FedFog: Network-Aware Optimization of Federated Learning over Wireless Fog-Cloud Systems

Van-Dinh Nguyen, Symeon Chatzinotas, Björn Ottersten, and Trung Q. Duong

Abstract—Federated learning (FL) is capable of performing large distributed machine learning tasks across multiple edge users by periodically aggregating trained local parameters. To address key challenges of enabling FL over a wireless fog-cloud system (e.g., non-i.i.d. data, users' heterogeneity), we first propose an efficient FL algorithm based on Federated Averaging (called `FedFog`) to perform the local aggregation of gradient parameters at fog servers and global training update at the cloud. Next, we employ `FedFog` in wireless fog-cloud systems by investigating a novel network-aware FL optimization problem that strikes the balance between the global loss and completion time. An iterative algorithm is then developed to obtain a precise measurement of the system performance, which helps design an efficient stopping criteria to output an appropriate number of global rounds. To mitigate the straggler effect, we propose a flexible user aggregation strategy that trains fast users first to obtain a certain level of accuracy before allowing slow users to join the global training updates. Extensive numerical results using several real-world FL tasks are provided to verify the theoretical convergence of `FedFog`. We also show that the proposed co-design of FL and communication is essential to substantially improve resource utilization while achieving comparable accuracy of the learning model.

Index Terms—Distributed learning, edge intelligence, fog computing, federated learning, hierarchical fog/cloud, inner approximation, resource allocation.

I. INTRODUCTION

Nowadays, Internet-connected devices are often equipped with advanced sensors that allow them to collect and store large amounts of data locally. This combined with the high computing capability of edge devices promotes the fog-cloud computing paradigm which brings data processing, storage, and intelligent control to the proximity of the network edge [1]. Besides, emerging technologies (e.g., autonomous driving, industry automation) are relying heavily on data-driven machine learning (ML) approaches to enable near real-time applications [2]. However, traditional ML models, which require that all local data is sent to a centralized server for model training, may not be practical due to high *round-trip* delays, energy constraints and privacy-sensitive concerns of edge devices. Fortunately, distributed ML is practically suited for the fog-cloud computing, which aims at leveraging the advantages of the increasing storage and computing capabilities of edge devices to train ML models while keeping device

datasets local.

Federated learning (FL) is an emerging distributed ML framework that can address many challenges in implementing ML over networks [3], [4]. The most widely used and effective FL algorithm is Federated Averaging (FedAvg) [5]. The FL optimization is commonly solved by an iterative procedure, where each iteration includes local training update and global aggregation. In particular, in each global round, edge devices compute local updates based on their available datasets, typically using gradient descent methods, which are sent back to the sever for global aggregation. Then, the server updates the new global model and broadcasts it to all devices to start the next global round of training. The main advantage of FedAvg over traditional distributed ML algorithms is that each device runs a series of local updates before communicating with the server. This process results in less global updates and reduced communication costs [5]–[7].

A. Related Works

In this section, we focus on the literature review of FL algorithm and wireless FL performance optimization.

FL and challenges. Inspired by FedAvg, FL has attracted considerable attention in recent years in the ML community (see a comprehensive survey in [8]). Several works have attempted to address the main challenges of FL (e.g., non independently and identically distribute (non-i.i.d.) data among devices and resource constraints) for improving communication-efficiency [9]–[11], incentive mechanism [12], privacy-preserving [13], [14], and guaranteeing fairness [15] and robustness [16]. However, these works mainly focused on characterizing and optimizing the FL performance on over-simplified and/or unrealistic communication models, and the impact of wireless factors on FL is often not taken into account.

Wireless FL. Recently, there has been an increasing effort in designing communications protocols and computational aspects of the FL implementation in wireless networks. To improve the convergence speed of FL algorithms, scheduling policies were proposed in [17], where only a portion of users (UEs) are scheduled for updates at each global round. The authors in [18] investigated the impact of various quantization and transmission methods for wireless FL. Various join FL model and radio resource allocation schemes have been proposed in [19]–[21] to minimize either the global loss function or the training time. While many works have focused on efficient wireless communications between the server and UEs to support FL [22]–[24], it is still challenging to employ them in distributed environments due to the causal setting (i.e., the loss value and associated costs of future rounds are not available in advance). Notably, Mahmoudi *et*

V.-D. Nguyen, S. Chatzinotas, and B. Ottersten are with the Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, L-1855 Luxembourg City, Luxembourg (e-mail: {dinh.nguyen, symeon.chatzinotas, bjorn.ottersten}@uni.lu).

