Efficient Federated Learning Algorithm for Resource Allocation in Wireless IoT Networks

Van-Dinh Nguyen, Shree Krishna Sharma, Thang X. Vu, Symeon Chatzinotas, and Björn Ottersten

***Abstract*—Federated learning (FL) allows multiple edge computing nodes to jointly build a shared learning model without having to transfer their raw data to a centralized server, thus reducing communication overhead. However, FL still faces a number of challenges such as non-iid distributed data and heterogeneity of user equipments (UEs). Enabling a large number of UEs to join the training process in every round raises a potential issue of the heavy global communication burden. To address these issues, we generalize the current state-of-the-art Federated Averaging (FedAvg) by adding a weight-based proximal term to the local loss function. The proposed FL algorithm runs stochastic gradient descent in parallel on a sampled subset of the total UEs with replacement during each global round. We provide a convergence upper bound characterizing the trade-off between convergence rate and global rounds, showing that a small number of active UEs per round still guarantees convergence. Next, we employ the proposed FL algorithm in wireless Internet-of-Things (IoT) networks to minimize either total energy consumption or completion time of FL, where a simple yet efficient path-following algorithm is developed for its solutions. Finally, numerical results on unbalanced datasets are provided to demonstrate the performance improvement and robustness on the convergence rate of the proposed FL algorithm over FedAvg. They also reveal that the proposed algorithm requires much less training time and energy consumption than the FL algorithm with full user participation. These observations advocate the proposed FL algorithm for a paradigm shift in bandwidth- constrained learning wireless IoT networks.**

***Index Terms*—Energy efficiency, federated learning, Internet-of-Things, inner approximation, resource allocation.**

I. Introduction

Nowadays, Internet-of-Things (IoT) and mobile devices are often equipped with advanced sensors and high computing capabilities that allow them to collect and process vast amounts of data generated at the network edge [1]–[4]. In addition, it is predicted that there will be over 10 billion smart objects in the IoT connected to the Internet and the overall mobile data will reach 49 exabytes per month by 2021 (an increase of about 188% compared to 2018) [5]. Computation and data storage services can be provided by a cloud and edge computing system [6], [7], in which users' tasks are intelligently uploaded to a cloud data center layer and an edge computing layer. Many IoT applications require pre-processing and classifying data and then are used to predict future events using machine learning (ML) techniques. The extensive amount of data of IoT devices is usually collected in private environments, and thus it is privacy-sensitive in nature. It is therefore generally not practical to send all data to a centralized server/cloud-center that trains a deep learning model. Besides, transferring a huge amount of data through wireless connectivity encounters expensive communication costs and high communication delays due to the limited resources of wireless systems.

To address the above challenges, it is necessary to devise a new ML technique through which each user equipment (UE) can be trained locally based on its collected data and by collaboratively building a shared global learning model. One of the most promising decentralized learning approaches to accomplish this goal is federated learning (FL) [8], [9]. FL allows multiple UEs to jointly train a global ML model without exchanging raw data between them or transferring their data to a centralized server. In particular, the server first broadcasts the latest global model to all participating UEs. Next, UEs compute local updates based on their available data and then send their local models back to the server. These steps are repeated until a certain level of global model accuracy is achieved. This way, only local model parameters are exchanged, and thus reducing the communication overhead. Communication-efficiency and incentive mechanism for FL have been investigated recently [10]–[14]. Nevertheless, there are still a number of challenges in implementing FL such as non-independent and identically distributed (non-iid) data across the network and high communication costs due to sending massive local model updates, which will be tackled in this paper.

A. Review of Related Literature

In this subsection, we review the state-of-the-art of FL techniques and FL performance optimization over wireless networks. Federated Averaging (FedAvg), which is a synchronous distributed optimization algorithm, is perhaps the most well-known FL algorithm [8]. FedAvg runs several updates of stochastic gradient descent (SGD) in parallel on UEs before averaging local model updates at a centralized server. Unlike GD and SGD, FedAvg executes more local updates and fewer global updates that improves communication efficiency. It was shown experimentally that FedAvg works well on non-iid data. However, its theoretical convergence guarantee in realistic settings was only recently provided in [15].

Fallah *et al.* proposed personalized federated learning (PerFedAvg) [16] to build a proper initialization model, inspired by Model-Agnostic Meta-Learning (MAML), that can be updated quickly to their own data after the training phase. Gradient-descent based FL for heterogeneous networks was studied in [17], where an adaptive control algorithm is proposed to obtain the desirable trade-off between local updates and global

The authors are with the Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, L-1855 Luxembourg City, Luxembourg (e-mail: {dinh.nguyen, shree.sharma, thang.vu, symeon.chatzinotas, bjorn.ottersten}@uni.lu). This work was supported in part by the ERC project AGNOSTIC.

aggregation steps. The work in [18] proposed communication-efficient FedAvg using a distributed form of Adam optimization and compression techniques to reduce the number of communication rounds and uploading data. Xie *et al.* [19] proposed an asynchronous federated optimization algorithm, which has been shown to achieve near-linear convergence to a global optimum. The authors of [20] proposed FedProx, which is generalized from FedAvg by adding the same proximal term to all the local functions that helps improve convergence speed. However, it is challenging to choose an appropriate value of the penalty constant in the proximal term, especially on unbalanced and non-iid datasets.

In another direction, many researchers have recently focused on the performance optimization of FL at the wireless network edge. In particular, three different scheduling policies were proposed in [21] to speed up the convergence of FL algorithms, accounting for the effects of user scheduling and interference. In [22], the desirable trade-off between total UEs' energy consumption and FL training time is studied, where all UEs are required to transmit their local updates in a synchronous manner. The work in [23] aimed at minimizing the global FL loss function under the effect of wireless communications. However, the computation delay of the local FL training model was not taken into account. An energy-efficient strategy for bandwidth allocation and scheduling was introduced in [24], which is capable of reducing UEs' energy consumption. The authors in [25] studied a sparse and low-rank problem to support FL algorithms, where user selection and beamforming design are jointly optimized. The superposition property of multi-access channel is exploited in [26] to obtain a low latency for the FL training time. To deal with non-iid data, the work in [27] proposed an online energy-aware dynamic worker scheduling policy to scale down the communication cost. Joint local accuracy, transmit power and UEs' processing frequency was studied in [28] to minimize the training time, where cell-free massive multiple-input multiple-output network is designed to support arbitrary FL algorithms. This work was extended in [29], in which frequency domain multiple access (FDMA) is adopted to transmit local updates asynchronously. However, the performance optimization of these works is mainly based on existing FL algorithms and requires all UEs to join the training process in every global round, leading to a suboptimal solution due to limited resources (i.e., bandwidth and energy).

B. Motivation and Main Contributions

In this paper, we aim at addressing the fundamental question: *Is it possible to efficiently utilize the limited communication and computation resources at the edge nodes to improve the performance of heterogeneous IoT networks by utilizing FL, while still guaranteeing convergence?* To do so, we entail the following inherent issues which may limit the FL performance in wireless IoT networks:

- **Non-iid and unbalanced data:** In distributed IoT networks, the collected data is distributed unevenly across UEs, and thus the data size and its distribution will be highly different among them. Therefore, random and uniform sampling schemes for selecting user participation result in the unstable and divergent convergence of FL algorithms.
- **Heterogeneity of UEs:** Each UE typically has very different types of computation capacity, channel gains and battery, which may have negative impacts on the performance of the synchronous FL algorithms (e.g., high completion time).
- **Large number of UEs and limited bandwidth:** The IoT network may constitute of a large number of UEs collaboratively building a shared learning model. However, in the FL-supported IoT network, it is not necessary to force all UEs to participate in every communication round, i.e., each device can be activated in several rounds of the training process. In addition, once the number of UEs is larger than a certain threshold, it may not be possible to obtain high-reliability and low-latency communications to upload local models due to the limited bandwidth. In this case, partial participation in each round is a good option, instead of full user participation [22]–[29].

To the best of our knowledge, this is the first work proposed for communication-efficient FL that takes into account all these issues. Our main contributions are summarized as follows:

- We propose a new FL algorithm generalized from the FedAvg by adding a proper weight-based proximal term to each local loss function to tackle non-iid and unbalanced data and heterogeneity of UEs. The key advantage of weight-based proximal term is to ensure that the global loss decreases steadily and smoothly, since local and global updates should have the same direction during the training process. That is to say, every UE can generate useful local model parameters to a centralized server which stabilizes the FL algorithm. In addition, we develop an efficient sampling strategy with replacement for partial user participation. Our theoretical analysis on non-iid data indicates that the convergence of the proposed FL algorithm can be guaranteed using a learning rate decay, despite the negative effects of sampling method.
- We formulate a resource allocation problem using the proposed FL algorithm in wireless IoT networks which targets key performance metrics of total energy consumption and completion time of FL. The problem captures a joint design of the signal transmission and the computation in one global round under both synchronous and asynchronous communication modes, which is formulated as a nonconvex optimization problem. Assuming that only the distribution of channels is known, we first transform the considered problem into an equivalent nonconvex form with a more computationally tractable form, and then develop an efficient path-following algorithm for its solution based on the inner approximation (IA) framework [30].

Numerical results in realistic federated settings are provided to validate our theoretical analysis and demonstrate the stability and robustness of the proposed FL algorithm on non-iid and unbalanced datasets. They also show that the proposed scheme

TABLE I: Summary of Main Notations and Symbols

$\mathbf{w}_g$	Global model at round g
$\mathbf{w}_{g,\ell}^k$	Local model at UE k in global round g and local round ℓ
$\mathbf{w}^*$	True optimal model corresponding to the minimum of $F(\mathbf{w}^*)$
$\lambda_{g,\ell}$	Learning rate in global round g and local round ℓ
$\xi_{g,\ell}^k$	A data sample uniformly picked from $\mathcal{D}_k$
$\mathcal{D}_k$ & D_k	Local input data set and number of samples of UE k, respectively
g & ℓ	Global and local round indexes, respectively
G & L	Total number of global rounds and number of local updates between two consecutive global rounds, resp.
$\mathcal{K}_{\mathtt{tot}}$ & $\mathcal{K}_g$	Set of total UEs and a subset of selected UEs at the global round g, respectively
$F(\mathbf{w})$ & $F_k(\mathbf{w})$	Global loss function and local loss function of UE k, respectively
$\mathbf{h}_k$	Channel vector between BS and UE k
B	Total bandwidth of system
$\mathtt{S}$	Data size of the global/local training update $\mathbf{w}$
c_k	Number of processing cycles of UE k (cycles/sample)
ϵ	A small positive constant to ensure high reliability
$\mathtt{f}_k$	CPU frequency of UE k (cycles/s)
$b_k^{\mathtt{dl}}$ & $b_k^{\mathtt{ul}}$	Bandwidth coefficients allocated to UE k in DL and UL, respectively
$\rho_k^{\mathtt{dl}}$ & $\rho_k^{\mathtt{ul}}$	Transmit power coefficients of UE k in DL and UL, respectively
$\langle \mathbf{a}, \mathbf{b} \rangle$	Inner product of two vectors $\mathbf{a}$ and $\mathbf{b}$
$\|\cdot\|$ & $\|\cdot\|$	A vector's Euclidean norm and absolute value of a complex scalar, respectively
$(\cdot)^H$ & $(\cdot)^T$	Hermitian transpose and normal transpose, respectively
∇	The gradient of a function
$\mathbb{E}\{\cdot\}$	Expectation of a random variable

requires much less training time and energy than existing FL-based schemes such as full user participation and equal bandwidth allocation. They also reveal the effectiveness of asynchronous communication in utilizing the limited communication and computation resources as well as handling UEs' heterogeneity.

C. Paper Organization and Mathematical Notation

The remainder of this paper is organized as follows. Preliminaries and definitions are given Section II. The proposed FL algorithm and resource allocation scheme for wireless IoT networks are provided in Sections III and IV, respectively. Numerical results are given in Section V, while Section VI concludes the paper. In order to make the rest of the paper easy to follow, we summarize the notations and symbols in Table I.

II. Preliminaries and Definitions

A. Network Model

We consider an FL-supported wireless IoT network consisting of one base station (BS) and a set $\mathcal{K}_{\mathtt{tot}} \triangleq \{1, 2, \cdots, K_{\mathtt{tot}}\}$ of $K_{\mathtt{tot}} = |\mathcal{K}_{\mathtt{tot}}|$ UEs, as illustrated in Fig. 1. The BS and UEs jointly build a shared ML model for data analysis and inference. In a radio-map-assisted wireless network [31], each UE collects measurement data in the wireless environment that is used to train ML algorithms to predict the performance of wireless networks. Heterogeneous computing capabilities of UEs can be empowered by Qualcomm Hexagon Vector eXtensions on Snapdragon 835 [32]. Each UE $k \in \mathcal{K}_{\mathtt{tot}}$ has a local input data set $\mathcal{D}_k \triangleq \{\mathbf{x}_{k1}, \mathbf{x}_{k2}, \cdots, \mathbf{x}_{kD_k}\}$, where D_k denotes the number of samples and each element $\mathbf{x}_{ki} \in \mathbb{R}^d$ is an input sample vector with d features. The local input data set of UE k may be different from other UEs, i.e., $\mathcal{D}_k \cap \mathcal{D}_{k'} = \emptyset, \forall k \neq k'$. Throughout the paper, we consider non-iid distributed data across the network, which are independent but not identically distributed. The total data size of all users can be defined by $D = \sum_{k \in \mathcal{K}_{\mathtt{tot}}} D_k$. In a typical learning algorithm, let $y_i \in \mathbb{R}$ be the output for the sample $\mathbf{x}_i$.

