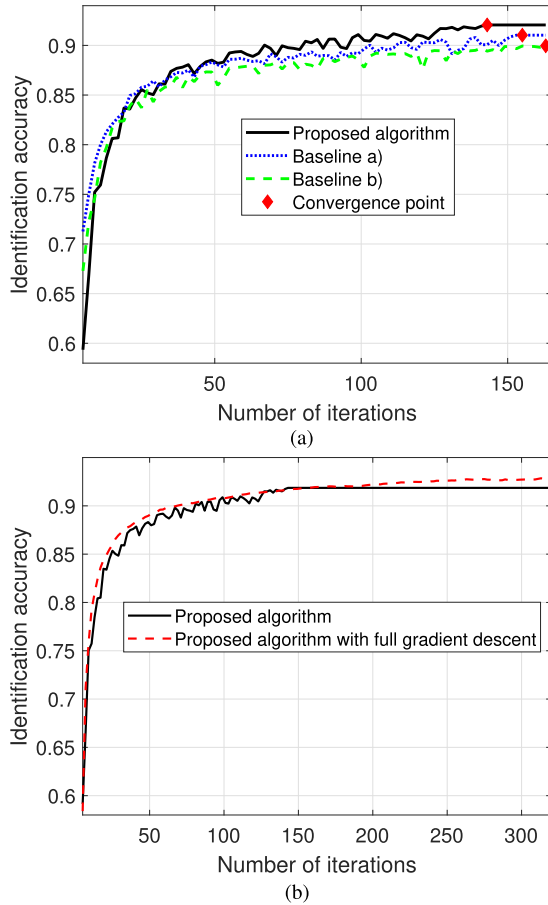
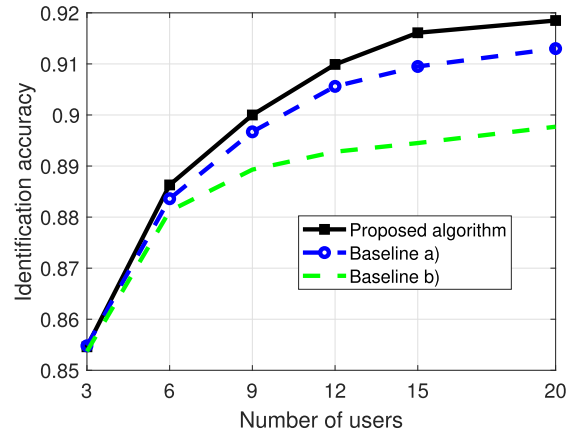Fig. 5.   Identification accuracy as the number of iterations varies.

Fig. 6.   Identification accuracy changes as the number of users varies.

Fig. 7.   Convergence time changes as the number of training data samples varies.

select the users for local FL model transmission and uses the ANNs to estimate the local FL model parameters of the users that do not RBs for local FL model transmission at each given learning step. From Fig. 5(b), we can see that the proposed algorithm converges faster than the proposed algorithm with full gradient descent. However, the proposed algorithm with full gradient descent can achieve better classification accuracy compared to the proposed algorithm. This is due to the fact that a full gradient descent method uses all training data samples to train the local FL models at each iteration while the SGD method uses a subset of training data samples to train the local FL models.

Fig. 6 shows how the identification accuracy changes as the number of users varies. In this figure, we can see that, as the number of users increases, the FL identification accuracy of all considered algorithms increases. This is because as the number of users increases, the number of data samples used for training FL increases. From Fig. 6, we can also see that, for a network with 20 users, the proposed FL algorithm can improve the identification accuracy by up to 1% and 3%, respectively, compared to baselines a) and b). These gains stem from the fact that, in the proposed FL algorithm, a probabilistic user selection scheme is developed for user selection and local FL model transmission. Meanwhile, to include additional local FL models to generate the global FL model, at each given learning step, the proposed FL algorithm uses ANNs to estimate the local FL model parameters of the users that are not allocated
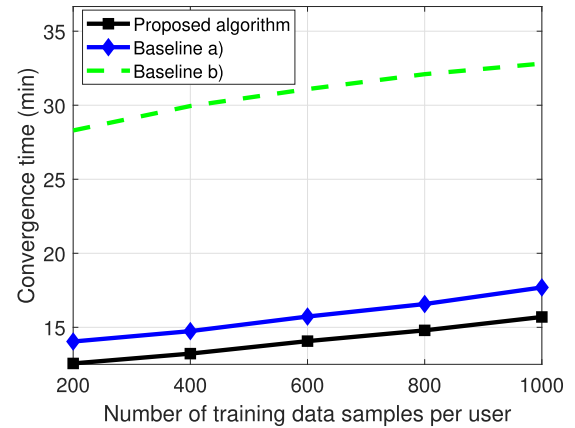
any RBs for local FL model transmission hence improving the classification accuracy. Fig. 6 also shows that, as the number of users increases, the gap between the identification accuracy resulting from the proposed FL algorithm and the baselines increases. This is because, for the considered baselines, as the number of users increases, the number of users that can transmit their local FL models to the BS remains the same due to the limited number of RBs. In contrast, the proposed FL algorithm can use MLPs to estimate the local FL models of the users that are not allocated any RBs for transmitting their local model parameters at each given learning step and, hence, include more local FL models to generate the global FL model.