T. Q. Duong is with the School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, Belfast BT7 1NN, United Kingdom (e-mail: trung.q.duong@qub.ac.uk).

al. [25] developed an iterative distributed algorithm which characterizes the end-to-end delay as the per-iteration cost. Liu *et al.* [26] proposed a client-edge-cloud hierarchical FL algorithm, called HierFAVG, which allows performing partial model aggregation at edge servers to improve communication-efficiency. However, the realistic cost function taking into account completion time was not considered in these works, and also, computation and wireless factors (i.e., the transmit power, UEs' CPU clock speed and bandwidth coefficients) were not jointly optimized.

Hierarchical FL-supported fog-cloud networks. Despite its potential, there have been only a few attempts to improve the resource utilization of FL-supported fog-cloud networks in the literature. Specifically, the authors in [27] recently introduced a new architecture, called Fog Learning (FogL), which leverages the multi-layer network structure of fog computing to handle ML tasks. This work was extended in [28] to develop a multi-stage hybrid model training, which incorporates multi-stage parameter relaying across networks layers. The work in [29] proposed FogFL to reduce energy consumption and communication latency, where fog nodes act as local aggregators to share location-based information for applications with the similar environment. By taking into account both the computational and communication costs, it was shown in [30] that the network-aware optimization greatly reduces the cost of model training. Very recently, the authors in [31] and [32] studied a general cost optimization of energy consumption and delay minimization within one global iteration. However, the model training was not jointly optimized in the global cost minimization. Wen *et al.* [33] proposed a joint design of the scheduling and resource allocation scheme in an hierarchical federated edge learning, allowing a subset of helpers to upload their updated gradients in each round of the model training. A privacy-preserving FL in fog computing was also proposed in [34], requiring each device to meet different privacy to resist data attacks. Here our focus is the effects of communication and edge devices' computation capability in realistic causal settings.

Straggler effect in FL is a major bottleneck in implementing FL over wireless networks, i.e., when a user has poor channel quality and significantly low computation capability, resulting in higher training time. The promising approaches to mitigating the straggler effect include user sampling [5], [23], [35] and user selection [36]–[38], which require only a subset of users to participate in the training process. However, to the best of our knowledge, these works neither consider a co-design between model training and communication nor simultaneously minimize associated training costs.

B. Research Gap and Main Contributions

Despite the potential benefits offered by FL, there are still several inherent challenges in implementing hierarchical FL over wireless fog-cloud networks, including but not limited to high communication costs, heterogeneity of edge devices (both datasets and computational capabilities), limited wireless resources and straggler effects. Though in-depth results of optimizing communication for FL were presented in [17]–[24], [31], [32], [37], they are not very practical for the actual implementation because the unique characteristics of the federated environment have not been fully addressed in these works. Moreover, it is often considered that the communication and model training are optimized separately [20], [22], [23], [26], [31], [32], [37]. Joint learning and communication [19] is done in the sense that all local losses are available at the server in advance, which violates the FL principle. We show in this paper that communication and FL model training should be optimized on different time scales in each global round. In [19], [20], [22]–[24], the upper bound on the convergence of FL algorithms is characterized by providing the trade-off between the convergence rate and number of global aggregations, which is commonly known in ML literature, but not taking into the cost of model training. Although the higher the number of global aggregations, the lower the training loss that can be obtained, the associated cost increases significantly. This phenomenon promotes a co-design of hierarchical FL and communication that strikes a good balance between the accuracy of the learning model and the running cost. In particular, the co-design should provide an adequate number of global rounds with minimal completion time while still guaranteeing the comparable accuracy of the FL model.