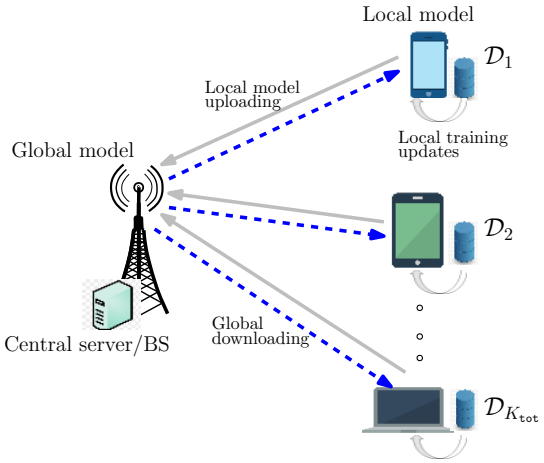


Fig. 1: Illustration of the FL-enabled wireless IoT network.

TABLE II: Loss Functions for Widely Used Machine Learning Models

Model	Loss function $f_i(\mathbf{w})$
Squared SVM	$\frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1}{2}\max\{0, 1 - y_i \mathbf{w}^T \mathbf{x}_i\|^2$, where λ is a const.
Linear regression	$\frac{1}{2}\|y_i - \mathbf{w}^T \mathbf{x}_i\|^2$
K-means	$\frac{1}{2}\min_{i' \in \{1,\cdots,\bar{K}\}} \|\mathbf{x}_i - \mathbf{w}_{i'}\|^2$, where $\bar{K}$ is the number of clusters
Neural network	$\frac{1}{2}\|y_i - \sum_{n \in \mathcal{N}} \omega_n \phi(\mathbf{w}_n^T \mathbf{w}_i)\|$, where ω_n, $\phi(\cdot)$ and $\mathcal{N}$ are the weight connecting neurons, the activation function and the set of neurons, respectively.

B. Loss Function

The main goal is to find the *model parameter* $\mathbf{w} \in \mathbb{R}^d$ that characterizes the output y_i with the loss function $f(\mathbf{w}, \mathbf{x}_i, y_i)$. For data sample $\mathbf{x}_i$, we rewrite $f(\mathbf{w}, \mathbf{x}_i, y_i)$ as $f_i(\mathbf{w})$ for simplicity. In Table II, we summarize various loss functions for widely used ML models [33]–[35]. The loss function on the data set $\mathcal{D}_k$ of UE k can be defined as

$$F_k(\mathbf{w}) \triangleq \frac{1}{D_k} \sum_{i \in \mathcal{D}_k} f_i(\mathbf{w}). \tag{1}$$

The aim of FL algorithms is to minimize the global loss value of the following distributed optimization problem:

$$\min_{\mathbf{w} \in \mathbb{R}^d} \quad F(\mathbf{w}) = \sum_{k \in \mathcal{K}_{\mathtt{tot}}} p_k F_k(\mathbf{w}), \tag{2}$$

where $p_k \triangleq D_k/D$ is the weighting factor for of UE k, satisfying $p_k \geq 0$ and $\sum_{k \in \mathcal{K}_{\mathtt{tot}}} p_k = 1$.

C. Review of Federated Averaging (FedAvg)

Definition 1. *The FL model generated by UEs using their local data set is called "local FL model" (or local model,*

for short), while that generated by BS is called "global FL model" (or global model, for short).

In a general FL framework, problem (2) is separately decomposed into K independent problems that are solved locally at UEs. Before we can consider ways to solve (2) efficiently, we first review FedAvg [8], which is also related to our proposed FL algorithm presented later in Section III. In IoT networks, each UE often has limited computation capability, and thus training on the entire (local) dataset using deterministic gradient descent (DGD) may not be realistic. In this paper, UE k solves its local problem by SGD with the same *step size* (a.k.a., the learning rate) and the number of *local updates* $L \geq 1$. At the g-th round of FedAvg, the centralized server co-located at BS transmits $\mathbf{w}_g$ to all UEs; each user (say, UE k) updates the latest global model $\mathbf{w}_{g,0}^k := \mathbf{w}_g$ and then runs SGD locally for L updates:

$$\mathbf{w}_{g,\ell+1}^k := \mathbf{w}_{g,\ell}^k - \lambda_{g,\ell} \nabla F_k(\mathbf{w}_{g,\ell}^k, \xi_{g,\ell}^k), \ell = 0, \cdots, L-1, \quad (3)$$

where $\lambda_{g,\ell}$ is the learning rate, and $\xi_{g,\ell}^k \in \mathcal{D}_k$ is a data sample uniformly picked from the local data set. Finally, the resulting local model updates are sent to the BS for averaging.

In the aggregation step of each round (or each iteration), there are two types of user participation such as full and partial user participation:

- **Full user participation:** All UEs send their local models back to BS for aggregation, and the aggregated global model to be used for the $g+1$-th round can be obtained as [22]–[24], [28], [29]:

$$\mathbf{w}_{g+1} := \sum_{k \in \mathcal{K}_{\texttt{tot}}} p_k \mathbf{w}_{g,L}^k. \quad (4)$$

  In this case, BS must wait for the slowest UEs (i.e., with low computing capability and low batter level), leading to serious straggler's effect and longer convergence time of FedAvg .
- **Partial user participation:** A subset $\mathcal{K}_g \triangleq \{1, 2, \cdots, K_g\}$ of K_g UEs with $K_g \leq K_{\texttt{tot}}$ are selected at round g to send local models [8], [15], [20], and BS then performs:

$$\mathbf{w}_{g+1} := \frac{1}{K_g} \sum_{k \in \mathcal{K}_g} \mathbf{w}_{g,L}^k. \quad (5)$$

  This scheme is more practically suited for large scale IoT networks in which a very large number of IoT devices (sensors, smartphones, and actuators, etc.) are deployed.

The training procedure of FedAvg with partial user participation can be summarized as follows:

1) BS transmits the latest global model to the selected UEs at random;
2) Each selected UE runs SGD locally for L iterations to output the trained local model (i.e., (3)), and sends it back to BS;
3) BS aggregates the received local models to update the global model (i.e., (5));
4) Steps 1-3 are repeated until convergence.

We note that the design of local updates at UEs and the aggregation strategy at BS may vary depending on the objective functions and different designs of FL algorithms [10], [21], [23], [36].

III. Federated Learning Algorithm Design

A. Proposed FL Algorithm Design

In IoT networks, there will be a massive number of deployed IoT users connected to collect data related to public safety, weather, energy and transportation, etc. The different UEs in IoT networks often have different resource constraints such as computation capabilities and power levels. In this paper, local models are updated via wireless links, and thus it may not be practical to collect all local models in each global aggregation step due to the limited bandwidth. Our FL algorithm design is similar to FedAvg [8] and FedProx [20] in the sense that a subset $\mathcal{K}_g$ of UEs are picked in each global round g to perform local training updates, and these will be sent back to BS to form a new global model.

However, to deal with two key challenges in the traditional federated optimization (i.e., systems heterogeneity and non-iid distributed data across the network) [20], we modify the local loss function of UE k at round g as:

$$f_i'(\mathbf{w}) \triangleq f_i(\mathbf{w}) + \frac{\mu p_k}{2} \|\mathbf{w} - \mathbf{w}_g\|^2, \forall i \in \mathcal{D}_k, \quad (6)$$

where $\mu > 0$ is a trade-off parameter. We note that $\frac{\mu p_k}{2}\|\mathbf{w} - \mathbf{w}_g\|^2$ can be viewed as a weight-based proximal term with a model parameter $\mathbf{w}$ and a parameter constant $\frac{\mu p_k}{2}$, which can be found in a wide range of applications such as l_2-regularized linear regression model $f_i'(\mathbf{w}) = \frac{1}{2}\|y_i - \mathbf{w}^T \mathbf{x}_i\|^2 + \frac{\mu}{2}\|\mathbf{w}\|^2$. We have the following remarks:

- The weight-based proximal term $\frac{\mu p_k}{2}\|\mathbf{w} - \mathbf{w}_g\|^2$ is incorporated to force the trained local model to be closer to the latest global model $\mathbf{w}_g$, thereby guaranteeing a smooth global loss value and improving the stability of FL algorithm. In other words, it ensures that new local models are not too diverse from the latest global model, so that UEs can generate useful local model parameters to the next global update, addressing the challenge of non-iid and unbalanced data as well as reducing negative impacts of random sampling strategy for selecting user participation.
- In some extreme cases, UE k can have a very large data (i.e., a high value of p_k), and the local model of this user is mainly determined by the latest global model. If $\mu_k \triangleq \mu p_k$ is set to be high for all UEs, which may drift the local models of UEs with low weights p too far away from the true model, leading to high divergence. On the other hand, a small value of μ_k may not make much difference. Therefore, it is necessary to select an appropriate value of μ for all UEs. The coefficient μ_k in (6) safely reflects the different amount of local data at UE k, which is more flexible than a fixed proximal term for all UEs [20].

Throughout this paper we consider $F_k(\mathbf{w})$ to be convex [15], [22], [23]. Thus, the local updates at UE k in (3) can be modified as:

$$\mathbf{w}_{g,\ell+1}^k := \mathbf{w}_{g,\ell}^k - \lambda_{g,\ell} \nabla F_k'(\mathbf{w}_{g,\ell}^k, \xi_{g,\ell}^k)$$

Algorithm 1 Federated Learning Algorithm Design

1: **Input:** $K_{\texttt{tot}}$, K_g, L, G, and $\mathcal{D}_k, \forall k, g$
2: Initialize the global model $\mathbf{w}_0$ and learning rate λ_0 to the same value for all UEs
3: **for** $g = 0, 1, \cdots, G-1$ **do**
4: BS picks a subset $\mathcal{K}_g$ of K_g UEs at random (UE k is selected with replacement according to the sampling probability p_k, $\forall k \in \mathcal{K}_{\texttt{tot}}$)
5: BS sends $\mathbf{w}_g$ to all UEs in $\mathcal{K}_g$
6: **for** $k \in \mathcal{K}_g$ *in parallel* **do**
7: $\mathbf{w}^k_{g,0} := \mathbf{w}_g$
8: **for** $\ell = 0, 1, \cdots, L-1$ **do**
9: Randomly pick a data point $\xi^k_{g,\ell} \in \mathcal{D}_k$
10: Update: $\mathbf{w}^k_{g,\ell+1} := \mathbf{w}^k_{g,\ell} - \lambda_{g,\ell}\big(\nabla F_k(\mathbf{w}^k_{g,\ell}, \xi^k_{g,\ell}) + \mu p_k(\mathbf{w}^k_{g,\ell} - \mathbf{w}_g)\big)$
11: **end for**
12: Send $\mathbf{w}^k_{g+1} := \mathbf{w}^k_{g,L}$ to BS
13: **end for**
14: BS aggregates to update the global model parameter as: $\mathbf{w}_{g+1} := \frac{1}{K_g}\sum_{k\in\mathcal{K}_g} \mathbf{w}^k_{g+1}$
15: **end for**

$$:= \mathbf{w}^k_{g,\ell} - \lambda_{g,\ell}\big(\nabla F_k(\mathbf{w}^k_{g,\ell}, \xi^k_{g,\ell}) + \mu p_k(\mathbf{w}^k_{g,\ell} - \mathbf{w}_g)\big). \quad (7)$$

The proposed FL algorithm design is summarized in Algorithm 1. Here, a slight but important modification in Step 10 of Algorithm 1 is expected to obtain significant performance improvement compared to FedAvg with strong theoretical convergence guaranteed, which will be detailed next. We note that the convergence of FedAvg on non-iid data has been studied in [15].

B. Convergence Analysis

In this subsection, we analyze the convergence of Algorithm 1 and provide an upper bound of $\mathbb{E}\{F(\mathbf{w}_G)\} - F(\mathbf{w}^*)$, where $\mathbf{w}^*$ denotes the optimal global model corresponding to the minimum of the global loss F.

1) Assumptions

To facilitate the analysis, we first make the following assumptions on the modified local loss functions, which are widely adopted in the literature [11], [15], [20], [22], [23].

Assumption 1. *$F'_k(\cdot)$ is Γ-smooth; i.e., $\forall \mathbf{w}, \tilde{\mathbf{w}} \in \mathbb{R}^d$*

$$F'_k(\mathbf{w}) \le F'_k(\tilde{\mathbf{w}}) + \langle \nabla F'_k(\tilde{\mathbf{w}}), \mathbf{w} - \tilde{\mathbf{w}} \rangle + \frac{\Gamma}{2}\|\mathbf{w} - \tilde{\mathbf{w}}\|^2, \ \forall k. \quad (8)$$

Assumption 2. *If $F_k(\cdot)$ is convex, $F'_k(\cdot)$ becomes μ_k-strongly convex (since the Hessian of $F'_k(\cdot)$ can be positive semi-definite); i.e., $\forall \mathbf{w}, \tilde{\mathbf{w}} \in \mathbb{R}^d$ and there exists $\mu_k > 0$ such that*

$$F'_k(\mathbf{w}) \ge F'_k(\tilde{\mathbf{w}}) + \langle \nabla F'_k(\tilde{\mathbf{w}}), \mathbf{w} - \tilde{\mathbf{w}} \rangle + \frac{\mu_k}{2}\|\mathbf{w} - \tilde{\mathbf{w}}\|^2, \ \forall k. \quad (9)$$

We note that $\sum_{k\in\mathcal{K}_{\texttt{tot}}} \mu_k = \mu \sum_{k\in\mathcal{K}_{\texttt{tot}}} p_k = \mu$.