In Fig. 7, we show how the convergence time changes as the number of training data samples varies. Fig. 7 shows that, as the number of training data samples increases, the convergence time of all considered FL algorithms increases. This is due to the fact that all of the considered FL algorithms use a stochastic gradient descent method to train their local FL models. Hence, as the number of training data samples increases, the various FL algorithms need to use more time to sample the local datasets. Fig. 7 also shows that, for a scenario in which each user has 200 training data samples, the proposed FL algorithm can reduce convergence time by up to 11% and 56%, compared to baselines a) and b). These gains stem from the fact that the proposed FL algorithm uses ANNs to estimate
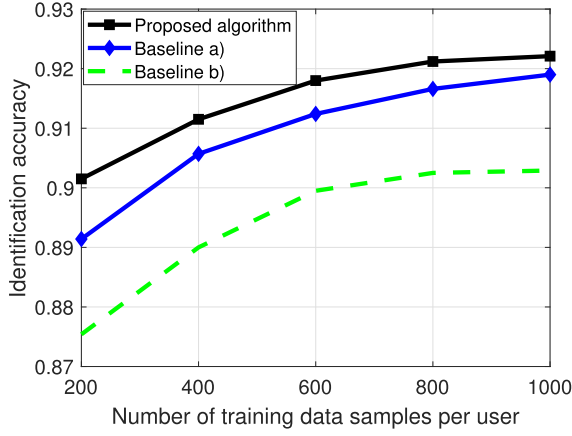
Fig. 8. Identification accuracy changes as the number of training data samples varies.
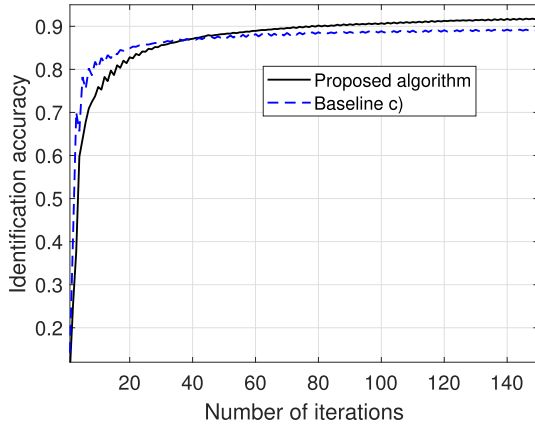

Fig. 9. Convergence of the FL algorithms.

the local FL models of the users that are not allocated any RBs for transmitting their local model parameters at each given learning step, and uses the probabilistic user selection scheme to determine the users that will transmit the local FL model parameters to the BS. In addition, the proposed algorithm optimizes RB allocation at each iteration.

Fig. 8 shows how the handwritten digit identification accuracy changes as the number of training data samples varies. From Fig. 8, we can see that, as the number of training data samples increases from 200 to 800, the identification accuracy of all considered algorithms increases. This is because, as the number of training data samples increases, each user can use more data samples to train the local FL model thus improving the identification accuracy of the FL algorithms. As the number of training data samples continues to increase, the identification accuracy of all considered algorithms improves slowly. This is because 800 training data samples may include all features of the MNIST dataset. Fig. 8 also shows that, as the number of training data samples increases, the gap between the identification accuracy resulting from the proposed FL algorithm and baseline a) decreases. This is due to the fact that, as the number of training data samples per user increases, each local FL model is trained by a dataset that contains all features of MNIST dataset, and, hence, the BS can use fewer local FL models to generate the global FL model.

In Fig. 9, we show the convergence of both the proposed FL algorithm and baseline c). In baseline c), the users that transmit local FL models to the BS are selected based on our proposed user selection scheme. For unselected users, the BS will use previous received local FL models for global FL model updates as done in LAG algorithm of [16]. From this figure, we can see that, the proposed FL algorithm can improve the identification accuracy by about 1.7% compared to baseline c). This is because the proposed FL algorithm estimates the local FL models of the users that are not allocated any RBs.

V. CONCLUSION

In this paper, we have developed a novel framework that enables the implementation of FL over wireless networks. The proposed FL framework can be used for various real-world applications such as mobile keyboard prediction, network traffic analysis and prediction, device monitoring, and user behavior analysis for virtual reality users. We have formulated an optimization problem that jointly considers user selection and resource allocation for the joint minimization of the FL convergence time and the FL training loss. To solve this problem, we have proposed a probablistic user selection scheme that allows the users whose local FL models have large impacts on the global FL model to associate with the BS with high probability. Given the user selection policy, the uplink RB allocation is determined. To further improve the FL convergence speed, we have studied the use of MLPs to estimate the local FL models of the users that are not allocated any RBs at each given learning step. Simulation results have shown that the proposed FL algorithm yields significant improvements in terms of convergence time compared to a standard FL algorithm.

APPENDIX

A. Proof of Theorem 1

Since the loss function $f\left(\boldsymbol{g}_{\mu}, \boldsymbol{x}_{ik}, y_{ik}\right)$ is strongly convex and twice-continuously differentiable, we have the following observations:

- $\nabla F\left(\boldsymbol{g}_{\mu}\right)$ is uniformly Lipschitz continuous with respect to $\boldsymbol{g}_{\mu}$ [30] and, hence, we can find a positive constant L, such that

$$\|\nabla F\left(\boldsymbol{g}_{\mu+1}\right) - \nabla F\left(\boldsymbol{g}_{\mu}\right)\| \le L\|\boldsymbol{g}_{\mu+1} - \boldsymbol{g}_{\mu}\|, \quad (23)$$

where $\|\boldsymbol{g}_{\mu+1} - \boldsymbol{g}_{\mu}\|$ is the norm of $\boldsymbol{g}_{\mu+1} - \boldsymbol{g}_{\mu}$.