In this paper, we propose a novel network-aware optimization framework to enable hierarchical FL over a cloud-fog system, taking into account all the issues mentioned previously. In the considered system, the cloud server (CS) and fog servers (FSs) do not have access to the UEs' local datasets, thus preserving data privacy. The main goal is to minimize the global loss function and completion time in a single framework, two prime objectives in FL algorithms, which are conflicting. A direct application of an offline algorithm to solve such a problem is inapplicable as it requires complete information at the beginning of the training process, which is impractical in federated settings. Towards a realistic causal setting, we decompose the network-aware optimization problem into two sub-problems, namely the hierarchical FL and resource allocation, which are executed in different time slots. The FedFog is capable of performing a flexible user aggregation which allows fewer UEs to participate in the training process in each round, resulting in low completion time. The convergence of FedAvg with partial user participation has been studied in [35]. Our main contributions are summarized as follows:

> We first propose an efficient FL algorithm for a fog-cloud system (called FedFog) based on FedAvg framework [5]. Acting as a participant, each FS plays the role of a local aggregator that collects the local gradient parameters trained at UEs and then forwards them to CS for global training update, reducing network traffic of backhaul links between FSs and CS. Compared to [26], we explicitly provide the new convergence upper bound of FedFog with a learning rate decay, taking into account non-i.i.d. data and stochastic noise of the random sampling of mini-batchs.
>
> We formulate a novel network-aware FL problem for wireless fog-cloud systems by jointly optimizing the

transmit power, UEs' CPU clock speed and bandwidth coefficients whose goal is to minimize the global loss and completion time in a single framework while meeting UEs' energy constraints. To solve the resource allocation sub-problem of of join computation and communication, we develop a simple yet efficient path-following procedure based on inner approximation (IA) framework [39], in which newly convex approximated functions are derived to tackle nonconvex constraints.

We characterize the discrete convex property of the general cost function to design a stopping criteria to produce a desirable number of global rounds without an additional cost. We then propose the network-aware optimization algorithm that solves the FL and communication problems in a distributed fashion.

To further mitigate the straggler effect, we relax the objective function to favor strong UEs. Then, we determine a time threshold that allows collecting only local gradient parameters of strong UEs. Once a certain accuracy level is reached, the time threshold is increased to allow slower UEs to join the training process, thereby reducing completion time without compromising the learning accuracy. We empirically evaluate the performance of the proposed algorithms using real datasets. The results show that the FedFog-based algorithms can improve network resource utilization while achieving good performance in terms of convergence rate and accuracy of the learning model.

Paper Organization and Mathematical Notations: The rest of this paper is organized as follows. Preliminaries and definitions are described in Section II. The proposed FedFog and the expected convergence rate are given in Section III. The network-aware optimization algorithms are presented in Section IV. Numerical results are analyzed in Section V, while conclusions are draw in Section VI. The main notations and symbols are summarized in Table I.

II. PRELIMINARIES AND DEFINITIONS
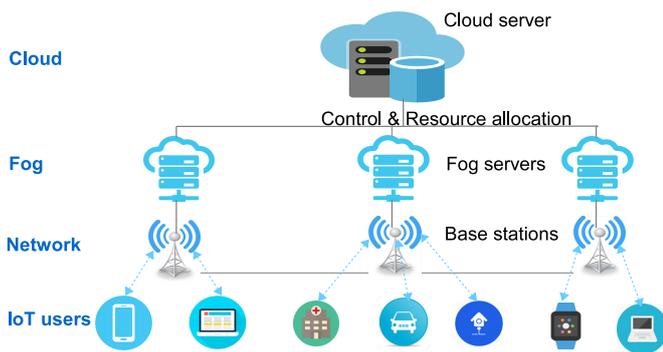### A. Wireless Fog-Cloud Computing Model

Fig. 1: A generic architecture of wireless fog-cloud systems.

A generic fog-cloud computing architecture consists of three layers [34], [40], as illustrated in Fig. 1.