Assumption 3. *For any k and $\mathbf{w}$, we define δ as an upper bound of the expected squared norm of stochastic gradients $\mathbb{E}\{\|\nabla F'_k(\mathbf{w}^k_{g,\ell}, \xi^k_{g,\ell})\|^2\}$, i.e.,*

$$\mathbb{E}\{\|\nabla F'_k(\mathbf{w}^k_{g,\ell}, \xi^k_{g,\ell})\|^2\} \le \delta. \quad (10)$$

2) Main Results

For simplicity, we assume that $\lambda_g = \lambda_{g,\ell}, \forall \ell$, i.e., all local iterations use the same learning rate in each aggregation step. Let us define that

$$\bar{\mathbf{w}}_{g,\ell+1} \triangleq \sum_{k\in\mathcal{K}_{\texttt{tot}}} p_k \Big(\mathbf{w}^k_{g,\ell} - \lambda_g \nabla F'_k(\mathbf{w}^k_{g,\ell}, \xi^k_{g,\ell})\Big), \quad (11)$$

which is the aggregation of one local update from all UEs. For $\Delta F'(\bar{\mathbf{w}}_{g,\ell}) \triangleq \sum_{k\in\mathcal{K}_{\texttt{tot}}} p_k \nabla F'_k(\mathbf{w}^k_{g,\ell}, \xi^k_{g,\ell})$, it follows that

$$\begin{aligned}\bar{\mathbf{w}}_{g,\ell+1} &= \sum_{k\in\mathcal{K}_{\texttt{tot}}} p_k \mathbf{w}^k_{g,\ell} - \lambda_g \nabla F'(\bar{\mathbf{w}}_{g,\ell}) \\ &= \bar{\mathbf{w}}_{g,\ell} - \lambda_g \nabla F'(\bar{\mathbf{w}}_{g,\ell}). \end{aligned} \quad (12)$$

In the case of full user participation in (4), we always have $\bar{\mathbf{w}}_{g,\ell+1} = \sum_{k\in\mathcal{K}_{\texttt{tot}}} p_k \mathbf{w}^k_{g,\ell+1}$, but not hold true for our proposed FL due to the randomness of sampling strategy. We recall that in the global round g, only a subset $\mathcal{K}_g$ of UEs are randomly selected to join the training process. Thus, the sampling strategy must be indifferent among all UEs with respect to their weights which guarantees the convergence of the FL algorithm. Inspired by [15], we define the following relationship to capture the *unbiased sampling strategy which is related how UEs are picked at different rounds.*

Definition 2. *The expectation of the next global model parameter of selected UEs is equal to the average of all UEs' updated local parameters, i.e.,*

$$\mathbb{E}\Big\{\frac{1}{K_g}\sum_{k\in\mathcal{K}_g} \mathbf{w}^k_{g+1}\Big\} = \bar{\mathbf{w}}_{g+1}. \quad (13)$$

For $\mathbf{w}_{g+1} \triangleq \frac{1}{K_g}\sum_{k\in\mathcal{K}_g} \mathbf{w}^k_{g+1}$, the expected *divergence* between $\mathbf{w}_{g+1}$ and $\bar{\mathbf{w}}_{g+1}$ is characterized by the following lemma.

Lemma 1. *Assuming $\{\lambda_g\}_{\forall g}$ is a non-increasing sequence (i.e., $\lambda_g \le \lambda_{g-1}$) with a learning rate decay as $\lambda_g \le \frac{\lambda_0}{1+ag}$ for any positive constant $a > 0$, the expected upper bound of $\|\mathbf{w}_{g+1} - \bar{\mathbf{w}}_{g+1}\|^2$ can be given as*

$$\mathbb{E}\big\{\|\mathbf{w}_{g+1} - \bar{\mathbf{w}}_{g+1}\|^2 \big| k \in \mathcal{K}_g\big\} \le \frac{L^2\lambda_0^2\delta}{K_g(1+ag)^2}. \quad (14)$$

Proof: We first use Definition 2 to overcome the difficulty of the expectation over the subset $\mathcal{K}_g$, based on which the upper bound in (14) is obtained. For details, please see Appendix A. ■

Theorem 1. *Let all Assumptions 1-3 hold. Given the optimal global model $\mathbf{w}^*$, the learning rate $\lambda_g \le \frac{\lambda_0}{1+ag}$ with $\lambda_0 \le \frac{2}{\mu+\Gamma}$ and $\varepsilon_0 = \|\mathbf{w}_0 - \mathbf{w}^*\|^2$, the expected convergence upper bound of $F(\mathbf{w}_G) - F(\mathbf{w}^*)$ after G global rounds can be given by*

$$\begin{aligned}&\mathbb{E}\{F(\mathbf{w}_G)\} - F(\mathbf{w}^*) \\ &\le \frac{\Gamma}{2}\Big(\frac{L^2\lambda_0^2\delta}{K_G(1+aG)^2} + \prod_{i=0}^{G-1}\big(1 - \frac{2\lambda_0\mu\Gamma}{(\mu+\Gamma)(1+ia)}\big)\varepsilon_0\Big). \end{aligned} \quad (15)$$

Proof: We first analyze the expected upper bound of

$\|\mathbf{w}_{g+1}-\mathbf{w}^*\|^2$ within the subset $\mathcal{K}_g$. Combining with the result in Lemma 1, we obtain the final result. For details, please see Appendix B. ■

Corollary 1. *Let all Assumptions 1-3 hold. Given the optimal global model* $\mathbf{w}^*$ *and* $\varepsilon_0 = \|\mathbf{w}_0-\mathbf{w}^*\|^2$. *If we fix the learning rate to be* $\lambda_g = \frac{2}{\mu+\Gamma}, \forall g$, *the expected convergence upper bound of* $F(\mathbf{w}_G) - F(\mathbf{w}^*)$ *in Theorem 1 becomes*

$$\mathbb{E}\{F(\mathbf{w}_G)\} - F(\mathbf{w}^*) \leq \frac{\Gamma}{2}\left(\frac{4L^2\delta}{K_G(\mu+\Gamma)^2} + \left(1-\frac{4\mu\Gamma}{(\mu+\Gamma)^2}\right)^G \varepsilon_0\right). \quad (16)$$

It can be seen that $\lim_{G\to\infty}\left(1-\frac{4\mu\Gamma}{(\mu+\Gamma)^2}\right)^G = 0$ due to $0 \leq 1-\frac{4\mu\Gamma}{(\mu+\Gamma)^2} < 1$.

Remark 1. *From Theorem 1, we can observe that the proposed Algorithm 1 will converge to the optimum (global model) after a sufficiently large number of global rounds. In addition, with a fixed learning rate in Corollary 1, it will converge to a suboptimal solution with a gap of the total loss of* $\frac{4L^2\delta\Gamma}{2K_g(\mu+\Gamma)^2}$ *away from the optimum due to the heterogeneity of the data distribution. These observations simply imply that an adaptive learning rate is necessary for the proposed FL algorithm to obtain the optimal global model. However, we note that decreasing the learning rate very often (e.g., after each local update) may slow down the convergence speed of the proposed FL algorithm. The key advantage of* `FedAvg` *is mainly attributed to the fact that it performs several local model updates before communicating with the centralized server for global aggregation, which achieves a low communication cost.*

Remark 2. *The sampling strategy is controlled by the central server co-located at BS. Compared to FL algorithms with full user participation and uniform sampling, Algorithm 1 may additionally require all UEs to update their number of samples to the central server before performing the sampling strategy. This step is done once in the entire implementation of Algorithm 1. For real-time video streaming applications, the number of training samples needs to be updated regularly that allows the system to adapt quickly to the newly collected data. The fine-tuned learning rate* λ_g, *trade-off parameter* μ, *and number of local iterations* L *are necessary to prevent divergence of Algorithm 1.*

Practical Implementation: In addition to the learning rate, we now discuss the selection of other parameters to successfully implement Algorithm 1.

1) *Selection of L:* In principle, the number of local updates L is controlled by UEs to achieve an approximate solution that satisfies certain local accuracy. If L is too large, Algorithm 1 becomes the one-short averaging [12], where $\mathbf{w}_{g,\ell}^k$ will converge only to an optimal solution of the local loss function $F_k(\cdot)$. On the other hand, Algorithm 1 with very small L will result in a heavy global communication burden. Therefore, an appropriate value of L is necessary not only to guarantee the convergence of Algorithm 1 but also to reduce communication costs.

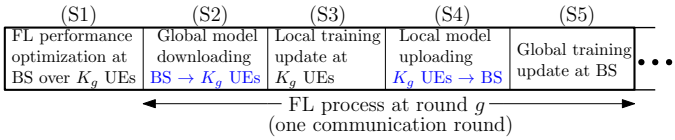


Fig. 2: Illustration of the proposed scheme to support FL over wireless IoT networks.

2) *Selection of K_g:* From Theorem 1, it can be seen that the convergence rate is less dependent on K_g than G and L. Thus, one can set the number of user participants to be small that still ensure high quality of training parameters transmitted over wireless links (due to limited resources) without compromising the convergence rate, suitable for IoT networks having a large number of UEs.

IV. Proposed FL-enabled Resource Optimization over Wireless IoT Networks

Given insights from Section III, we are now in a position to develop a resource allocation algorithm to optimize the FL performance over wireless IoT networks.

In what follows, *one FL process* is referred to as the entire implementation of Algorithm 1 until convergence. The proposed scheme to support FL is illustrated in Fig. 2, consisting of five steps. Compared to Algorithm 1, an additional step (S1) is added to enhance the performance of FL (e.g., in terms of energy consumption and training time). This step is done at the BS side before executing Algorithm 1 in each communication round.

Remark 3. *In practice, the BS is often equipped with much higher computation capacity than UEs for executing tasks. In addition, the BS is located in a place closer to UEs, and thus, the latency of performing Steps (S1) and (S5) can be neglected. This assumption is also adopted in recent works on the FL for wireless communications [22]–[24], [28], [29].*

A. System Model

We assume that BS is equipped with N antennas to serve single-antenna UEs via a shared wireless medium. We consider the channel reciprocity between BS and UEs in time division duplex mode. The channel vector between BS and UE k is denoted by $\mathbf{h}_k \in \mathbb{C}^{N\times 1}$, which is generally modeled as $\mathbf{h}_k = \sqrt{\varphi_k}\bar{\mathbf{h}}_k$; Here, φ_k accounts for the effect of large-scale fading (e.g., path loss and shadowing), and $\bar{\mathbf{h}}_k \sim \mathcal{CN}(0, \mathbf{I}_N)$ denotes the small-scale fading.

1) Communication Model: BS transmits the latest global model to the selected UEs using frequency-division multiple access (FDMA) over the total bandwidth B (i.e., step S2 in Fig. 2). We denote the bandwidth allocated to UE k via downlink by $b_k^{\mathtt{dl}}B$, satisfying $\sum_{k\in\mathcal{K}_g} b_k^{\mathtt{dl}} \leq 1$. Applying maximum ratio transmission (MRT) beamforming (i.e., $\mathbf{h}_k/\|\mathbf{h}_k\|$), UE k downloads the FL global model from BS with the transmission rate (bits/s):

$$r_k^{\mathtt{dl}} = b_k^{\mathtt{dl}} B \log\left(1+\mathsf{SNR}_k^{\mathtt{dl}}\right), \quad (17)$$

where $\mathsf{SNR}_k^{\mathtt{dl}} \triangleq \frac{\|\rho_k^{\mathtt{dl}}\mathbf{h}_k\|^2}{b_k^{\mathtt{dl}}BN_0}$ and N_0 are the downlink signal-to-noise ratio (SNR) and noise power spectral density at UE k, respectively; $\rho_k^{\mathtt{dl}}$ is the transmit power coefficient allocated to UE k subject to $\sum_{k\in\mathcal{K}_g}(\rho_k^{\mathtt{dl}})^2 \le P_{\mathtt{BS}}$, where $P_{\mathtt{BS}}$ is the power budget at BS. Let $\mathtt{S}$ be the data size (in bits) that BS and UEs require to transmit the global/local training update $\mathbf{w}$ over wireless links.[1] Thus, the downlink communication delay for UE k to download the global model is

$$t_{co,k}^{\mathtt{dl}} = \frac{\mathtt{S}}{r_k^{\mathtt{dl}}}. \tag{18}$$

UEs perform the local training based on the latest global model, and then transmit the trained local models to BS using FDMA (i.e., step S4). Applying maximum ratio combining (MRC) receiver at BS (i.e., $\mathbf{h}_k^H/\|\mathbf{h}_k\|$), the data rate of UE k to transmit its local FL model to BS can be given as

$$r_k^{\mathtt{ul}} = b_k^{\mathtt{ul}} B\log\big(1+\mathsf{SNR}_k^{\mathtt{ul}}\big), \tag{19}$$

where $\mathsf{SNR}_k^{\mathtt{ul}} \triangleq \frac{\|\rho_k^{\mathtt{ul}}\mathbf{h}_k\|^2}{b_k^{\mathtt{ul}}BN_0}$ is the uplink SNR of UE k; $b_k^{\mathtt{ul}}$ and $\rho_k^{\mathtt{ul}}$ are the bandwidth and transmit power coefficients of UE k in uplink, respectively. The uplink communication delay to upload the local FL model is thus

$$t_{co,k}^{\mathtt{ul}} = \frac{\mathtt{S}}{r_k^{\mathtt{ul}}}. \tag{20}$$

2) Computation and Energy Consumption Model at UEs: Let c_k (cycles/sample) be the number of processing cycles of UE k to execute one sample of data, which assumes to be measured offline and known *a prior* [37]. Denoting the central processing unit (CPU) frequency of UE k by f_k (cycles/s), the computation time for the local training update at UE k ((i.e., step S3) over L local iterations is given by

$$t_{cp,k} = L\frac{c_k D_k}{\mathsf{f}_k}, \tag{21}$$

where $\frac{c_k D_k}{\mathsf{f}_k}$ can be interpreted as the computation time per one local iteration. The energy consumption at UE k in one global round can be formulated as:

$$E_k = E_{co,k} + E_{cp,k}, \tag{22}$$

where $E_{co,k} = (\rho_k^{\mathtt{ul}})^2 t_{co,k}^{\mathtt{ul}}$ is the energy consumption required to transmit the local update via the uplink, and $E_{cp,k} = L\frac{\theta_k}{2}c_k D_k \mathsf{f}_k^2$ denotes the CPU energy consumption for L local iterations [38] with $\theta_k/2$ being the effective capacitance coefficient depending on the chipset of UE k. Next, the energy consumption of K_g UEs at round g is expressed as follows:

$$E_g = \sum_{k\in\mathcal{K}_g} E_k = \sum_{k\in\mathcal{K}_g}(E_{co,k}+E_{cp,k}). \tag{23}$$

B. Problem Formulation

To convey local models to BS in the uplink, we consider two different schemes, i.e., synchronous (Syn) and asynchronous (Asyn) communication. The former requires all selected UEs to complete the current step before starting the next step [22], [28], while the latter allows each selected UE to communicate with BS in an asynchronous manner [29].