- Since $F\left(\boldsymbol{g}\right)$ is strongly convex, we have

$$F\left(\boldsymbol{g}_{\mu+1}\right) \ge F\left(\boldsymbol{g}_{\mu}\right) + \left(\boldsymbol{g}_{\mu+1} - \boldsymbol{g}_{\mu}\right)^{T} \nabla F\left(\boldsymbol{g}_{\mu}\right) + \frac{\vartheta}{2}\|\boldsymbol{g}_{\mu+1} - \boldsymbol{g}_{\mu}\|^{2}. \quad (24)$$

where ϑ is a positive parameter. By minimizing both sides of (24) with respect to $\boldsymbol{g}_{\mu+1}$, we have

$$F\left(\boldsymbol{g}^{*}\right) \ge F\left(\boldsymbol{g}_{\mu}\right) - \frac{1}{2\vartheta}\|\nabla F\left(\boldsymbol{g}_{\mu}\right)\|^{2}. \quad (25)$$

- Since $F\left(\boldsymbol{g}\right)$ is twice-continuously differentiable, we have

$$\vartheta \boldsymbol{I} \preceq \nabla^{2} F\left(\boldsymbol{g}\right) \preceq L\boldsymbol{I}. \quad (26)$$

- We also assume that $\|\nabla F_i\left(\boldsymbol{g}_\mu\right)\|^2 \leq \zeta_\mu^1 + \zeta_\mu^2\|\nabla F\left(\boldsymbol{g}_\mu\right)\|^2$ with $\zeta_{i,\mu}^1 \geq 0$ and $\zeta_{i,\mu}^2 \geq 1$.

Using the second-order Taylor expansion, $F\left(\boldsymbol{g}_{\mu+1}\right)$ can be rewritten as follows:

$$
\begin{aligned}
&F\left(\boldsymbol{g}_{\mu+1}\right) \\
&= F\left(\boldsymbol{g}_\mu\right) + \left(\boldsymbol{g}_{\mu+1} - \boldsymbol{g}_\mu\right)^T \nabla F\left(\boldsymbol{g}_\mu\right) \\
&\quad + \frac{1}{2}\left(\boldsymbol{g}_{\mu+1} - \boldsymbol{g}_\mu\right)^T \nabla^2 F\left(\boldsymbol{g}\right)\left(\boldsymbol{g}_{\mu+1} - \boldsymbol{g}_\mu\right), \\
&\overset{(a)}{\leq} F\left(\boldsymbol{g}_\mu\right) + \left(\boldsymbol{g}_{\mu+1} - \boldsymbol{g}_\mu\right)^T \nabla F\left(\boldsymbol{g}_\mu\right) + \frac{L}{2}\|\boldsymbol{g}_{\mu+1} - \boldsymbol{g}_t\|^2,
\end{aligned}
\tag{27}
$$

where (a) stems from the fact that $\nabla^2 F\left(\boldsymbol{g}\right) \preceq L\boldsymbol{I}$. Based on (2) and (3), we have $\boldsymbol{g}_{\mu+1} - \boldsymbol{g}_\mu = \lambda\left(\nabla F\left(\boldsymbol{g}_\mu\right) - \boldsymbol{e}_\mu\right)$ where $\boldsymbol{e}_\mu$ is a gradient deviation caused by the users that do not transmit their local FL models to the BS at iteration μ and the prediction errors of the local FL models that are estimated by the BS at iteration μ. In particular, $\boldsymbol{e}_\mu$ can be expressed as

$$
\begin{aligned}
&\boldsymbol{e}_\mu \\
&= \nabla F\left(\boldsymbol{g}_\mu\right) - \frac{\sum\limits_{i=1}^{U} K_i a_{i,\mu} \nabla F_i\left(\boldsymbol{g}_\mu\right)}{\sum\limits_{i=1}^{U} K_i\left(1 - a_{i,\mu}\right)\mathbb{1}_{\{E_{i,\mu} \leq \gamma\}} + \sum\limits_{i=1}^{U} K_i a_{i,\mu}} \\
&\quad - \frac{\sum\limits_{i=1}^{U} K_i\left(1 - a_{i,\mu}\right)\left(\nabla F_i\left(\boldsymbol{g}_\mu\right) + \nabla \hat{F}_i\left(\boldsymbol{g}_\mu\right)\right)\mathbb{1}_{\{E_{i,\mu} \leq \gamma\}}}{\sum\limits_{i=1}^{U} K_i\left(1 - a_{i,\mu}\right)\mathbb{1}_{\{E_{i,\mu} \leq \gamma\}} + \sum\limits_{i=1}^{U} K_i a_{i,\mu}},
\end{aligned}
\tag{28}
$$

where $\frac{1}{K}\sum\limits_{i=1}^{U} K_i\left(1 - a_{i,\mu}\right)\left(\nabla F_i\left(\boldsymbol{g}_\mu\right) + \nabla \hat{F}_i\left(\boldsymbol{g}_\mu\right)\right)\mathbb{1}_{\{E_{i,\mu} \leq \gamma\}}$ is the sum of the gradients in the local FL models that are estimated by the BS while $\frac{1}{K}\sum\limits_{i=1}^{U} K_i a_{i,\mu} \nabla F_i\left(\boldsymbol{g}_\mu\right)$ is the sum of the gradients in the local FL models that are transmitted from the users.

Let the learning rate $\lambda = \frac{1}{L}$. Based on (27), $\mathbb{E}\left[F\left(\boldsymbol{g}_{\mu+1}\right)\right]$ can be expressed as

$$
\begin{aligned}
&\mathbb{E}\left[F\left(\boldsymbol{g}_{\mu+1}\right)\right] \\
&\leq \mathbb{E}\left(F\left(\boldsymbol{g}_\mu\right)\right) - \frac{1}{L}\left(\|\nabla F\left(\boldsymbol{g}_\mu\right)\|^2 - \boldsymbol{e}_\mu^T \nabla F\left(\boldsymbol{g}_\mu\right)\right) \\
&\quad + \frac{1}{2L}\left(\|\nabla F\left(\boldsymbol{g}_\mu\right)\|^2 - 2\boldsymbol{e}_\mu^T \nabla F\left(\boldsymbol{g}_\mu\right) + \|\boldsymbol{e}_\mu\|^2\right), \\
&= \mathbb{E}\left(F\left(\boldsymbol{g}_\mu\right)\right) - \frac{1}{2L}\|\nabla F\left(\boldsymbol{g}_\mu\right)\|^2 + \frac{1}{2L}\mathbb{E}\left(\|\boldsymbol{e}_\mu\|^2\right).
\end{aligned}
\tag{29}
$$