Cloud layer contains large-scale cloud data centers (CDCs) equipped with powerful processing units, providing off-premise computing services to IoT users (or UEs for short). In the context of FL, CS is mainly responsible

TABLE I: Summary of Main Notations and Symbols

$F(\mathbf{w})$ & $F_{ij}(\mathbf{w})$	Global loss function and local loss function of UE (i,j)
$\mathbf{w}^g$ & $\mathbf{w}^g_{ij,\ell}$	Global model at round g and local model of UE (i,j) at global round g and local iteration ℓ
D_{ij} & D_{ij}	Set of data samples and number of samples of UE (i,j)
I & J	Sets of fog nodes and UEs, respectively
J_i & J_i	Set and number of UEs at fog node i
G & L	Sets of global rounds and local iterations
$S(g)$ & $S(g)$	Set and number of UEs selected at round g
g & ℓ	Indexes of global round and local iteration
$B^g_{ij,\ell}$	Mini-batch with size B
η^g	Step size (learning rate) at global round g
$\mathbf{w}^*$	Optimal global model
$p_{ij}(g)$	Transmit power coefficient of UE (i,j)
$f_{ij}(g)$	The CPU clock speed of UE (i,j) at round g
$\beta_{ij}(g)$	UL bandwidth fraction allocated to UE (i,j) at round g
$\langle \mathbf{x}, \mathbf{y} \rangle$	Inner product of vectors $\mathbf{x}$ and $\mathbf{y}$
$\|\mathbf{x}\|$ & $\|z\|$	Euclidean norm of vector $\mathbf{x}$ and absolute value of z, respectively
$\mathbf{h}^H$	Hermitian transpose of vector $\mathbf{h}$
$\nabla F(\cdot)$	Gradient of the function $F(\cdot)$
$\mathbb{E}\{\cdot\}$	Expectation of a random variable
$\mathcal{CN}(0, \sigma^2)$	Circularly-symmetric complex Gaussian random variable with zero mean and variance σ^2
$\mathbb{C}$ & $\mathbb{R}$	Sets of complex and real numbers, respectively

to collect local models generated by IoT UEs to produce a new global model, which is then sent back to them through FSs to start a new training round.

Fog layer comprises a set of FSs deployed close to IoT UEs that can perform local data processing services. Each FS acts as a local aggregator to exchange the model between CS and UEs, which is connected to a base station (BS) via the wired backhaul link, while BS communicates to its UEs through wireless links.

User layer deploys a large number of IoT UEs collected measurement data from the environment that is used to train ML algorithms. Since IoT UEs are placed in the vicinity of their FSs, allowing each UE to transmit the trained parameters to its FS for local aggregations that can obtain low-latency high quality-of-experience (QoE) of IoT UEs.

We consider a fog-cloud network consisting of a set $I \triangleq \{1, 2, \ldots, I\}$ of I fog servers (each fog server already associated with one BS) and a set $J \triangleq \{1, 2, \ldots, J\}$ of J UEs. We assume that FS $i \in I$ serves a separate set $J_i \triangleq \{1, 2, \ldots, J_i\}$ of J_i UEs with $J = \sum_{i \in I} J_i$, and each UE is associated with one FS only. In addition, *one learning process* (i.e., the entire implementation of an FedFog algorithm until convergence) requires G global rounds, each with the same number of local updates, L. Let us denote $G \triangleq \{0, 1, \ldots, G-1\}$ and $L \triangleq \{0, 1, \ldots, L-1\}$ as the sets of G global rounds and L local iterations, respectively. The number of global and local iterations may depend on the specific ML application.

We denote user $j \in J_i$ associated with FS $i \in I$ by UE (i,j), which collects a local input data set D_{ij} of $D_{ij} = |D_{ij}|$ data samples. Considering non-i.i.d. distributed data across the network, we assume that $D_{ij} \cap D_{i'j'} = \emptyset, \forall (i,j) \neq (i',j')$. Each element $\mathbf{x}_d \in D_{ij}$ is an input sample vector with $q > 1$ features. In a typical learning algorithm, we use $(\mathbf{x}_d, y_d)$ to

express the data sample d, where $y_d \in \mathbb{R}$ is the output (label) for the sample $\mathbf{x}_d$.