The total time of one communication round of the FL process at iteration g is

$$T_g^{\mathrm{X}} = \begin{cases} \max\limits_{k\in\mathcal{K}_g}\{t_{co,k}^{\mathtt{dl}}\} + \max\limits_{k\in\mathcal{K}_g}\{t_{cp,k}\} + \max\limits_{k\in\mathcal{K}_g}\{t_{co,k}^{\mathtt{ul}}\}, & \\ \qquad\qquad\qquad\qquad\qquad \text{if X is Syn}, & \text{(24a)} \\ \max\limits_{k\in\mathcal{K}_g}\{t_{co,k}^{\mathtt{dl}} + t_{cp,k} + t_{co,k}^{\mathtt{ul}}\}, \ \text{if X is Asyn}. & \text{(24b)} \end{cases}$$

We introduce a constant parameter $\eta\in\{0,1\}$ to formulate the utility function as $\eta E_g + (1-\eta)T_g^{\mathrm{X}}$, where $\eta = 1$ ($\eta = 0$, respectively) corresponds to the energy consumption minimization problem (the time minimization, respectively). At the global round g, we seek the solution to the following minimization problem:

$$\begin{aligned}
\min_{\boldsymbol{\rho},\mathbf{f},\mathbf{b}} \quad & \eta E_g + (1-\eta)T_g^{\mathrm{X}} && \text{(25a)}\\
\text{s.t.} \quad & E_k \le \mathcal{E}_{\max},\ \forall k\in\mathcal{K}_g, && \text{(25b)}\\
& T_k \le \mathcal{T}_{\max},\ \forall k\in\mathcal{K}_g, && \text{(25c)}\\
& \mathsf{SNR}_k^{x} \ge \gamma^{\min},\ \forall k\in\mathcal{K}_g, x\in\{\mathtt{dl},\mathtt{ul}\}, && \text{(25d)}\\
& (\rho_k^{\mathtt{ul}})^2 \le P_k^{\max},\ \forall k\in\mathcal{K}_g, && \text{(25e)}\\
& \sum_{k\in\mathcal{K}_g}(\rho_k^{\mathtt{dl}})^2 \le P_{\mathtt{BS}}, && \text{(25f)}\\
& \mathsf{f}_k^{\min} \le \mathsf{f}_k \le \mathsf{f}_k^{\max},\ \forall k\in\mathcal{K}_g, && \text{(25g)}\\
& \sum_{k\in\mathcal{K}_g} b_k^{x} \le 1, \forall x\in\{\mathtt{dl},\mathtt{ul}\}, && \text{(25h)}
\end{aligned}$$

where $T_k = t_{co,k}^{\mathtt{dl}} + t_{cp,k} + t_{co,k}^{\mathtt{ul}}$, $\boldsymbol{\rho} \triangleq \{\boldsymbol{\rho}^{\mathtt{dl}},\boldsymbol{\rho}^{\mathtt{ul}}\}$, $\boldsymbol{\rho}^{\mathtt{dl}} \triangleq \{\rho_k^{\mathtt{dl}}\}_{k\in\mathcal{K}_g}$, $\boldsymbol{\rho}^{\mathtt{ul}} \triangleq \{\rho_k^{\mathtt{ul}}\}_{k\in\mathcal{K}_g}$, $\mathbf{f} \triangleq \{\mathsf{f}_k\}_{k\in\mathcal{K}_g}$ and $\mathbf{b} \triangleq \{b_k^{\mathtt{dl}}, b_k^{\mathtt{ul}}\}_{k\in\mathcal{K}_g}$. Constraints (25b) and (25c) indicate the maximum energy consumption $\mathcal{E}_{\max}$ and the delay requirement $\mathcal{T}_{\max}$, respectively, for executing one communication round. Constraint (25d) with a minimum SNR requirement $\gamma^{\min}$ is added to ensure that BS can successfully decode the message. The parameters $(\mathcal{E}_{\max},\ \mathcal{T}_{\max},\gamma^{\min})$ assume to be known *a prior* at the BS. When the condition in (25c) is not met for some UEs, BS simply ignores these stragglers from the global training update. (25e), (25g) and (25h) are the CPU-frequency, transmit power and bandwidth constraints for each UE, respectively, which also capture UEs' heterogeneity with different types of computation capability and battery level.

Definition 3. *The effective completion time and the total energy consumption of all UEs for implementing the FL algorithm are computed as $T^X = \sum_{g=1}^{G}\mathbb{E}\{T_g^X\}$ and $E = \sum_{g=1}^{G}\mathbb{E}\{E_g\}$, respectively. Here, $\mathbb{E}\{T_g^X\}$ and $\mathbb{E}\{E_g\}$ are the average of T_g^X and E_g over the random sampling of devices and large-scale fading realizations, respectively.*

C. Proposed Path-Following Algorithm

We note that the functions $E_{co,k}$, $t_{co,k}^{\mathtt{dl}}$, $t_{co,k}^{\mathtt{ul}}$, $\mathsf{SNR}_k^{\mathtt{dl}}$ and $\mathsf{SNR}_k^{\mathtt{ul}}$, $\forall k$ are neither convex nor concave in $(\boldsymbol{\rho},\mathbf{b})$, which can be verified by checking the Hessian matrix. As a consequence, the objective (25a) and constraints (25b)-(25d) are nonconvex, causing problem (25) to be nonconvex. In what follows, we first transform problem (25) to a computationally

[1]The transmit data size can be calculated as $\mathtt{S} = 32d$ bits, since it typically takes $32d$ bits to encode the vector $\mathbf{w}$ of length d.

tractable form and then apply IA method [30] to develop an efficient path-following algorithm for its solution.

Let us start by rewriting (25) equivalently as

$$\min_{\boldsymbol{\rho},\mathbf{f},\mathbf{b},\boldsymbol{\vartheta},\mathbf{t}} \quad \eta E_g(\boldsymbol{\rho}^{\mathtt{ul}},\mathbf{f},\boldsymbol{\vartheta}^{\mathtt{ul}})+(1-\eta)T_g^{\mathrm{X}}(\mathbf{t}) \tag{26a}$$

$$\text{s.t.} \quad r_k^x \geq \vartheta_k^x,\ \forall k\in\mathcal{K}_g, x\in\{\mathtt{dl},\mathtt{ul}\}, \tag{26b}$$

$$E_k(\rho_k^{\mathtt{ul}},\mathsf{f}_k,\vartheta_k^{\mathtt{ul}})\leq \mathcal{E}_{\max},\ \forall k\in\mathcal{K}_g, \tag{26c}$$

$$T_k(\mathsf{f}_k,\vartheta_k^{\mathtt{ul}},\vartheta_k^{\mathtt{dl}})\leq \mathcal{T}_{\max},\ \forall k\in\mathcal{K}_g, \tag{26d}$$

$$\begin{cases} L\frac{c_kD_k}{\mathsf{f}_k}\leq t_{cp},\ \forall k\in\mathcal{K}_g, \text{ if X is Syn},\\ \frac{\mathsf{S}}{\vartheta_k^x}\leq t_{co}^x, \forall k\in\mathcal{K}_g, x\in\{\mathtt{dl},\mathtt{ul}\} \text{ if X is Syn},\\ T_k(\mathsf{f}_k,\vartheta_k^{\mathtt{ul}},\vartheta_k^{\mathtt{dl}})\leq t,\ \forall k\in\mathcal{K}_g, \text{ if X is Asyn}, \end{cases} \tag{26e}$$

$$(25\text{d}),(25\text{e}),(25\text{f}),(25\text{g}),(25\text{h}), \tag{26f}$$

where

$$\begin{aligned} E_g(\boldsymbol{\rho}^{\mathtt{ul}},\mathbf{f},\boldsymbol{\vartheta}^{\mathtt{ul}})&=\sum_{k\in\mathcal{K}_g}E_k(\rho_k^{\mathtt{ul}},\mathsf{f}_k,\vartheta_k^{\mathtt{ul}}),\\ E_k(\rho_k^{\mathtt{ul}},\mathsf{f}_k,\vartheta_k^{\mathtt{ul}})&=\mathsf{S}\frac{(\rho_k^{\mathtt{ul}})^2}{\vartheta_k^{\mathtt{ul}}}+L\frac{\theta_k}{2}c_kD_k\mathsf{f}_k^2,\\ T_k(\mathsf{f}_k,\vartheta_k^{\mathtt{ul}},\vartheta_k^{\mathtt{dl}})&=\frac{\mathsf{S}}{\vartheta_k^{\mathtt{dl}}}+L\frac{c_kD_k}{\mathsf{f}_k}+\frac{\mathsf{S}}{\vartheta_k^{\mathtt{ul}}}, \end{aligned} \tag{27}$$

$$T_g^{\mathrm{X}}(\mathbf{t})=\begin{cases} t_{co}^{\mathtt{dl}}+t_{cp}+t_{co}^{\mathtt{ul}}, & \text{if X is Syn},\\ t, & \text{if X is Asyn}, \end{cases}$$

and $\mathbf{t}\triangleq\{t_{cp},t_{co}^{\mathtt{ul}},t_{co}^{\mathtt{dl}},t\}$ and $\boldsymbol{\vartheta}\triangleq\{\boldsymbol{\vartheta}^{\mathtt{ul}},\boldsymbol{\vartheta}^{\mathtt{dl}}\}$ with $\boldsymbol{\vartheta}^{\mathtt{ul}}\triangleq\{\vartheta_k^{\mathtt{ul}}\}_{k\in\mathcal{K}_g}$ and $\boldsymbol{\vartheta}^{\mathtt{dl}}\triangleq\{\vartheta_k^{\mathtt{dl}}\}_{k\in\mathcal{K}_g}$ are newly introduced variables to unravel the nonsmooth objective function. It is observed that the objective and all the constraints are convex and linear, except (26b).

Due to limited resources and low computation capacity of UEs, it may take more than one communication block to complete one global round. Thus, the small-scale fading may change during local training updates, and the perfect instantaneous CSI of UEs may be difficult to obtain in practice. In addition, one large-scale coherence time can be invariant at least 40 small-scale fading coherence intervals [39]. Assuming that the large-scale fading is a known deterministic quantity, we replace (26b) by the following outage constraint [40]:

$$\begin{aligned} (26\text{b})\Rightarrow \Big\{ b_k^xB\log\big(1+\gamma_k^x\big)\geq\vartheta_k^x\Big| \\ \mathsf{Prob}\Big(\frac{\varphi_k\|\rho_k^x\bar{\mathbf{h}}_k\|^2}{b_k^xBN_0}<\gamma_k^x\Big)\leq\epsilon\Big\}, \forall k\in\mathcal{K}_g, x\in\{\mathtt{dl},\mathtt{ul}\}, \end{aligned} \tag{29}$$

which ensures a sufficient margin. Here, γ_k^x is a new variable representing the soft SNR of UE k, and ϵ is a small positive constant (close to 0) to ensure high reliability. (29) is a non-linear probabilistic constraint, which may not be solved directly. Instead, we introduce the following lemma to evaluate (29).

Lemma 2. *Constraint* (26b) *is equivalent to the following constraint:*

$$(26\text{b})\Leftrightarrow b_k^xB\log\big(1+\gamma_k^x\big)\geq\vartheta_k^x, \forall k\in\mathcal{K}_g, x\in\{\mathtt{dl},\mathtt{ul}\}, \tag{30}$$

under the condition:

$$\frac{\varphi_kN}{BN_0}\ln(1-\epsilon)+\frac{\gamma_k^xb_k^x}{(\rho_k^x)^2}=0,\ \forall k\in\mathcal{K}_g, x\in\{\mathtt{dl},\mathtt{ul}\}. \tag{31}$$

Proof: Based on the foundation results in [40], a self-contained proof is given in Appendix C. ■

From Lemma 2, we rewrite (26) as

$$\min_{\boldsymbol{\rho},\mathbf{f},\mathbf{b},\boldsymbol{\vartheta},\mathbf{t},\boldsymbol{\gamma}} \quad \eta E_g(\boldsymbol{\rho}^{\mathtt{ul}},\mathbf{f},\boldsymbol{\vartheta}^{\mathtt{ul}})+(1-\eta)T_g^{\mathrm{X}}(\mathbf{t}) \tag{32a}$$

$$\text{s.t.} \quad b_k^xB\log\big(1+\gamma_k^x\big)\geq\vartheta_k^x, k\in\mathcal{K}_g, x\in\{\mathtt{dl},\mathtt{ul}\}, \tag{32b}$$

$$\frac{\varphi_kN}{BN_0}\ln(1-\epsilon)+\frac{\gamma_k^xb_k^x}{(\rho_k^x)^2}\leq 0,\quad \forall k\in\mathcal{K}_g, x\in\{\mathtt{dl},\mathtt{ul}\}, \tag{32c}$$

$$\gamma_k^x\geq\gamma^{\min},\ \forall k\in\mathcal{K}_g, x\in\{\mathtt{dl},\mathtt{ul}\} \tag{32d}$$

$$(25\text{e}),(25\text{f}),(25\text{g}),(25\text{h}),(26\text{c}),(26\text{d}),(26\text{e}), \tag{32e}$$

where (32c) must hold with equality at the optimum, and $\boldsymbol{\gamma}\triangleq\{\gamma_k^{\mathtt{dl}},\gamma_k^{\mathtt{ul}}\}_{k\in\mathcal{K}_g}$. The linear constraint (32d) is derived from (25d).