Let $\mathcal{N}_1 = \{a_{i,\mu} = 1 | i \in \mathcal{U}\}$ be the set of selected users that transmit their local FL models to the BS, $\mathcal{N}_2 = \{a_{i,\mu} = 0, E_{i,\mu} \leq \gamma | i \in \mathcal{U}\}$ be the set of users that the BS can accurately estimate their local FL models, and $\mathcal{N}_3 = \{i \in \mathcal{U} | i \notin \mathcal{N}_1, i \notin \mathcal{N}_2\}$. Meanwhile, let $A = \sum\limits_{i=1}^{U} K_i\left(1 - a_{i,\mu}\right)\mathbb{1}_{\{E_{i,\mu} \leq \gamma\}} + \sum\limits_{i=1}^{U} K_i a_{i,\mu}$. Given (28), $\mathbb{E}\left(\|\boldsymbol{e}_\mu\|^2\right)$ can be expressed in (30). Since $\|\nabla F_i\left(\boldsymbol{g}_\mu\right)\|^2 \leq \zeta_\mu^1 + \zeta_\mu^2\|\nabla F\left(\boldsymbol{g}_\mu\right)\|^2$, $\|\nabla \hat{F}_i\left(\boldsymbol{g}_\mu\right)\|^2 \leq \zeta_\mu^1 + \zeta_\mu^2\|\nabla F\left(\boldsymbol{g}_\mu\right)\|^2$,

(30) can be simplified by

$$
\begin{aligned}
&\mathbb{E}\left(\|\boldsymbol{e}_\mu\|^2\right) \\
&\leq \mathbb{E}\left(\frac{(K-A)\sqrt{\zeta_\mu^1 + \zeta_\mu^2\|\nabla F\left(\boldsymbol{g}_\mu\right)\|^2}}{K}\right. \\
&\quad + \frac{\sum\limits_{i \in \mathcal{N}_2} K_i\left\|\nabla \hat{F}_i\left(\boldsymbol{g}_\mu\right)\right\|}{A} \\
&\quad \left. + \frac{(K-A)\sqrt{\zeta_\mu^1 + \zeta_\mu^2\|\nabla F\left(\boldsymbol{g}_\mu\right)\|^2}}{K}\right)^2 \\
&\leq \mathbb{E}\left(\frac{2(K-A)\sqrt{\zeta_\mu^1 + \zeta_\mu^2\|\nabla F\left(\boldsymbol{g}_\mu\right)\|^2}}{K}\right. \\
&\quad \left. + \sqrt{\zeta_\mu^1 + \zeta_\mu^2\|\nabla F\left(\boldsymbol{g}_\mu\right)\|^2}\right)^2, \\
&= \mathbb{E}\left(\frac{4(K-A)^2\left(\zeta_\mu^1 + \zeta_\mu^2\|\nabla F\left(\boldsymbol{g}_\mu\right)\|^2\right)}{K^2}\right. \\
&\quad + \zeta_\mu^1 + \zeta_\mu^2\|\nabla F\left(\boldsymbol{g}_\mu\right)\|^2 \\
&\quad \left. + \frac{4(K-A)\left(\zeta_\mu^1 + \zeta_\mu^2\|\nabla F\left(\boldsymbol{g}_\mu\right)\|^2\right)}{K}\right), \\
&\leq \mathbb{E}\left(\frac{(9K - 8A)\left(\zeta_\mu^1 + \zeta_\mu^2\|\nabla F\left(\boldsymbol{g}_\mu\right)\|^2\right)}{K}\right) \\
&= \frac{(9K - 8\mathbb{E}(A))}{K}\left(\zeta_\mu^1 + \zeta_\mu^2\|\nabla F\left(\boldsymbol{g}_\mu\right)\|^2\right).
\end{aligned}
\tag{31}
$$

Since $\mathbb{E}\left(1 - a_{i,\mu}\right) = 1 - p_{i,\mu}$ and $\mathbb{E}\left(a_{i,\mu}\right) = p_{i,\mu}$, we have $\mathbb{E}(A) = \sum\limits_{i=1}^{U} K_i\left(1 - p_{i,\mu}\right)\mathbb{1}_{\{E_{i,\mu} \leq \gamma\}} + \sum\limits_{i=1}^{U} K_i p_{i,\mu}$.

Substitute (31) into (29), we have

$$
\begin{aligned}
&\mathbb{E}\left[F\left(\boldsymbol{g}_{\mu+1}\right)\right] \\
&\leq \mathbb{E}\left(F\left(\boldsymbol{g}_t\right)\right) + \frac{\zeta_\mu^1\left(9K - 8\mathbb{E}(A)\right)}{2LK} \\
&\quad - \frac{1}{2L}\left(1 - \frac{\zeta_\mu^2\left(9K - 8\mathbb{E}(A)\right)}{K}\right)\|\nabla F\left(\boldsymbol{g}_t\right)\|^2.
\end{aligned}
\tag{32}
$$

Subtract $\mathbb{E}\left[F\left(\boldsymbol{g}^*\right)\right]$ in both sides of (32), we have

$$
\begin{aligned}
&\mathbb{E}\left[F\left(\boldsymbol{g}_{\mu+1}\right) - F\left(\boldsymbol{g}^*\right)\right] \\
&\leq \mathbb{E}\left(F\left(\boldsymbol{g}_\mu\right) - F\left(\boldsymbol{g}^*\right)\right) + \frac{\zeta_\mu^1\left(9K - 8\mathbb{E}(A)\right)}{2LK} \\
&\quad - \frac{1}{2L}\left(1 - \frac{\zeta_\mu^2\left(9K - 8\mathbb{E}(A)\right)}{K}\right)\|\nabla F\left(\boldsymbol{g}_t\right)\|^2.
\end{aligned}
\tag{33}
$$