Remark 1. *In general, it is possible for each UE to forward its local model to more than one FSs to reduce the UL communication delay. However, this will likely lead to a case where some UEs may not be willing to share their proprietary information (i.e., computational capability and battery level) to strange FSs that will be used to optimize system performance. In the proposed scheme, FS i can be seen as a trusted fog server to a group of J_i UEs based on their prior agreement, which further alleviates privacy concerns of data sharing. In many FL applications (e.g., healthcare industry, FinTech, insurance sector and IoT), FSs are often deployed by private organizations to keep their privacy and data preserved.*

B. Federated Learning Model

Definition 1. *Throughout the paper, the model produced by FSs and UEs is referred to as "local aggregation model" and "local model," respectively, while that averaged at CS is called "global model."*

Local loss function: Following the commonly used FL framework [4], [5], the main goal is to jointly learn the *global model parameter* $\mathbf{w} \in \mathbb{R}^q$ (e.g., a neural network or support vector machine [23], [41]) that produces the output y_d given the input sample $\mathbf{x}_d$ through the local loss function $f(\mathbf{w}; \mathbf{x}_d; y_d)$ [4]. On the local data set, the loss function at UE (i, j) can be generally defined as

$$F_{ij}(\mathbf{w}|D_{ij}) \triangleq \frac{1}{D_{ij}} \sum_{d \in D_{ij}} f(\mathbf{w}; \mathbf{x}_d; y_d). \tag{1}$$

The federated learning problem: Since the overall data distributions on UEs are unknown, we consider the empirical loss function across the entire network data set $D = \bigcup_{i,j} D_{ij}$, defined as

$$F(\mathbf{w}|D) \triangleq \frac{\sum_{i \in I} \sum_{j \in J_i} F_{ij}(\mathbf{w}|D_{ij})}{J}. \tag{2}$$

The aim of the common FL algorithms is to find the optimal model $\mathbf{w}$ that minimizes the global loss value of the following optimization problem:

$$\mathbf{w} = \arg \min_{\mathbf{w} \in \mathbb{R}^q} F(\mathbf{w}|D). \tag{3}$$

To achieve this in a distributed fashion, problem (3) is separately decomposed into J independent sub-problems that can be solved locally at UEs. Here, we directly adopt FedAvg [5] which iteratively minimizes the local loss (2) using gradient descent technique before communicating with CS for global update. The key steps of FedAvg training procedure used for the considered fog-cloud system can be summarized as follows:

1) **Global model downloading**: At the start of global round g, CS broadcasts the latest global model $\mathbf{w}^g$ to all FSs, and then FS i broadcasts $\mathbf{w}^g$ to J_i UEs, $\forall i \in I$.
2) **Local training update**: UE (i, j) sets $\mathbf{w}^g_{ij;0} := \mathbf{w}^g$ and then updates the local parameters for $L > 1$ local iterations as

$$\mathbf{w}^g_{ij;\ell+1} := \mathbf{w}^g_{ij;\ell} - \eta^g \nabla F_{ij}(\mathbf{w}^g_{ij;\ell}|D_{ij}), \quad \ell = 0, 1, \dots, L-1 \tag{4}$$

where $\eta^g > 0$ is the step size (learning rate), which is often decreased over time.

3) **Local model uploading**: UE (i, j) with $j \in J_i$ sends $\mathbf{w}^g_{ij;L}$ back to FS i, $\forall i \in I$, and then FS i forwards $\mathbf{w}^g_{ij;L}$ to CS for averaging.
4) **Global training update at CS**: CS performs global training update as

$$\mathbf{w}^{g+1} := \frac{\sum_{i \in I} \sum_{j \in J_i} \mathbf{w}^g_{ij;L}}{J}. \tag{5}$$

5) Set $g := g + 1$ and repeat Steps 1-4 until convergence.

Compared to prior works [19]–[24], one of the key challenges here is how FSs can efficiently convey trained local models to CS while ensuring convergence. We note that forwarding all local models from FSs to CS (e.g., [22], [37] in the context of cell-free networks) will increase the backhaul overhead traffic , which becomes prohibitive in a large-scale system. The proposed approach for local aggregations will be discussed in the sequel.

III. FedFog: Proposed Federated Learning Algorithm Design

We first develop the FedFog algorithm (Section III-A) and then provide a detailed convergence analysis (Section III-B).
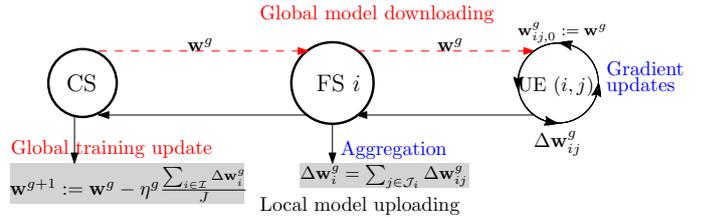
A. Proposed FedFog Algorithm Design

Fig. 2: One FedFog update between global rounds g and $g+1$.