We are now ready to approximate the nonconvex constraints (32b) and (32c). By IA method [30], (32b) is iteratively replaced by the following convex constraint at iteration κ:

$$\Phi^{(\kappa)}(\gamma_k^x,b_k^x)\triangleq\mathsf{A}^{(\kappa)}-\mathsf{B}^{(\kappa)}\frac{1}{\gamma_k^x}-\mathsf{C}^{(\kappa)}\frac{1}{b_k^x}\geq\frac{\vartheta_k^x}{B},\forall k,x, \tag{33}$$

whose proof is given in Appendix D, where $\mathsf{A}^{(\kappa)}\triangleq 2b_k^{x,(\kappa)}\log\big(1+\gamma_k^{x,(\kappa)}\big)+\frac{b_k^{x,(\kappa)}\gamma_k^{x,(\kappa)}\log(e)}{(\gamma_k^{x,(\kappa)}+1)}$, $\mathsf{B}^{(\kappa)}\triangleq\frac{b_k^{x,(\kappa)}(\gamma^{x,(\kappa)})^2\log(e)}{(\gamma^{x,(\kappa)}+1)}$ and $\mathsf{C}^{(\kappa)}\triangleq(b_k^{x,(\kappa)})^2\log\big(1+\gamma_k^{x,(\kappa)}\big)$. It is noted that the function $\Phi^{(\kappa)}(\gamma_k^x,b_k^x)$ is concave and lower bound of $b_k^xB\log\big(1+\gamma_k^x\big)$, satisfying $\Phi^{x,(\kappa)}(\gamma_k^{x,(\kappa)},b_k^{x,(\kappa)})=b_k^{x,(\kappa)}B\log\big(1+\gamma_k^{x,(\kappa)}\big)$. Next, we rewrite (32c) as

$$\frac{(\rho_k^x)^2}{b_k^x}\geq\frac{-BN_0}{\varphi_k\ln(1-\epsilon)N}\gamma_k^x. \tag{34}$$

It is clear that $(\rho_k^x)^2/b_k^x$ is a quadratic-over-linear function, which is convex over the domain $(\rho_k^x>0,b_k^x>0)$. Thus, constraint (34) is iteratively convexified as

$$\frac{2\rho_k^{x,(\kappa)}}{b_k^{x,(\kappa)}}\rho_k^x-\Big(\frac{\rho_k^{x,(\kappa)}}{b_k^{x,(\kappa)}}\Big)^2b_k^x\geq\frac{-BN_0}{\varphi_k\ln(1-\epsilon)N}\gamma_k^x,\forall k,x. \tag{35}$$

Summing up, at iteration κ of round g, we solve the following convex program

$$\min_{\boldsymbol{\rho},\mathbf{f},\mathbf{b},\boldsymbol{\vartheta},\mathbf{t},\boldsymbol{\gamma}} \quad \eta E_g(\boldsymbol{\rho}^{\mathtt{ul}},\mathbf{f},\boldsymbol{\vartheta}^{\mathtt{ul}})+(1-\eta)T_g^{\mathrm{X}}(\mathbf{t}) \tag{36a}$$

$$\text{s.t.} \quad (25\text{e})-(25\text{h}),(26\text{c})-(26\text{e}),(32\text{d}),(33),(35), \tag{36b}$$

to generate the next feasible point $(\boldsymbol{\rho}^{(\kappa+1)},\mathbf{b}^{(\kappa+1)},\boldsymbol{\gamma}^{(\kappa+1)})$. The procedure is successively repeated until convergence. A pseudo-code of the path-following algorithm to solve (25) is given in Algorithm 2. An initial feasible point to start Algorithm 2 can be easily found by setting $\gamma_k^{x,(0)}=\gamma^{\min},b_k^{x,(0)}=1/K_g$ and $\rho_k^{x,(0)}=\sqrt{\frac{-BN_0}{\varphi_k\ln(1-\epsilon)N}\gamma_k^{x,(0)}b_k^{x,(0)}},\ \forall k\in\mathcal{K}_g,x\in\{\mathtt{dl},\mathtt{ul}\}$.

Convergence and Complexity Analysis: Algorithm 2

Algorithm 2 Proposed Path-Following Algorithm for Solving (25)

Initialization: Set $\kappa := 0$, a subset of selected UEs K_g, and generate a feasible point $(\boldsymbol{\rho}^{(0)}, \mathbf{b}^{(0)}, \boldsymbol{\gamma}^{(0)})$ for constraints in (36b).

1: **repeat**
2: Solve the convex program (36) to obtain the optimal solution, denoted by $(\boldsymbol{\rho}^*, \mathbf{f}^*, \mathbf{b}^*, \boldsymbol{\vartheta}^*, \mathbf{t}^*, \boldsymbol{\gamma}^*)$.
3: Update $(\boldsymbol{\rho}^{(\kappa+1)}, \mathbf{b}^{(\kappa+1)}, \boldsymbol{\gamma}^{(\kappa+1)}) := (\boldsymbol{\rho}^*, \mathbf{b}^*, \boldsymbol{\gamma}^*)$.
4: Set $\kappa := \kappa + 1$.
5: **until** Convergence
6: **Output:** $(\boldsymbol{\rho}^*, \mathbf{f}^*, \mathbf{b}^*)$ and the objective value (25a)

is developed using IA framework, where the convergence was provided in [30]. Specifically, let $\mathcal{F}^{(\kappa)} \triangleq \{\boldsymbol{\rho}, \mathbf{f}, \mathbf{b}, \boldsymbol{\vartheta}, \mathbf{t}, \boldsymbol{\gamma} | \text{constraints in (36b) hold}\}$ be the feasible set of (36) at iteration κ. By IA properties [41], Algorithm 2 produces a sequence $\{\boldsymbol{\rho}^{(\kappa)}, \mathbf{b}^{(\kappa)}, \boldsymbol{\gamma}^{(\kappa)}\}$ of improved solutions and a sequence of non-increasing objective values of (36) (and hence (25)). By [30, Theorem 1], Algorithm 2 is guaranteed to obtain a locally optimal solution for (25), which satisfies Karush-Kuhn-Tucker conditions when $\kappa \to \infty$. Problem (36) involves $7K_g + 3$ scalar decision variables and $12K_g + 3$ linear/quadratic constraints for synchronous communication, and $7K_g+1$ scalar decision variables and $10K_g+3$ linear/quadratic constraints for asynchronous communication. As a result, the per-iteration computational complexity of Algorithm 2 is $\mathcal{O}\big((12K_g + 3)^{2.5}(49K_g^2 + 54K_g + 12)\big)$ for synchronous communication and $\mathcal{O}\big((10K_g + 3)^{2.5}(49K_g^2 + 24K_g + 4)\big)$ for asynchronous communication, respectively. Similarly, the per-iteration complexity for the energy minimization problem is $\mathcal{O}\big((9K_g + 3)^{2.5}(49K_g^2 + 9K_g + 3)\big)$. We recall that only a subset of UEs K_g are selected at round g of Algorithm 1, making Algorithm 2 implementable.

V. Numerical Results

In this section, we first examine our theoretical results to validate the proposed FL's performance and then provide numerical results for FL over a wireless IoT network.

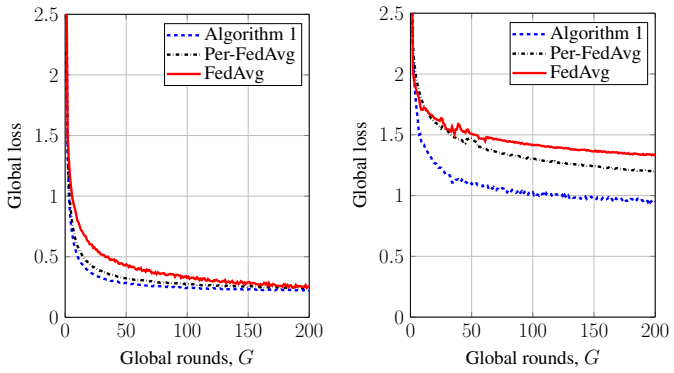
A. Numerical Results for the proposed FL Algorithm 1

Model and Loss Function: We consider a multinomial logistic regression with a convex loss function (cross-entropy error loss function). We decay the trade-off parameter μ after every global round g as $\mu_g = \frac{\mu_0}{1+0.1g}$ to balance the loss function and the weight-based proximal term, where μ_0 is the best value chosen from the set $\mu_0 \in \{100, 10, 1\}$.

Datasets: In order to examine the performance of the proposed FL algorithm in a heterogeneous setting, we consider both real (MNIST dataset [42]) and synthetic datasets with 100 UEs taking part in the training process. In MNIST dataset, each UE has samples of only two digits where the number of samples among UEs follows a power law. Synthetic data are generated by following the same settings in [15], [20]. In particular, samples $(\mathbf{X}_k, \mathbf{Y}_k)$ are generated by $y = \text{argmax}(\text{softmax}(\mathbf{W}_k x + \mathbf{b}_k))$ with $x \in \mathbb{R}^{60}$, $\mathbf{W}_k \in \mathbb{R}^{10\times 60}$ and $\mathbf{b} \in \mathbb{R}^{10}$. Each entry of $\mathbf{W}_k$ and $\mathbf{b}_k$ is modeled as $\mathcal{N}(\nu_k, 1)$ with $\nu_k \sim \mathcal{N}(0, \alpha)$. x is modeled as $\mathcal{N}(\tau_k, \Sigma)$, where $\tau_k \sim \mathcal{N}(B_k, 1)$ with $B_k \sim \mathcal{N}(0, \beta)$ and the matrix Σ is diagonal with $\Sigma_{jj} = \frac{1}{j^{1.2}}$. Here, α and β control how much the local model and the local data differ from each other, denoted Synthetic(α, β). The number of samples among UEs follows a power law. We summarize the statistics of datasets in Table III.

TABLE III: Statistics of MNIST and Synthetic Datasets

Dataset	#UEs	#Samples	Samples/UE mean	Samples/UE std
MNIST	100	66246	662	1595
Synthetic(0,0)	100	40893	408	911
Synthetic(1,1)	100	26144	261	333



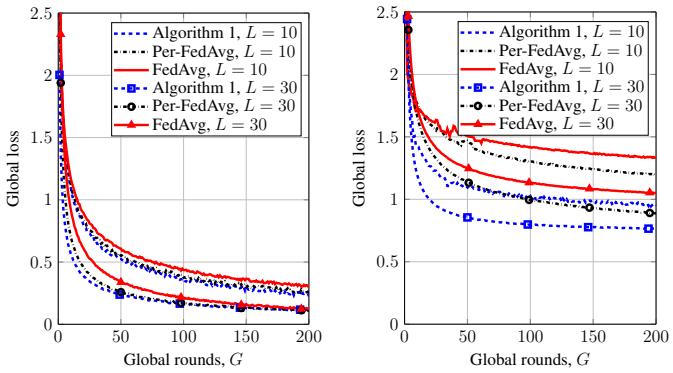
(a) MNIST dataset, with $\mathcal{B} = 60, L = 20$ and $K_g = 10, \forall g$.

(b) Synthetic(0,0) dataset, with $\mathcal{B} = 20, L = 10$ and $K_g = 10, \forall g$.

Fig. 3: Global loss values versus the number of global rounds.

Simulation Setting: To guarantee the convergence of Algorithm 1 (as mentioned in Remark 1), we consider a non-increasing learning rate as $\lambda_g = \frac{\lambda_0}{1+0.01g}$, where λ_0 is carefully chosen from the set $\lambda_0 \in \{0.1, 0.03, 0.01\}$. In all experiments, we generate an initial model as $\mathbf{w}_0 = 0$. We use batch sizes ($\mathcal{B}$) of 60 and 20 for MNIST and synthetic datasets, respectively. We then compare the performance of the proposed FL with FedAvg [8] and Per-FedAvg [16]. To conduct fair comparison, the subset of selected UEs in each round is the same for all considered FL algorithms. Each local data is randomly split to 80% for training and 20% for testing. The FL algorithms are implemented in TensorFlow.

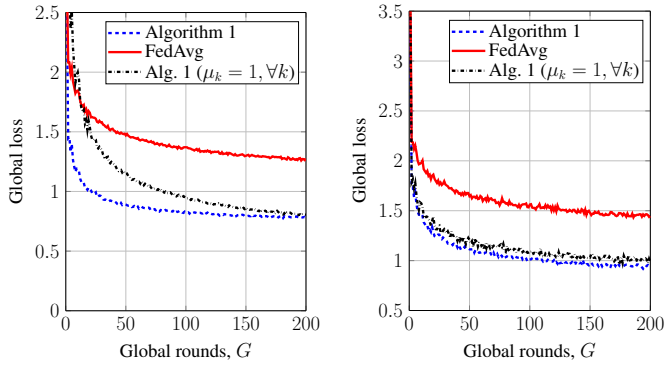
Fig. 3 shows the convergence of the proposed FL algorithm for different datasets. For FedAvg and Per-FedAvg, we also decay the learning rate as $\lambda_g = \frac{\lambda_0}{1+0.1g}$, where the initial value of λ_0 is carefully adjusted from the set $\{0.1, 0.03, 0.01\}$. The results demonstrate that Algorithm 1 outperforms FedAvg and Per-FedAvg on both MNIST and synthetic datasets. In particular, compared to FedAvg and Per-FedAvg, the improvements in terms of global loss are approximately 15.6% and 9.7% for the MNIST dataset in Fig. 3(a), and 28.7% and 24.6% for the Synthetic(0,0) dataset in Fig. 3(b), respectively. In addition to its better performance, it can be seen that Algorithm 1 is more stable and converges faster than FedAvg and Per-



(a) MNIST dataset, with $\mathcal{B} = 60$ and $K_g = 10,\ \forall g$.

(b) Synthetic(0,0) dataset, with $\mathcal{B} = 20$ and $K_g = 10,\ \forall g$.

Fig. 4: Global loss values versus the number of global rounds with the different numbers of local iterations.



(a) Synthetic(0,0) dataset, with $\mathcal{B} = 20, L = 20$ and $K_g = 10,\ \forall g$.