Given (25), (33) can be rewritten as

$$
\begin{aligned}
&\mathbb{E}\left[F\left(\boldsymbol{g}_{\mu+1}\right) - F\left(\boldsymbol{g}^*\right)\right] \\
&\leq \frac{\zeta_\mu^1\left(9K - 8\mathbb{E}(A)\right)}{2LK} \\
&\quad + \left(1 - \frac{\vartheta}{L} + \frac{\vartheta\zeta_\mu^2\left(9K - 8\mathbb{E}(A)\right)}{LK}\right)\mathbb{E}\left(F\left(\boldsymbol{g}_\mu\right) - F\left(\boldsymbol{g}^*\right)\right).
\end{aligned}
\tag{34}
$$

This completes the proof.

$$
\begin{aligned}
\mathbb{E}\left(\|\boldsymbol{e}_\mu\|^2\right) = \mathbb{E}&\left(\left\|\nabla F\left(\boldsymbol{g}_\mu\right) - \frac{\sum\limits_{i=1}^{U} K_i\left(1-a_{i,\mu}\right)\left(\nabla F_i\left(\boldsymbol{g}_\mu\right)+\nabla\hat{F}_i\left(\boldsymbol{g}_\mu\right)\right)\mathbb{1}_{\{E_{i,\mu}\leq\gamma\}}}{\sum\limits_{i=1}^{U} K_i\left(1-a_{i,\mu}\right)\mathbb{1}_{\{E_{i,\mu}\leq\gamma\}}+\sum\limits_{i=1}^{U} K_i a_{i,\mu}}\right.\right.\\
&\left.\left.-\frac{\sum\limits_{i=1}^{U} K_i a_{i,\mu}\nabla F_i\left(\boldsymbol{g}_\mu\right)}{\sum\limits_{i=1}^{U} K_i\left(1-a_{i,\mu}\right)\mathbb{1}_{\{E_{i,\mu}\leq\gamma\}}+\sum\limits_{i=1}^{U} K_i a_{i,\mu}}\right\|^2\right),\\
= \mathbb{E}&\left(\left\|-\frac{(K-A)\sum\limits_{i\in\mathcal{N}_1} K_i\nabla F_i\left(\boldsymbol{g}_\mu\right)}{KA}-\frac{(\text{K-A})\sum\limits_{i\in\mathcal{N}_2} K_i\nabla F_i\left(\boldsymbol{g}_\mu\right)}{KA}\right.\right.\\
&\left.\left.-\frac{\sum\limits_{i\in\mathcal{N}_2} K_i\nabla\hat{F}_i\left(\boldsymbol{g}_\mu\right)}{A}+\frac{\sum\limits_{i\in\mathcal{N}_3} K_i\nabla F_i\left(\boldsymbol{g}_\mu\right)}{K}\right\|^2\right),\\
\leq \mathbb{E}&\left(\frac{(K-A)\sum\limits_{i\in\mathcal{N}_1} K_i\left\|\nabla F_i\left(\boldsymbol{g}_\mu\right)\right\|}{KA}+\frac{(\text{K-A})\sum\limits_{i\in\mathcal{N}_2} K_i\left\|\nabla F_i\left(\boldsymbol{g}_\mu\right)\right\|}{KA}\right.\\
&\left.+\frac{\sum\limits_{i\in\mathcal{N}_2} K_i\left\|\nabla\hat{F}_i\left(\boldsymbol{g}_\mu\right)\right\|}{A}+\frac{\sum\limits_{i\in\mathcal{N}_3} K_i\left\|\nabla F_i\left(\boldsymbol{g}_\mu\right)\right\|}{K}\right)^2.
\end{aligned}
\tag{30}
$$

B. Proof of Theorem 2

To prove Theorem 2, we only need to derive $\mathbb{E}\left(\|\boldsymbol{e}_\mu\|^2\right)$ in (29). For the proposed FL algorithm with SGD, the gradient deviation e_μ^{SGD} caused by the users that do not transmit their local FL models and the SGD training method is given by

$$
\begin{aligned}
&\boldsymbol{e}_\mu^{\text{SGD}}\\
&= \nabla F\left(\boldsymbol{g}_\mu\right)\\
&-\frac{\sum\limits_{i=1}^{U}(1-a_{i,\mu})K_i\left(\nabla F_i\left(\boldsymbol{g}_\mu\right)+\nabla\hat{F}_i\left(\boldsymbol{g}_\mu\right)\right)\mathbb{1}_{\{E_{i,\mu}\leq\gamma\}}}{\sum\limits_{i=1}^{U} K_i\left(1-a_{i,\mu}\right)\mathbb{1}_{\{E_{i,\mu}\leq\gamma\}}+\sum\limits_{i=1,i\neq i^*}^{U}\sum\limits_{k\in\mathcal{K}_{i,\mu}} a_{i,\mu}+K_{i^*}}\\
&-\frac{\sum\limits_{i=1,i\neq i^*}^{U}\sum\limits_{k\in\mathcal{K}_{i,\mu}} a_{i,\mu}\nabla f_{ik}\left(\boldsymbol{g}_\mu\right)-K_i^*\nabla F_{i^*}\left(\boldsymbol{g}_\mu\right)}{\sum\limits_{i=1}^{U} K_i\left(1-a_{i,\mu}\right)\mathbb{1}_{\{E_{i,\mu}\leq\gamma\}}+\sum\limits_{i=1,i\neq i^*}^{U}\sum\limits_{k\in\mathcal{K}_{i,\mu}} a_{i,\mu}+K_{i^*}},
\end{aligned}
\tag{35}
$$