Similar to the extension of FedAvg to the fog-cloud system presented in Section II-B, the training procedure of the proposed FedFog algorithm is detailed in Fig. 2. With the latest global model $\mathbf{w}^g$ at round g, UE (i, j) computes L gradient updates on the local data. FSs aggregate the received local gradient parameters, and then convey them to CS to update the new global model $\mathbf{w}^{g+1}$, which is then sent back to UEs through FSs to begin a new global round. The dimension of learning parameter vectors is the same for every layer.

Local gradient update: In FedFog, UEs will send their gradient parameters to FSs and CS, instead of the trained local models in (4). When the amount of (local) training data is large, it is often impractical for IoT UEs to compute local updates with the full batch gradient using deterministic gradient descent (DGD). Unlike DGD in [26], we use stochastic gradient descent (SGD) method to compute the gradient on mini-batches. Let $B^g_{ij;\ell}$ be the mini-batch with size $B = |B^g_{ij;\ell}|$, randomly sampled from D_{ij} of UE (i, j) at the ℓ-th local iteration of round g. The local parameter estimates in (4) for UE (i, j) is revised accordingly as

$$\mathbf{w}^g_{ij;\ell+1} := \mathbf{w}^g_{ij;\ell} - \eta^g \nabla F_{ij}(\mathbf{w}^g_{ij;\ell}|B^g_{ij;\ell}), \quad \ell = 0, 1, \dots, L-1 \tag{6}$$

Algorithm 1 FedFog: Proposed Federated Learning for Fog-Cloud Systems

---

1: **Input:** L, G, I, J_i, and D_{ij}, $\forall i, j$
2: **Initial parameters at CS:** Initialize the global model $\mathbf{w}^0$ and learning rate η^0
3: **for** $g = 0, 1, \dots, G - 1$ **do**
4:    CS broadcasts $\mathbf{w}^g$ to all FSs
5:    **for** $i \in I$ *in parallel* **do**
6:       FS i broadcasts $\mathbf{w}^g$ to J_i UEs
7:       **for** $j \in J_i$ *in parallel* **do**
8:          Overwrite $\mathbf{w}^g_{ij,0} := \mathbf{w}^g$
9:          **for** $\ell = 0, 1, \dots, L - 1$ **do**
10:            UE (i, j) randomly samples a new mini-batch $B^g_{ij,\ell}$ with size B and computes the gradient $\nabla F_{ij}(\mathbf{w}^g_{ij,\ell}, B^g_{ij,\ell})$
11:          **end for**
12:          UE (i, j) sends $\Delta \mathbf{w}^g_{ij} = \sum_{\ell \in L} \nabla F_{ij}(\mathbf{w}^g_{ij,\ell}, B^g_{ij,\ell})$ to FS i
13:       **end for**
14:       FS i aggregates all gradient parameters $\Delta \mathbf{w}^g_i := \sum_{j \in J_i} \Delta \mathbf{w}^g_{ij}$ and then forwards it to CS for averaging
15:    **end for**
16:    CS performs global training update $\mathbf{w}^{g+1} := \mathbf{w}^g - \eta^g \frac{\sum_{i \in I} \Delta \mathbf{w}^g_i}{J}$
17: **end for**
18: **Output:** Final global model $\mathbf{w}^G$

---

where the stochastic gradient is computed by

$$\nabla F_{ij}(\mathbf{w}^g_{ij,\ell}, B^g_{ij,\ell}) := \frac{1}{B} \sum_{d \in B^g_{ij,\ell}} \nabla f(\mathbf{w}^g_{ij,\ell}; \mathbf{x}_d, y_d). \quad (7)$$

For unbiased estimate of gradient, the condition $E\{\nabla F_{ij}(\mathbf{w}^g_{ij,\ell}, B^g_{ij,\ell})\} = \nabla F_{ij}(\mathbf{w}^g_{ij,\ell}, D_{ij})$ should be satisfied. In particular, the mini-batch for each UE changes for every local iteration but its size is fixed during the whole training process. The total stochastic gradient updates of UE (i, j) at round g is

$$\Delta \mathbf{w}^g_{ij} := \sum_{\ell \in L} \nabla F_{ij}(\mathbf{w}^g_{ij,\ell}, B^g_{ij,\ell}) \quad (8)$$

which also implies that $\mathbf{w}^g_{ij,L} - \mathbf{w}^g_{ij,0} = -\eta^g \Delta \mathbf{w}^g_{ij}$.