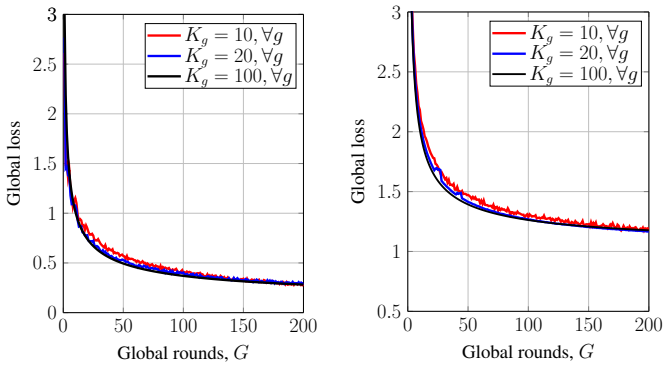
(b) Synthetic(1,1) dataset, with $\mathcal{B} = 20, L = 20$ and $K_g = 10,\ \forall g$.

Fig. 5: Global loss values versus the number of global rounds with different FL algorithms.

FedAvg. This further confirms the effectiveness of weight-based proximal term in (6) in dealing with the negative effects of the random sampling strategy to select user participation in the heterogeneous setting.

In Fig. 4, we investigate the effect of different numbers of local iterations $L \in \{10, 30\}$ on the performance of the proposed FL algorithm. It can be observed that Algorithm 1 outperforms FedAvg and Per-FedAvg in all settings. In addition, a larger L leads to faster convergence and allows to handle the instability of all FL algorithms with non-iid data in a better way. However, we recall that a very large value of L may result in the one-short averaging of FL algorithms [12]. In practice, it is beneficial to allow an appropriate value of L, which not only boosts the convergence speed but also prevents divergence of the proposed FL algorithm.

We next examine the impact of weight-based proximal term by fixing $\mu_k = \mu p_k = 1$ for all UEs, as shown in Fig. 5. We use Synthetic(0,0) and Synthetic(1,1) datasets with $\mathcal{B} = 20, L = 20$ and $K_g = 10,\ \forall g$. It can be seen that Algorithm 1 using the individual penalty constant (i.e., μp_k for UE k) achieves a significant performance gap over FedAvg



(a) MNIST dataset, with $\mathcal{B} = 60$ and $L = 5$.

(b) Synthetic(0,0) dataset, with $\mathcal{B} = 20$ and $L = 5$.

Fig. 6: Global loss values versus the number of global rounds with the different numbers of selected UEs in each round.

and Algorithm 1 with $\mu_k = 1, \forall k$ in terms of global loss in both cases. For Synthetic(0,0) dataset in Fig. 5(a), we can see that Algorithm 1 with a fixed penalty constant exhibits a lower convergence speed compared to an individual and flexible penalty constant. This is probably attributed to the fact that the local model of some UE k with a small number of training samples may strongly depend the latest global model, and thus it generates less useful local parameters to the next global aggregation. In other words, it may not reflect the actual local loss of those UEs that might slow down convergence on non-iid data. For highly heterogeneous Synthetic(1,1) data in Fig. 5(b), it is also worth noting that Algorithm 1 is less volatile than other algorithms.

A natural question that arises is how many user participants are optimal to accelerate the convergence speed of the proposed FL algorithm while still ensuring a minimum global loss. In Fig. 6, we select K_g from the set $K_g \in \{10, 20, 100\}$. We set $L = 5$ to reduce the local training time due to a large number of UEs taking part in the training process. It can be seen that the number of selected UEs in each global round has a limited impact on the convergence of Algorithm 1 in both MNIST and Synthetic datasets, which is also aligned with our theoretical result in Theorem 1 as well as the findings of [15]. This phenomenon plays an important role in improving the performance of the FL algorithm for wireless IoT networks in terms of energy consumption and FL training time, which will be elaborated next.

B. Numerical Results for the FL-supported wireless IoT network (Algorithm 2)

We consider a wireless IoT network, where $K_{\texttt{tot}} = 100$ UEs are uniformly deployed in a circle area of 1-km radius where the BS equipped with 4 antennas is located at its center. We use the same settings in Fig. 3(b) on Synthetic(0,0) dataset with $\mathcal{B} = 20, L = 10$, and the number of selected UEs in each round is fixed to as $K_g = 10, \forall g$. The number of global rounds is determined when Algorithm 1 converges. The large-scale fading is modeled as: $\varphi_k = 10^{\frac{-\texttt{PL}_k + \sigma_{\texttt{sh}} z}{10}}$, where the shadow fading is considered as a random variable $z \sim \mathcal{N}(0, 1)$ with

TABLE IV: Simulation Parameters

Parameter	Value
System bandwidth, B	1 MHz
Noise power spectral density, N_0	-174 dBm/Hz
SNR threshold, $\gamma^{\min}$	0 dB
Power budget at UEs, $P_k^{\max}, \forall k$	23 dBm
Power budget at BS, $P_{\mathtt{BS}}$	30 dBm
CPU frequency, $(\mathrm{f}_k^{\min},\ \mathrm{f}_k^{\max}), \forall k$	$(10^6,\ 3.10^9)$ cycles/s
Number of processing cycles, $c_k, \forall k$	10 cycles/bit
Effective capacitance coefficient, $\theta_k/2, \forall k$	10^{-28}
Maximum energy requirement, $\mathcal{E}_{\max}$	1 J
Maximum delay requirement, $\mathcal{T}_{\max}$	1 s
Global/local update size, $\mathtt{S}$	4.5 KB
Outage constant, ϵ	0.01

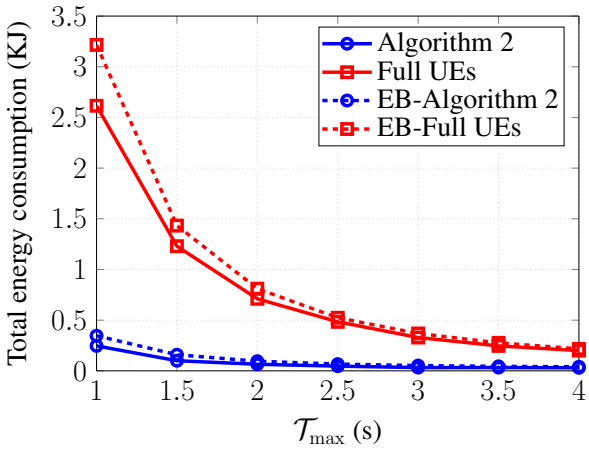


Fig. 7: Total energy consumption versus the maximum delay requirement.

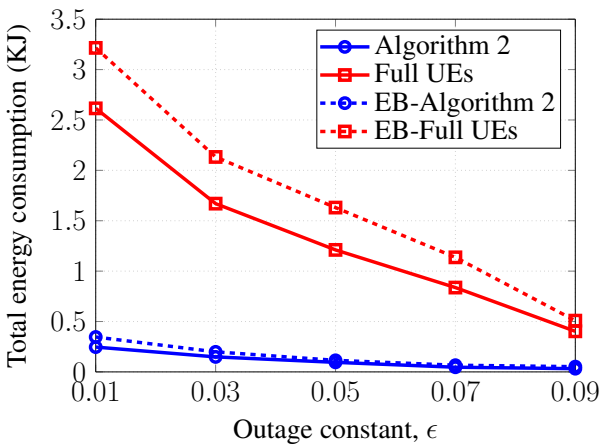


Fig. 8: Total energy consumption versus the outage constant.

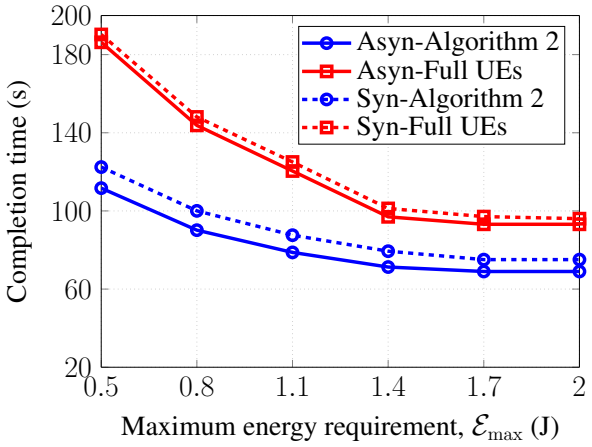


Fig. 9: Completion time versus maximum energy requirement.

$\sigma_{\mathtt{sh}} = 8$ dB. We consider the path loss as $\mathsf{PL}_k = 103.8 + 20.9\log(d_k)$, where d_k (in km) is the distance between BS and UE k. The training size D_k of UE k is uniformly distributed in [5, 10] Mbits [22]. Unless specifically stated otherwise, other parameters are given in Table IV, following the studies in [22], [23], [28], [29].

For comparison purpose, we investigate two other schemes: i) "Equal bandwidth (EB):" In every communication round, each UE is allocated with equal bandwidth, i.e., $b_k^x = 1/K_g, \forall k \in \mathcal{K}_g, x \in \{\mathtt{dl}, \mathtt{ul}\}$ and $b_k^x = 1/K_{\mathtt{tot}}, \forall k \in \mathcal{K}_{\mathtt{tot}}, x \in \{\mathtt{dl}, \mathtt{ul}\}$ for partial and full user participation, respectively; ii) "Full UEs:" In every global round, all UEs participate the training process, studied in [22], [28], [29]. The solutions of these schemes can readily be obtained using Algorithm 2 after some slight modifications. On average, Algorithm 2 converges in about 6 iterations for $K_g = 10$.

Energy Consumption Minimization: In Fig. 7, we plot the trade-off between total energy consumption and maximum delay requirement for executing one communication round. The observations from the figure are as follows. First, one can see that the proposed Algorithm 2 offers a remarkable gain in the total energy consumption compared with the full user participation in the range of $\mathcal{T}_{\max} \in [1,\ 4]$ s. Although a larger K_g can slightly accelerate the convergence of the FL process (e.g., see Fig. 6), it requires a substantially high power consumption in both computation and communication phases. In addition, a lower number of UEs taking part in the training process will exploit limited resources (i.e., bandwidth) more efficiently. Second, increasing $\mathcal{T}_{\max}$ leads to a significant reduction in the effective energy consumption. This is reasonable because the higher the maximum delay requirement, the lower the optimal value of power consumption and CPU frequency can be obtained while still satisfying constraint (25c), leading to energy saving. Third, joint optimization of bandwidth results in better performance, especially when the delay requirement is more stringent.

The impact of the outage constant on the total energy consumption is shown in Fig. 8. Increasing the threshold ϵ leads to a dramatic decrease in energy consumption, e.g., by up to 39% and 36% for Algorithm 2 and "Full UEs", respectively, with $\epsilon = 0.01$ in comparison with that of $\epsilon = 0.03$. This outcome is not surprising because with a higher value of ϵ, less power at UEs is required to meet the condition (31), i.e., by a factor of $-1/\log(1-\epsilon)$. However, we recall that in practice, a small value of ϵ should be considered to ensure high reliability of the uplink transmission. Nevertheless, Algorithm 2 still achieves the best performance out of the schemes considered.

Completion Time Minimization: In Fig. 9, we show the impact of UEs' maximum energy consumption $\mathcal{E}_{\max}$ on the effective completion time to implement the FL algorithm. We

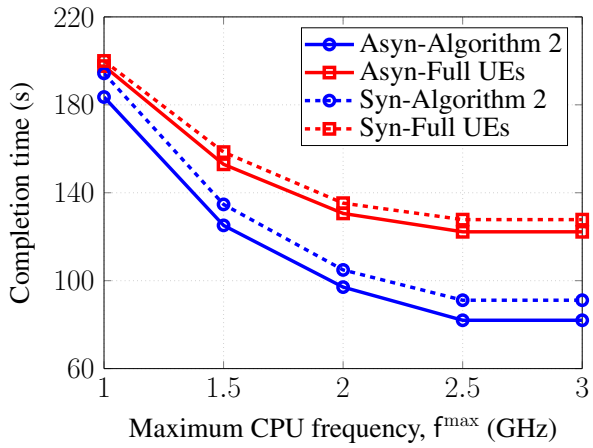


Fig. 10: Completion time versus maximum CPU frequency of each UE.

consider both synchronous and asynchronous communications. First, as can be seen that increasing $\mathcal{E}_{\max}$ requires much less completion time. In order to minimize the completion time, all UEs must run at a higher CPU frequency and transmission rate as possible all the time, but this will certainly increase the energy consumption. Consequently, increasing the UE's maximum energy requirement will result in a greater feasible region of the optimization problem (25). Another important observation is that asynchronous communication is superior to the synchronous one in both schemes. This is not difficult to see that the latter has to wait for the slowest UEs to complete local training updates before communicating to BS, leading to serious delays compared to the former, especially in a highly heterogeneous environment. Fig. 10 illustrates the impact of UEs' maximum CPU frequency by setting $\mathsf{f}^{\max} = \mathsf{f}_k^{\max}, \forall k$. As expected, the effective training time increases when $\mathsf{f}^{\max}$ decreases, as it requires more time to complete the local computation.

One can observe from Figs. 9 and 10 that at $\mathcal{E}_{\max} = 0.5$ J and $\mathsf{f}^{\max} = 3$ GHz, significant gains of up to 40% and 33% are offered by "Asyn-Algorithm 2" compared to "Asyn-Full UEs," respectively. This is the result of using a small number of active UEs per round, since the communication phase in Algorithm 2 with higher bandwidth for each UE is willing to consume less energy than the local computation, allowing to use higher UEs' CPU frequency.

VI. Conclusion

In this paper, we have proposed an efficient FL algorithm relying on a weight-based proximal term, which is an extension of FedAvg, to tackle the heterogeneity across UEs data and UEs' characteristics in federated networks. The proposed FL algorithm allows a small number of UEs per round to be participated in the training process based on the unbiased sampling strategy. Under the assumption of strongly convex and smooth FL's problem, we have theoretically characterized the convergence of the proposed FL algorithm. Empirical results on both real and synthetic datasets have verified our theoretical findings and demonstrated the stabilization and robustness of the proposed FL algorithm compared to FedAvg in highly heterogeneous environments. Next, we have formulated a wireless IoT resource allocation problem employing the proposed FL algorithm to minimize either total energy consumption or completion time. To deal with the non-convex nature of the problem and uncertainty of wireless channel, we have developed a new path-following algorithm based on IA framework to obtain at least a locally optimal solution. Numerical results are provided to confirm the effectiveness of the proposed FL algorithm over existing baseline approaches (e.g., full user participation and equal bandwidth) in wireless IoT networks with limited resources.