In (35), $\sum\limits_{i=1,i\neq i^*}^{U} a_{i,\mu}\nabla f\left(\boldsymbol{g}\left(\boldsymbol{a}_\mu\right)\right)$ is the sum of the local FL models that are transmitted from the users, except user i^*, that have RBs. Since the BS needs to use the local FL model parameters of user i^* for the predictions of other users' local FL models, user i^*'s local FL model must be trained by all of its collected data. Hence, $K_i^*\nabla F_{i^*}\left(\boldsymbol{g}_\mu\right)$ is the local model that is transmitted by user i^*. Given (35) and $\|\nabla F_i\left(\boldsymbol{g}_\mu\right)\|^2, \|\nabla\hat{F}_i\left(\boldsymbol{g}_\mu\right)\|^2, \|\nabla f_{ik}\left(\boldsymbol{g}_\mu\right)\|^2 \leq \zeta_\mu^1 +$

$\zeta_\mu^2\|\nabla F\left(\boldsymbol{g}_\mu\right)\|^2$, $\|e_\mu^{\text{SGD}}\|$ can be expressed by

$$
\begin{aligned}
\|e_\mu^{\text{SGD}}\| \leq\; & \frac{\left(K-A^{\text{S}}\right)K_i^*\left\|\nabla F_{i^*}\left(\boldsymbol{g}_\mu\right)\right\|}{KA^{\text{S}}}\\
&+\frac{\left(K-A^{\text{S}}\right)\sum\limits_{i\in\mathcal{N}_2} K_i\left\|\nabla F_i\left(\boldsymbol{g}_\mu\right)\right\|}{KA^{\text{S}}}\\
&+\frac{\sum\limits_{i\in\mathcal{N}_2} K_i\left\|\nabla\hat{F}_i\left(\boldsymbol{g}_\mu\right)\right\|}{A^{\text{S}}}\\
&+\frac{\left(K-A^{\text{S}}\right)\sum\limits_{i\in\mathcal{N}_1}\sum\limits_{k\in\mathcal{K}_{i,\mu}}\left\|\nabla f_{ik}\left(\boldsymbol{g}_\mu\right)\right\|}{KA^{\text{S}}}\\
&+\frac{\sum\limits_{i\in\mathcal{N}_3} K_i\left\|\nabla F_i\left(\boldsymbol{g}_\mu\right)\right\|}{K}\\
&+\frac{\sum\limits_{i\in\mathcal{N}_1}\sum\limits_{k\in\mathcal{K}_i/\mathcal{K}_{i,\mu}}\left\|\nabla f_{ik}\left(\boldsymbol{g}_\mu\right)\right\|}{K},\\
\leq\; & \frac{2\left(K-A^{\text{S}}\right)\sqrt{\zeta_\mu^1+\zeta_\mu^2\|\nabla F\left(\boldsymbol{g}_\mu\right)\|^2}}{K}\\
&+\sqrt{\zeta_\mu^1+\zeta_\mu^2\|\nabla F\left(\boldsymbol{g}_\mu\right)\|^2},
\end{aligned}
\tag{36}
$$

where the second inequality stems from the fact that
$\frac{(K-A^{\text{S}})K_i^*\|\nabla F_{i^*}(\boldsymbol{g}_\mu)\|}{KA^{\text{S}}} + \frac{(K-A^{\text{S}})\sum\limits_{i\in\mathcal{N}_2} K_i\|\nabla F_i(\boldsymbol{g}_\mu)\|}{KA^{\text{S}}} +$
$\frac{(K-A^{\text{S}})\sum\limits_{i\in\mathcal{N}_1}\sum\limits_{k\in\mathcal{K}_{i,\mu}}\|\nabla f_{ik}(\boldsymbol{g}_\mu)\|}{KA^{\text{S}}} = \frac{\sum\limits_{i\in\mathcal{N}_3} K_i\|\nabla F_i(\boldsymbol{g}_\mu)\|}{K} +$

$$\frac{\sum_{i \in \mathcal{N}_1} \sum_{k \in \mathcal{K}_i / \mathcal{K}_{i,\mu}} \|\nabla f_{ik}(\boldsymbol{g}_\mu)\|}{K} = \frac{(K - A^S)\sqrt{\zeta_\mu^1 + \zeta_\mu^2 \|\nabla F(\boldsymbol{g}_\mu)\|^2}}{K} \text{ and}$$

$$\frac{\sum_{i \in \mathcal{N}_2} K_i \|\nabla \hat{F}_i(\boldsymbol{g}_\mu)\|}{A^S} \leq \sqrt{\zeta_\mu^1 + \zeta_\mu^2 \|\nabla F(\boldsymbol{g}_\mu)\|^2}.$$

Based on (31) and (36), we have $\mathbb{E}\left(\|e_\mu^{\text{SGD}}\|^2\right) = \frac{(9K - 8\mathbb{E}(A^S))}{K}\left(\zeta_\mu^1 + \zeta_\mu^2 \|\nabla F(\boldsymbol{g}_\mu)\|^2\right)$ with $\mathbb{E}(A^S) = \sum_{i=1}^{U} K_i (1 - p_{i,\mu}) \mathbb{1}_{\{E_{i,\mu} \leq \gamma\}} + \sum_{i=1, i \neq i^*}^{U} \sum_{k \in \mathcal{K}_{i,\mu}} p_{i,\mu} + K_{i^*}.$

Based on (32) to (37), we have

$$\mathbb{E}\left[F(\boldsymbol{g}_{\mu+1}) - F(\boldsymbol{g}^*)\right]$$
$$\leq \frac{\zeta_\mu^1 (9K - 8\mathbb{E}(A^S))}{2LK}$$
$$+ \left(1 - \frac{\vartheta}{L} + \frac{\vartheta \zeta_\mu^2 (9K - 8\mathbb{E}(A^S))}{LK}\right) \mathbb{E}\left(F(\boldsymbol{g}_\mu) - F(\boldsymbol{g}^*)\right).$$

$$(37)$$

This completes the proof.