Local aggregation and global update: After L local updates, FS i will periodically aggregate gradient parameters as:

$$\Delta \mathbf{w}^g_i := \sum_{j \in J_i} \sum_{\ell \in L} \nabla F_{ij}(\mathbf{w}^g_{ij,\ell}, B^g_{ij,\ell}) = \sum_{j \in J_i} \Delta \mathbf{w}^g_{ij}. \quad (9)$$

All FSs then send their aggregated gradient parameters to CS for averaging (i.e., global training update), as commonly done:

$$\mathbf{w}^{g+1} := \mathbf{w}^g - \eta^g \frac{\sum_{i \in I} \sum_{j \in J_i} \Delta \mathbf{w}^g_{ij}}{J} = \mathbf{w}^g - \eta^g \frac{\sum_{i \in I} \Delta \mathbf{w}^g_i}{J} \quad (10)$$

Once $\mathbf{w}^{g+1}$ is calculated, it will be sent back to UEs to start a new global round. The proposed FedFog algorithm is summarized in Algorithm 1. We have the following remarks:

- Similar to the literature on FL, our method does not require UEs to transfer their raw data to FSs and CS, improving privacy of training data and eliminating communication overhead.
- In the proposed FedFog, the main challenge is to compute the global training update at CS, as given in (10). In FL with one server and multiple UEs [19], [20], [23], local models will be uploaded directly from UEs to the server. This, however, is prohibitive in fog-cloud systems, where CS is often located away from end UEs. The reasons

are two-fold: *i*) It may require extremely high energy consumption of UEs to transmit their local models via wireless links, even not possible to reach the main server due to UEs' limited-battery; 2) It may also cause heavy-communication burdens and high-latency communications due to a very large number of uploading parameters and long-distance transmission. In addition, if each FS naively forwards all received gradient parameters to CS then it will induce very high network traffic, especially with a large number of UEs. In this regard, the local aggregation at FSs given in (9) will result in a learning parameter vector with the same dimension for every layer while still guaranteeing the theoretical performance.

In general, a very large value of L will cause the local models to converge only to an optimal solution of their local loss functions [10], while a very small value of L will result in high communication costs. In this paper, an appropriate value of L will be numerically evaluated, and it is assumed to be predefined in our design.

The optimal value G of global rounds is generally unknown in federated settings. In Section IV, we will consider a stopping criteria based on gradient parameters received at CS which will help not only reduce completion time but also save UEs' energy consumption.

B. Convergence Analysis

To facilitate the analysis, we make the following common assumptions and additional definition to the loss function.

Assumption 1. *For $F_{ij}(\mathbf{w})$, $\forall i \in I$ and $j \in J_i$, we assume that: i) $F_{ij}(\mathbf{w})$ is μ-strongly convex, i.e. $F_{ij}(\mathbf{w}) \geq F_{ij}(\bar{\mathbf{w}}) + \langle \nabla F_{ij}(\bar{\mathbf{w}}), \mathbf{w} - \bar{\mathbf{w}} \rangle + \frac{1}{2}\mu \|\mathbf{w} - \bar{\mathbf{w}}\|^2, \forall \mathbf{w}, \bar{\mathbf{w}}; and ii) $F_{ij}(\mathbf{w})$ is β-smooth (β-Lipschitz gradient), i.e. $\|\nabla F_{ij}(\mathbf{w}) - \nabla F_{ij}(\bar{\mathbf{w}})\| \leq \beta \|\mathbf{w} - \bar{\mathbf{w}}\|, \forall \mathbf{w}, \bar{\mathbf{w}}.$*

Assumption 2 (Bounded variance)**.** *For any $\ell \in L$ and $g \in G$, the variance of the stochastic gradients at UE (i, j) is bounded as:* $E\left\| \nabla F_{ij}(\mathbf{w}^g_{ij,\ell}, B^g_{ij,\ell}) - \nabla F_{ij}(\mathbf{w}^g_{ij,\ell}, D_{ij}) \right\|^2 \leq \sigma^2_{ij}, \forall i \in I, j \in J_i.$