Next-generation wireless IoT networks will be extremely dynamic and complex due to the emergence of new applications and mobile broadband services such as video streaming and online gaming, calling for inclusive and innovative approaches. Thus, interesting future works include: i) Novel gradient coding schemes for addressing the problem of stragglers; and ii) Personalized FL models to adapt to newly collected data. In addition, it would be interesting to develop prototypes/testbeds to validate the proposed FL's performance presented in this work.

Appendix A: Proof of Lemma 1

Given that each activation of the sampling strategy is independent with the rest and following [15], we have

$$\begin{aligned}&\mathbb{E}\big\{\|\mathbf{w}_{g+1} - \bar{\mathbf{w}}_{g+1}\|^2 \big| k \in \mathcal{K}_g\big\}\\ &= \frac{1}{K_g^2}\sum_{k\in\mathcal{K}_g}\mathbb{E}\big\{\|\mathbf{w}_{g+1}^k - \bar{\mathbf{w}}_{g+1}\|^2\big\}.\end{aligned} \tag{A.1}$$

From Definition 2 for an unbiased sampling strategy, it follows that

$$\begin{aligned}&\frac{1}{K_g^2}\sum_{k\in\mathcal{K}_g}\mathbb{E}\big\{\|\mathbf{w}_{g+1}^k - \bar{\mathbf{w}}_{g+1}\|^2\big\}\\ &= \frac{1}{K_g}\sum_{k\in\mathcal{K}_{\text{tot}}} p_k\mathbb{E}\big\{\|\mathbf{w}_{g+1}^k - \bar{\mathbf{w}}_{g+1}\|^2\big\}.\end{aligned} \tag{A.2}$$

We know that $\mathbf{w}_g$ is the same for all UEs (i.e., Step 7 of Algorithm 1). Hence,

$$\begin{aligned}&\sum_{k\in\mathcal{K}_{\text{tot}}} p_k\|\mathbf{w}_{g+1}^k - \bar{\mathbf{w}}_{g+1}\|^2\\ &= \sum_{k\in\mathcal{K}_{\text{tot}}} p_k\|(\mathbf{w}_{g+1}^k - \bar{\mathbf{w}}_g) - (\bar{\mathbf{w}}_{g+1} - \bar{\mathbf{w}}_g)\|^2\\ &= \sum_{k\in\mathcal{K}_{\text{tot}}} p_k\|\mathbf{w}_{g+1}^k - \bar{\mathbf{w}}_g\|^2 - \|\bar{\mathbf{w}}_{g+1} - \bar{\mathbf{w}}_g\|^2\\ &\le \sum_{k\in\mathcal{K}_{\text{tot}}} p_k\|\mathbf{w}_{g+1}^k - \bar{\mathbf{w}}_g\|^2,\end{aligned} \tag{A.3}$$

where $\sum_{k\in\mathcal{K}_{\text{tot}}} p_k = 1$ and $\sum_{k\in\mathcal{K}_{\text{tot}}} p_k\mathbf{w}_{g+1}^k = \bar{\mathbf{w}}_{g+1}$. Substituting (A.3) into (A.2) and from Assumption 3, we have

$$\begin{aligned}&\frac{1}{K_g}\sum_{k\in\mathcal{K}_{\text{tot}}} p_k\mathbb{E}\big\{\|\mathbf{w}_{g+1}^k - \bar{\mathbf{w}}_{g+1}\|^2\big\}\\ &\le \frac{1}{K_g}\sum_{k\in\mathcal{K}_{\text{tot}}} p_k\mathbb{E}\big\{\|\mathbf{w}_{g+1}^k - \bar{\mathbf{w}}_g\|^2\big\}\end{aligned}$$

$$\leq \frac{1}{K_g}\sum_{k\in\mathcal{K}_{\text{tot}}} p_k\mathbb{E}\{\|\mathbf{w}_{g+1}^k-\mathbf{w}_g^k\|^2\}$$
$$\leq \frac{L\lambda_g^2}{K_g}\sum_{k\in\mathcal{K}_{\text{tot}}} p_k\sum_{\ell=0}^{L-1}\mathbb{E}\{\|\nabla F_k'(\mathbf{w}_{g,\ell}^k,\xi_{g,\ell}^k)\|^2\}$$
$$\leq \frac{L^2\lambda_g^2\delta}{K_g}\leq \frac{L^2\lambda_0^2\delta}{K_g(1+ga)^2}\ \big(\text{due to } \lambda_g\leq\frac{\lambda_0}{(1+ga)}\big), \quad \text{(A.4)}$$

where the third inequality is obtained due to the fact that each selected UE independently performs L local iterations.

Appendix B: Proof of Theorem 1

For the strong convexity of F', if it has a minimizer $\mathbf{w}^*$, then

$$\mathbb{E}\{F'(\mathbf{w}_{g+1})\}-F'(\mathbf{w}^*)\leq\frac{\Gamma}{2}\mathbb{E}\{\|\mathbf{w}_{g+1}-\mathbf{w}^*\|^2\}, \quad \text{(B.1)}$$

which follows from the upper bound in Assumption 1 at $\mathbf{w}=\mathbf{w}_{g+1}$ and $\bar{\mathbf{w}}=\mathbf{w}^*$. We know that $F'(\mathbf{w}_{g+1})=F(\mathbf{w}_{g+1})+\sum_{k\in\mathcal{K}_g}\frac{\mu p_k}{2}\|\mathbf{w}_{g+1}-\mathbf{w}_g\|^2$. At the optimum, F' and F share the same minimizer $\mathbf{w}^*$ (i.e., $F'(\mathbf{w}^*)=F(\mathbf{w}^*)$), and thus

$$\begin{aligned}\mathbb{E}\{F(\mathbf{w}_{g+1})\}-F(\mathbf{w}^*)&\leq\frac{\Gamma}{2}\mathbb{E}\{\|\mathbf{w}_{g+1}-\mathbf{w}^*\|^2\}\\&\quad-\sum_{k\in\mathcal{K}_g}\frac{\mu p_k}{2}\|\mathbf{w}_{g+1}-\mathbf{w}_g\|^2\\&\leq\frac{\Gamma}{2}\mathbb{E}\{\|\mathbf{w}_{g+1}-\mathbf{w}^*\|^2\}. \quad \text{(B.2)}\end{aligned}$$

In addition, we have

$$\begin{aligned}&\mathbb{E}\{\|\mathbf{w}_{g+1}-\mathbf{w}^*\|^2\}\\&=\mathbb{E}\{\|(\mathbf{w}_{g+1}-\bar{\mathbf{w}}_{g+1})+(\bar{\mathbf{w}}_{g+1}-\mathbf{w}^*)\|^2\}\\&=\mathbb{E}\{\|(\mathbf{w}_{g+1}-\bar{\mathbf{w}}_{g+1})\|^2\}+\mathbb{E}\{\|(\bar{\mathbf{w}}_{g+1}-\mathbf{w}^*)\|^2\}, \quad \text{(B.3)}\end{aligned}$$

where $\mathbb{E}\{\mathbf{w}_{g+1}\}-\bar{\mathbf{w}}_{g+1}=0$ due to (13). We note that $\mathbb{E}\{\|\mathbf{w}_{g+1}-\bar{\mathbf{w}}_{g+1}\|^2\}$ is already bounded by (14). We now focus on the bound of $\mathbb{E}\{\|\bar{\mathbf{w}}_{g+1}-\mathbf{w}^*\|^2\}$. Let us start by rewriting it as

$$\begin{aligned}\|\bar{\mathbf{w}}_{g+1}-\mathbf{w}^*\|^2&=\|\bar{\mathbf{w}}_g-\mathbf{w}^*\|^2+\sum_{\ell=0}^{L-1}\Big(\lambda_g^2\|\nabla F'(\bar{\mathbf{w}}_{g,\ell})\|^2\\&\quad-2\lambda_g\left\langle\nabla F'(\bar{\mathbf{w}}_{g,\ell}),\bar{\mathbf{w}}_{g,\ell}-\mathbf{w}^*\right\rangle\Big). \quad \text{(B.4)}\end{aligned}$$

Let Assumptions 1 and 2 hold. By $Q_{g+1}\triangleq\mathbb{E}\{\|\bar{\mathbf{w}}_{g+1}-\mathbf{w}^*\|^2\}$, $\sum_{k\in\mathcal{K}_{\text{tot}}}\mu_k=\mu\sum_{k\in\mathcal{K}_{\text{tot}}}p_k=\mu$ and [43, Theorem 2.1.12], we have

$$\begin{aligned}Q_{g+1}&=\Big(1-2\lambda_g\frac{\mu\Gamma}{\mu+\Gamma}\Big)Q_g\\&\quad-2\lambda_g\frac{\mu\Gamma}{\mu+\Gamma}\sum_{\ell=1}^{L-1}\mathbb{E}\{\|\bar{\mathbf{w}}_{g,\ell}-\mathbf{w}^*\|^2\}\\&\quad+\Big(\lambda_g^2-2\lambda_g\frac{1}{\mu+\Gamma}\Big)\sum_{\ell=0}^{L-1}\mathbb{E}\{\|\nabla F'(\bar{\mathbf{w}}_g)\|^2\}. \quad \text{(B.5)}\end{aligned}$$

If we choose the learning rate $\lambda_g\leq\frac{2}{\mu+\Gamma}$, yielding $0\leq 1-2\lambda_g\frac{\mu\Gamma}{\mu+\Gamma}<1$ and $\lambda_g^2-2\lambda_g\frac{1}{\mu+\Gamma}\leq 0$. As a result, we have

$$\begin{aligned}Q_{g+1}&\leq\Big(1-2\lambda_g\frac{\mu\Gamma}{\mu+\Gamma}\Big)Q_g\\&\leq\prod_{i=0}^{g}\Big(1-2\lambda_i\frac{\mu\Gamma}{\mu+\Gamma}\Big)\|\mathbf{w}_0-\mathbf{w}^*\|^2. \quad \text{(B.6)}\end{aligned}$$

Combining the results from (14) and (B.6), the expected upper bound of (B.2) is

$$\begin{aligned}&\mathbb{E}\{F(\mathbf{w}_{g+1})\}-F(\mathbf{w}^*)\\&\leq\frac{\Gamma}{2}\Big(\frac{L^2\lambda_0^2\delta}{K_g(1+ga)^2}+\prod_{i=0}^{g}\big(1-\frac{2\lambda_i\mu\Gamma}{\mu+\Gamma}\big)\|\mathbf{w}_0-\mathbf{w}^*\|^2\Big). \quad \text{(B.7)}\end{aligned}$$

With the learning rate $\lambda_i\leq\frac{\lambda_0}{(1+ia)}$, we obtain the result in (15).

Appendix C: Proof of Lemma 2

At first, we note that $\|\rho_k^x\bar{\mathbf{h}}_k\|^2$ is an exponentially distributed random variable with parameter $\frac{1}{N(\rho_k^x)^2}$. It follows that

$$\begin{aligned}\mathsf{Prob}\Big(\frac{\varphi_k\|\rho_k^x\bar{\mathbf{h}}_k\|^2}{b_k^xBN_0}<\gamma_k^x\Big)&=\mathsf{Prob}\Big(\|\rho_k^x\bar{\mathbf{h}}_k\|^2<\frac{\gamma_k^xb_k^xBN_0}{\varphi_k}\Big)\\&=\int_0^{\frac{\gamma_k^xb_k^xBN_0}{\varphi_k}}\frac{1}{N(\rho_k^x)^2}e^{-\frac{1}{N(\rho_k^x)^2}x}dx\\&=1-e^{-\frac{\gamma_k^xb_k^xBN_0}{\varphi_k(\rho_k^x)^2N}}. \quad \text{(C.1)}\end{aligned}$$

As a result, we have

$$\begin{aligned}&\mathsf{Prob}\Big(\frac{\varphi_k\|\rho_k^x\bar{\mathbf{h}}_k\|^2}{b_k^xBN_0}<\gamma_k^x\Big)\leq\epsilon\\\Leftrightarrow&1-e^{-\frac{\gamma_k^xb_k^xBN_0}{\varphi_k(\rho_k^x)^2N}}\leq\epsilon\\\Leftrightarrow&\frac{\varphi_kN\ln(1-\epsilon)}{BN_0}+\frac{\gamma_k^xb_k^x}{(\rho_k^x)^2}\leq 0. \quad \text{(C.2)}\end{aligned}$$

For given b_k^x and ρ_k^x, the left-hand side of (C.2) is an increase function in γ_k^x. Therefore, $\log\big(1+\frac{\varphi_k\|\rho_k^x\bar{\mathbf{h}}_k\|^2}{b_k^xBN_0}\big)=\log(1+\gamma_k^x)$ if the equality in (C.2) holds.

Appendix D: Proof of Inequality (33)

By [44, Appendix A], the function $\Psi(x,y)\triangleq\ln(1+1/x)/y$ is convex in the domain $(x>0,y>0)$. Thus, $\Psi(x,y)$ is innerly approximated as:

$$\begin{aligned}\Psi(x,y)\geq&\Psi(x^{(\kappa)},y^{(\kappa)})-\Big\langle\big[\nabla_{x^{(\kappa)}}\Psi(x^{(\kappa)},y^{(\kappa)}),\\&\nabla_{y^{(\kappa)}}\Psi(x^{(\kappa)},y^{(\kappa)})\big]\big[(x-x^{(\kappa)}),\ (y-y^{(\kappa)})\big]^T\Big\rangle\\=&2\Psi(x^{(\kappa)},y^{(\kappa)})+\frac{1}{y^{(\kappa)}(x^{(\kappa)}+1)}\\&-\frac{1}{y^{(\kappa)}x^{(\kappa)}(x^{(\kappa)}+1)}x-\frac{\Psi(x^{(\kappa)},y^{(\kappa)})}{y^{(\kappa)}}y. \quad \text{(D.1)}\end{aligned}$$

By substituting $(\gamma_k^x,\gamma_k^{x,(\kappa)})=(x^{-1},(x^{(\kappa)})^{-1})$ and $(b_k^x,b_k^{x,(\kappa)})=(y^{-1},(y^{(\kappa)})^{-1})$ into (D.1), we obtain (33).