REFERENCES

[1] M. Chen, H. V. Poor, W. Saad, and S. Cui, "Convergence time minimization of federated learning over wireless networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Dublin, Ireland, Jun. 2020, pp. 1–6.

[2] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems," *IEEE Netw.*, vol. 34, no. 3, pp. 134–142, May/Jun. 2020.

[3] K. Bonawitz *et al.*, "Towards federated learning at scale: System design," in *Proc. Syst. Mach. Learn. Conf.*, Stanford, CA, USA, 2019.

[4] G. Zhu, D. Liu, Y. Du, C. You, J. Zhang, and K. Huang, "Toward an intelligent edge: Wireless communication meets machine learning," *IEEE Commun. Mag.*, vol. 58, no. 1, pp. 19–25, Jan. 2020.

[5] H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Blockchained on-device federated learning," *IEEE Commun. Lett.*, vol. 24, no. 6, pp. 1279–1283, Jun. 2020.

[6] S. Wang *et al.*, "Adaptive federated learning in resource constrained edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1205–1221, Jun. 2019.

[7] 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.*, early access, Oct. 1, 2020, doi: 10.1109/TWC.2020.3024629.

[8] N. H. Tran, W. Bao, A. Zomaya, M. N. H. Nguyen, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Paris, France, Apr. 2019, pp. 1387–1395.

[9] 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.*, early access, Nov. 19, 2020, doi: 10.1109/TWC.2020.3037554.

[10] M. Chen, O. Semiari, W. Saad, X. Liu, and C. Yin, "Federated echo state learning for minimizing breaks in presence in wireless virtual reality networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 177–191, Jan. 2020.

[11] T. T. Vu, D. T. Ngo, N. H. Tran, H. Q. Ngo, M. N. Dao, and R. H. Middleton, "Cell-free massive MIMO for wireless federated learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 10, pp. 6377–6392, Oct. 2020.

[12] J. Ren, G. Yu, and G. Ding, "Accelerating DNN training in wireless federated edge learning systems," *IEEE J. Sel. Areas Commun.*, early access, Nov. 9, 2020, doi: 10.1109/JSAC.2020.3036971.

[13] H. H. Yang, Z. Liu, T. Q. S. Quek, and H. V. Poor, "Scheduling policies for federated learning in wireless networks," *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 317–333, Jan. 2020.

[14] S. Feng, D. Niyato, P. Wang, D. I. Kim, and Y.-C. Liang, "Joint service pricing and cooperative relay communication for federated learning," in *Proc. Int. Conf. Internet Things (iThings), IEEE Green Comput. Commun. (GreenCom), IEEE Cyber, Phys. Social Comput. (CPSCom), IEEE Smart Data (SmartData)*, Atlanta, GA, USA, Jul. 2019, pp. 815–820.

[15] N. Skatchkovsky, H. Jang, and O. Simeone, "Federated neuromorphic learning of spiking neural networks for low-power edge intelligence," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Barcelona, Spain, May 2020, pp. 8524–8528.

[16] T. Chen, G. Giannakis, T. Sun, and W. Yin, "LAG: Lazily aggregated gradient for communication-efficient distributed learning," in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, Dec. 2018, pp. 5050–5060.

[17] M. M. Amiri and D. Gunduz, "Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Paris, France, Jul. 2019, pp. 1–5.

[18] J. Konečný, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," 2016, *arXiv:1610.02527*. [Online]. Available: http://arxiv.org/abs/1610.02527

[19] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," 2018, *arXiv:1812.06127*. [Online]. Available: http://arxiv.org/abs/1812.06127

[20] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, "Artificial neural networks-based machine learning for wireless networks: A tutorial," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3039–3071, 4th Quart., 2019.

[21] J. Konecny, H. B. McMahan, F. X. Yu, P. Richtarik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," in *Proc. Adv. Neural Inf. Process. Syst. Workshop Private Multi-Party Mach. Learn.*, Barcelona, Spain, Dec. 2016.

[22] Q. Ye, B. Rong, Y. Chen, M. Al-Shalash, C. Caramanis, and J. G. Andrews, "User association for load balancing in heterogeneous cellular networks," *IEEE Trans. Wireless Commun.*, vol. 12, no. 6, pp. 2706–2716, Jun. 2013.

[23] Y. Zhang, "Solving large-scale linear programs by interior-point methods under the MATLAB environment," *Optim. Methods Softw.*, vol. 10, no. 1, pp. 1–31, Jan. 1998.

[24] W. Wu, J. Wang, M. Cheng, and Z. Li, "Convergence analysis of online gradient method for BP neural networks," *Neural Netw.*, vol. 24, no. 1, pp. 91–98, Jan. 2011.

[25] M. Zinkevich, M. Weimer, L. Li, and A. J. Smola, "Parallelized stochastic gradient descent," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, Dec. 2010, pp. 2595–2603.

[26] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[27] Y. LeCun. *The MNIST Database of Handwritten Digits*. Accessed: Jan. 2020. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[28] *Classify MNIST Digits Using a Feedforward Neural Network With MATLAB*. Accessed: Jan. 2020. [Online]. Available: https://www.machinelearningtutorial.net/2017/01/14/matlab-mnist-dataset-tutorial/#m1

[29] MATLAB. *Function Fitting Neural Networks*. Accessed: Jan. 2020. [Online]. Available: https://www.mathworks.com/help/deeplearning/ref/fitnet.html.

[30] M. P. Friedlander and M. Schmidt, "Hybrid deterministic-stochastic methods for data fitting," *SIAM J. Sci. Comput.*, vol. 34, no. 3, pp. A1380–A1405, Jan. 2012.