REFERENCES

[1] S. Wiedemann, K. Müller, and W. Samek, "Compact and computationally efficient representation of deep neural networks," *IEEE Trans. Neural Networks and Learning Systems*, vol. 31, no. 3, pp. 772–785, 2020.

[2] R. Taylor, D. Baron, and D. Schmidt, "The world in 2025-predictions for the next ten years," in *Proc. 10th International Microsystems, Packaging, Assembly and Circuits Technology Conference (IMPACT)*, 2015, pp. 192–195.

[3] S. K. Sharma and X. Wang, "Live data analytics with collaborative edge and cloud processing in wireless iot networks," *IEEE Access*, vol. 5, pp. 4621–4635, 2017.

[4] A. Ghasempour, "Internet of things in smart grid: Architecture, applications, services, key technologies, and challenges," *Inventions Journal*, vol. 4, no. 1, pp. 1–12, 2019.

[5] *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016-2021*, Mar. 2017. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html

[6] H. Yuan and M. Zhou, "Profit-maximized collaborative computation offloading and resource allocation in distributed cloud and edge computing systems," *IEEE Trans. Automation Science and Engineering*, 2020, doi: 10.1109/TASE.2020.3000946.

[7] H. Yuan, J. Bi, W. Tan, M. Zhou, B. H. Li, and J. Li, "TTSA: An effective scheduling approach for delay bounded tasks in hybrid clouds," *IEEE Trans. Cybernetics*, vol. 47, no. 11, pp. 3658–3668, 2017.

[8] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Inter. Conf. Artificial Intelligence and Statistics*, Apr. 2017, pp. 1273–1282.

[9] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

[10] M. Jaggi, V. Smith, M. Takac, J. Terhorst, S. Krishnan, T. Hofmann, and M. I. Jordan, "Communication-efficient distributed dual coordinate ascent," in *Proc. 27th Int. Conf. Neural Inform. Process. Syst. (NIPS)*, 2014, p. 3068–3076.

[11] S. U. Stich, "Local SGD converges fast and communicates little," in *Proc. International Conference on Learning Representations*, 2019.

[12] Y. Zhang, J. C. Duchi, and M. J. Wainwright, "Communication-efficient algorithms for statistical optimization," *Journal of Machine Learning Research*, vol. 14, no. 1, p. 3321–3363, 2013.

[13] X. Wang, C. Wang, X. Li, V. C. M. Leung, and T. Taleb, "Federated deep reinforcement learning for internet of things with decentralized cooperative edge caching," *IEEE Internet of Things J.*, pp. 1–1, 2020.

[14] J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang, "Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory," *IEEE Internet of Things J.*, vol. 6, no. 6, pp. 10 700–10 714, 2019.

[15] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of FedAvg on non-iid data," in *Proc. International Conference on Learning Representations (ICLR)*, 2020.

[16] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning: A meta-learning approach," Feb. 2020. [Online]. Available: http://arxiv.org/abs/2002.07948

[17] 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. Select. Areas in Commun.*, vol. 37, no. 6, pp. 1205–1221, 2019.

[18] J. Mills, J. Hu, and G. Min, "Communication-efficient federated learning for wireless edge intelligence in IoT," *IEEE Internet of Things J.*, pp. 1–1, 2019. Early Access.

[19] C. Xie, S. Koyejo, and I. Gupta, "Asynchronous federated optimization," *arXiv preprint arXiv:1903.03934*, 2019.

[20] T. Li, A. K. Sahu, M. Sanjabi, M. Zaheer, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. of the 1st Adaptive & Multitask Learning, ICML Workshop*, Long Beach, CA, 2019.

[21] 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.

[22] C. Dinh, N. H. Tran, M. N. H. Nguyen, C. S. Hong, W. Bao, A. Y. Zomaya, and V. Gramoli, "Federated learning over wireless networks: Convergence analysis and resource allocation," 2019. [Online]. Available: https://arxiv.org/abs/1910.13067

[23] 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," 2019. [Online]. Available: https://arxiv.org/abs/1909.07972

[24] Q. Zeng, Y. Du, K. K. Leung, and K. Huang, "Energy-efficient radio resource allocation for federated edge learning," 2019. [Online]. Available: https://arxiv.org/abs/1907.06040

[25] K. Yang, T. Jiang, Y. Shi, and Z. Ding, "Federated learning via over-the-air computation," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 2022–2035, 2020.

[26] G. Zhu, Y. Wang, and K. Huang, "Broadband analog aggregation for low-latency federated edge learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 491–506, 2020.

[27] Y. Sun, S. Zhou, and D. Gunduz, "Energy-aware analog aggregation for federated learning with redundant data," 2019. [Online]. Available: https://arxiv.org/abs/1911.00188

[28] 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," 2019. [Online]. Available: https://arxiv.org/abs/1909.12567

[29] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, "Energy efficient federated learning over wireless communication networks," 2019. [Online]. Available: https://arxiv.org/abs/1911.02417

[30] B. R. Marks and G. P. Wright, "A general inner approximation algorithm for nonconvex mathematical programs," *Operations Research*, vol. 26, no. 4, pp. 681–683, July-Aug. 1978.

[31] S. Bi, J. Lyu, Z. Ding, and R. Zhang, "Engineering radio maps for wireless resource management," *IEEE Wireless Commun.*, vol. 26, no. 2, pp. 133–141, 2019.

[32] Qualcomm, "We are making on-device AI ubiquitous," *IEEE Wireless Commun.*, 2017. [Online]. Available: https://www.qualcomm.com/news/onq/2017/08/16/we-are-makingdevice-ai-ubiquitous

[33] S. Shalev-Shwartz and S. Ben-David, *Understanding machine learning: From theory to algorithms*. Cambridge University Press, 2014.

[34] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org.

[35] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. COMPSTAT'2010. Springer*, 2010, pp. 177–186.

[36] C. Ma, J. Konecny, M. Jaggi, V. S. M. I. Jordan, P. Richtarik, and M. Takac, "Distributed optimization with arbitrary local solvers," *Optimization Methods & Software*, vol. 32, no. 4, pp. 813–848, July 2017.

[37] A. P. Miettinen and J. K. Nurminen, "Energy efficiency of mobile clients in cloud computing," in *USENIX HotCloud'10*, Berkeley, CA, USA, 2010.

[38] T. D. Burd and R. W. Brodersen, "Processor design for portable systems," *Journal of VLSI Signal Processing System*, vol. 13, no. 23, pp. 203–221, Aug. 1996.

[39] A. Ashikhmin, L. Li, and T. L. Marzetta, "Interference reduction in multi-cell massive MIMO Systems with large-scale fading precoding," *IEEE Trans. Infor. Theory*, vol. 64, no. 9, pp. 6340–6361, Sept. 2018.

[40] S. Kandukuri and S. Boyd, "Optimal power control in interference-limited fading wireless channels with outage-probability specifications," *IEEE Trans. Wireless Commun.*, vol. 1, no. 1, pp. 46–55, 2002.

[41] A. Beck, A. Ben-Tal, and L. Tetruashvili, "A sequential parametric convex approximation method with applications to nonconvex truss topology design problems," *J. Global Optim.*, vol. 47, no. 1, pp. 29–51, May 2010.

[42] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Procee. IEEE*, vol. 86, no. 11, p. 2278–2324, 1998.

[43] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*. Springer Science & Business Media, 2013, vol. 87.

[44] V.-D. Nguyen, H. V. Nguyen, O. A. Dobre, and O.-S. Shin, "A new design paradigm for secure full-duplex multiuser systems," *IEEE J. Select. Areas Commun.*, vol. 36, no. 7, pp. 1480–1498, July 2018.

Van-Dinh Nguyen (S'14-M'19) received the B.E. degree in electrical engineering from Ho Chi Minh City University of Technology, Vietnam, in 2012 and the M.E. and Ph.D. degrees in electronic engineering from Soongsil University, Seoul, South Korea, in 2015 and 2018, respectively. He is currently a Research Associate with the Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg. He was a Postdoc Researcher and a Lecturer with Soongsil University, a Postdoctoral Visiting Scholar with University of Technology Sydney, AUS (July-August 2018) and a Ph.D. Visiting Scholar with Queen's University Belfast, U.K. (June-July 2015 and August 2016). His current research activity is focused on fog/edge computing, Internet of Things, 5G networks and machine learning for wireless communications.

Dr. Nguyen received several best conference paper awards, IEEE Transaction on Communications Exemplary Reviewer 2018 and IEEE GLOBECOM Student Travel Grant Award 2017. He has authored or co-authored in some 40 papers published in international journals and conference proceedings. He has served as a reviewer for many top-tier international journals on wireless communications, and has also been a Technical Programme CommitteeMember for several flag-ship international conferences in the related fields. He is an Editor for the IEEE Open Journal of the Communications Society and IEEE Communications Letters.

Shree Krishna Sharma (S'12-M'15-SM'18) is currently Research Scientist at the SnT, University of Luxembourg. Prior to this, he worked as a Postdoctoral Fellow at the Western University, Canada, and as a Research Associate at the SnT being involved in different European, national and ESA projects after receiving his PhD degree in Wireless Communications from the University of Luxembourg in 2014. His current research interests include 5G and beyond wireless, Internet of Things, machine learning, edge computing and optimization of distributed communications, computing and caching resources.

He has published about 100 technical papers in scholarly journals, international conferences and book chapters, and has over 2000 google scholar citations. He is a Senior Member of IEEE and is the recipient of several prestigious awards including "2018 EURASIP JWCN Best Paper Award", "CROWNCOM 2015 Best Paper Award" and "FNR Award for Outstanding PhD Thesis 2015", and the co-recipient of "FNR Award for Outstanding Scientific Publication 2019". He has been serving as a Reviewer for several international journals and conferences; as a TPC member for a number of international conferences including IEEE ICC, IEEE GLOBECOM, IEEE PIMRC, IEEE VTC and IEEE ISWCS; and an Associate Editor for IEEE Access journal. He co-organized a special session in IEEE PIMRC 2017, a workshop in IEEE SECON 2019, worked as a Track co-chair for IEEE VTC-fall 2018 conference, and published an IET book on "Satellite Communications in the 5G Era" as a lead editor.

Thang X. Vu (M'15) was born in Hai Duong, Vietnam. He received the B.S. and the M.Sc., both in Electronics and Telecommunications Engineering, from the VNU University of Engineering and Technology, Vietnam, in 2007 and 2009, respectively, and the Ph.D. in Electrical Engineering from the University Paris-Sud, France, in 2014.

In 2010, he received the Allocation de Recherche fellowship to study Ph.D. in France. From September 2010 to May 2014, he was with the Laboratory of Signals and Systems (LSS), a joint laboratory of CNRS, CentraleSupelec and University Paris-Sud XI, France. From July 2014 to January 2016, he was a postdoctoral researcher with the Information Systems Technology and Design (ISTD) pillar, Singapore University of Technology and Design (SUTD), Singapore. Currently, he is a research scientist at the Interdisciplinary Center for Security, Reliability and Trust (SnT), University of Luxembourg. His research interests are in the field of wireless communications, with particular interests of 5G networks and beyond, machine learning for communications and cross-layer resources optimization. He was a recipient of the SigTelCom 2019 best paper award.

Symeon Chatzinotas is currently Full Professor/Chief Scientist I and Co-Head of the SIGCOM Research Group at SnT, University of Luxembourg. In the past, he has been a Visiting Professor at the University of Parma, Italy and he was involved in numerous Research and Development projects for the National Center for Scientific Research Demokritos, the Center of Research and Technology Hellas and the Center of Communication Systems Research, University of Surrey. He received the M.Eng. degree in telecommunications from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 2003, and the M.Sc. and Ph.D. degrees in electronic engineering from the University of Surrey, Surrey, U.K., in 2006 and 2009, respectively. He was a co-recipient of the 2014 IEEE Distinguished Contributions to Satellite Communications Award, the CROWNCOM 2015 Best Paper Award and the 2018 EURASIC JWCN Best Paper Award. He has (co-)authored more than 400 technical papers in refereed international journals, conferences and scientific books. He is currently in the editorial board of the IEEE Open Journal of Vehicular Technology and the International Journal of Satellite Communications and Networking.

Björn Ottersten (S'87–M'89–SM'99–F'04) was born in Stockholm, Sweden, in 1961. He received the M.S. degree in electrical engineering and applied physics from Linköping University, Linköping, Sweden, in 1986, and the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, USA, in 1990. He has held research positions with the Department of Electrical Engineering, Linköping University, the Information Systems Laboratory, Stanford University, the Katholieke Universiteit Leuven, Leuven, Belgium, and the University of Luxembourg, Luxembourg. From 1996 to 1997, he was the Director of Research with ArrayComm, Inc., a start-up in San Jose, CA, USA, based on his patented technology. In 1991, he was appointed Professor of signal processing with the Royal Institute of Technology (KTH), Stockholm, Sweden. Dr. Ottersten has been Head of the Department for Signals, Sensors, and Systems, KTH, and Dean of the School of Electrical Engineering, KTH. He is currently the Director for the Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg.

He is a recipient of the IEEE Signal Processing Society Technical Achievement Award and the European Research Council advanced research grant twice. He has co-authored journal papers that received the IEEE Signal Processing Society Best Paper Award in 1993, 2001, 2006, 2013, and 2019, and 8 IEEE conference papers best paper awards. He has been a board member of IEEE Signal Processing Society, the Swedish Research Council and currently serves of the boards of EURASIP and the Swedish Foundation for Strategic Research. He has served as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING and the Editorial Board of the IEEE Signal Processing Magazine. He is currently a member of the editorial boards of IEEE Open Journal of Signal Processing, EURASIP Signal Processing Journal, EURASIP Journal of Advances Signal Processing and Foundations and Trends of Signal Processing. He is a fellow of EURASIP.