Mingzhe Chen (Member, IEEE) received the Ph.D. degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 2019. From 2016 to 2019, he was a Visiting Researcher with the Department of Electrical and Computer Engineering, Virginia Tech. He is currently a Post-Doctoral Researcher with the Electrical Engineering Department, Princeton University. His research interests include federated learning, reinforcement learning, virtual reality, unmanned aerial vehicles, and wireless networks. He was a recipient of the IEEE International Conference on Communications (ICC) 2020 Best Paper Award, the IEEE Global Communications Conference (GLOBECOM) 2020 Best Paper Award, and an exemplary reviewer for IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS in 2018 and IEEE TRANSACTIONS ON COMMUNICATIONS in 2018 and 2019. He has served as a co-chair for workshops on edge learning and wireless communications in several conferences including IEEE ICC, IEEE GLOBECOM, and IEEE Wireless Communications and Networking Conference. He currently serves as an Associate Editor of *Frontiers in Communications and Networks*, Specialty Section on Data Science for Communications and a Guest Editor of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS (JSAC) Special Issue on Distributed Learning over Wireless Edge Networks.

H. Vincent Poor (Life Fellow, IEEE) received the Ph.D. degree in EECS from Princeton University in 1977. From 1977 to 1990, he was on the faculty of the University of Illinois at Urbana-Champaign. Since 1990, he has been on the faculty at Princeton, where he is currently the Michael Henry Strater University Professor of Electrical Engineering. From 2006 to 2016, he served as the Dean of Princeton's School of Engineering and Applied Science. He has also held visiting appointments at several other universities, including most recently at Berkeley and Cambridge. His research interests are in the areas of information theory, machine learning and network science, and their applications in wireless networks, energy systems and related fields. Among his publications in these areas is the recent book *Multiple Access Techniques for 5G Wireless Networks and Beyond* (Springer, 2019).

Dr. Poor is a member of the National Academy of Engineering and the National Academy of Sciences, and is a foreign member of the Chinese Academy of Sciences, the Royal Society, and other national and international academies. Recent recognition of his work includes the 2017 IEEE Alexander Graham Bell Medal and a D.Eng. *honoris causa* from the University of Waterloo awarded in 2019.

Walid Saad (Fellow, IEEE) received the Ph.D. degree from the University of Oslo in 2010. He is currently a Professor with the Department of Electrical and Computer Engineering at Virginia Tech, where he leads the Network sciEnce, Wireless, and Security (NEWS) Laboratory. His research interests include wireless networks, machine learning, game theory, security, unmanned aerial vehicles, cyber-physical systems, and network science. He is an IEEE Distinguished Lecturer. He was also a recipient of the NSF CAREER award in 2013, the AFOSR summer faculty fellowship in 2014, and the Young Investigator Award from the Office of Naval Research (ONR) in 2015. He was the author/coauthor of nine conference best paper awards at WiOpt in 2009, ICIMP in 2010, IEEE WCNC in 2012, IEEE PIMRC in 2015, IEEE SmartGridComm in 2015, EuCNC in 2017, IEEE GLOBECOM in 2018, IFIP NTMS in 2019, and IEEE ICC in 2020. He was a recipient of the 2015 Fred W. Ellersick Prize from the IEEE Communications Society, of the 2017 IEEE ComSoc Best Young Professional in Academia award, of the 2018 IEEE ComSoc Radio Communications Committee Early Achievement Award, and of the 2019 IEEE ComSoc Communication Theory Technical Committee. He was also a coauthor of the 2019 IEEE Communications Society Young Author Best Paper. From 2015 to 2017, he was named the Stephen O. Lane Junior Faculty Fellow at Virginia Tech and, in 2017, he was named College of Engineering Faculty Fellow. He received the Dean's award for Research Excellence from Virginia Tech in 2019. He currently serves as an Editor of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE TRANSACTIONS ON MOBILE COMPUTING, and IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING. He is an Editor-at-Large of the IEEE TRANSACTIONS ON COMMUNICATIONS.

Shuguang Cui (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Stanford University, California, USA, in 2005. Afterwards, he has been working as Assistant, Associate, Full, Chair Professor in electrical and computer engineering with the University of Arizona, Texas A&M University, UC Davis, and CUHK at Shenzhen, respectively. He has also been the Executive Dean of the School of Science and Engineering, CUHK, Shenzhen, and the Executive Vice Director at the Shenzhen Research Institute of Big Data. His current research interests focus on data driven large-scale system control and resource management, large data set analysis, the IoT system design, energy harvesting based communication system design, and cognitive network optimization. He was selected as the Thomson Reuters Highly Cited Researcher and listed in the Worlds' Most Influential Scientific Minds by ScienceWatch in 2014. He was a recipient of the IEEE Signal Processing Society 2012 Best Paper Award. He has served as the general co-chair and TPC co-chairs for many IEEE conferences. He has also been serving as the Area Editor of the *IEEE Signal Processing Magazine*, and an Associate Editor of the IEEE TRANSACTIONS ON BIG DATA, IEEE TRANSACTIONS ON SIGNAL PROCESSING, IEEE JSAC SERIES ON GREEN COMMUNICATIONS AND NETWORKING, and IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS. He has been the elected member for IEEE Signal Processing Society SPCOM Technical Committee (2009–2014) and the elected Chair for IEEE ComSoc Wireless Technical Committee (2017–2018). He is a member of the Steering Committee of the IEEE TRANSACTIONS ON BIG DATA and the Chair of the Steering Committee of the IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING. He was also a member of the IEEE ComSoc Emerging Technology Committee. He was elected as an IEEE ComSoc Distinguished Lecturer in 2014, and IEEE VT Society Distinguished Lecturer in 2019. He has won the IEEE ICC best paper award, ICIP best paper finalist, the First Class Prize in Natural Science from Chinese Institute of Electronics, and the First Class Prize in Technology Invention from the China Institute of Communications all in 2020.