Assumption 3. *For any $\ell \in L$ and $g \in G$, let ς be an upper bound of the expected squared norm of stochastic gradients, i.e.* $E\left\| \nabla F_{ij}(\mathbf{w}^g_{ij,\ell}, B^g_{ij,\ell}) \right\|^2 \leq \varsigma^2, \forall i \in I, j \in J_i.$

Assumption 1 is standard (see [9], [11], [19], [20], [35]) used for the squared-SVM, logistic regression, and softmax classifier, while Assumptions 2 and 3 have been used in [9], [35], [42] to quantify the sampling noise.

Definition 2. *Let Assumption 1 hold. We quantify the heterogeneity of the data distribution between UE (i, j) and other UEs by defining $\varepsilon_{ij} \triangleq F_{ij}(\mathbf{w}^*, D_{ij}) - F^*_{ij}$, where F^*_{ij} is the minimum local loss of UE (i, j). It is clear that ε_{ij} is finite for strongly convex loss function, and $\varepsilon_{ij} = 0$ if data distribution of clients are i.i.d.*

We first introduce additional notation and then provide key lemmas to support the proof of convergence. Motivated by [35], [42], let $\bar{\mathbf{w}}^g_{\ell+1}$ be the average of one local update from all UEs, i.e., $\bar{\mathbf{w}}^g_{\ell+1} \triangleq \frac{1}{J} \sum_{i \in I} \sum_{j \in J_i} \mathbf{w}^g_{ij,\ell} - \eta^g \nabla F_{ij}(\mathbf{w}^g_{ij,\ell}, B^g_{ij,\ell})$. For $\nabla F(\bar{\mathbf{w}}^g) \triangleq \frac{1}{J} \sum_{i \in I} \sum_{j \in J_i}$

access (FDMA) for UL wireless links from UEs to BSs, we denote the UL bandwidth allocated to UE (i,j) at round g by $\beta_{ij}(g)W^{ul}$, satisfying $\sum_{i\in I}\sum_{j\in J_i}\beta_{ij}(g) \leq 1, \forall g$. Under FDMA, the optimal receiver (i.e., maximum ratio combining, $\mathbf{h}_k^H = k\mathbf{h}_k k$) is used for the UL transmission. Let S_{dl} and S_{ul} denote the data size (in bits) of the updated parameters in DL and UL, respectively. In this paper, we have $\mathsf{S}_{dl} < \mathsf{S}_{ul}$ since UEs are required to additionally send the local loss value to CS to design a stopping criteria.

Each BS $i \in I$ broadcasts the latest global model to all J_i UEs using physical-layer multicasting, where the broadcasting transmission rate of each BS is simply determined by its slowest user [46]. The DL achievable rate (a lower bound on the ergodic rate) in bits/s of UE (i,j) to download the latest global model from BS i at round g can be computed as[1]:

$$r_{ij}^{dl}(g) = W_i^{dl} \log\left(1 + \min_{j\in J_i}\mathsf{SNR}_{ij}^{dl}(g)\right) \quad (16)$$

where the signal-to-noise ratio (SNR) is $\mathsf{SNR}_{ij}^{dl}(g) = \frac{P_i^{\max}\mathbb{E}\{k\mathbf{h}_{ij}^{dl}(g)k^2\}g}{W^{dl}N_0} = \frac{P_i^{\max}K_i\gamma_{ij}(g)}{W^{dl}N_0}$, and $P_i^{\max}$ denotes the maximum transmit power at BS i. Here, we consider a worst-case SNR against the noise power, i.e., $W_i^{dl}N_0 \approx W^{dl}N_0$. The DL communication delay of UE (i,j) at round g is

$$t_{ij}^{dl,co}(g) = \frac{\mathsf{S}_{dl}}{r_{ij}^{dl}(g)} \quad (17)$$

which is the same for all UE (i,j) associated with BS i. Similarly, the UL communication delay for UE (i,j) to transmit the trained parameters to BS i using FDMA is

$$t_{ij}^{ul,co}(g) = \frac{\mathsf{S}_{ul}}{r_{ij}^{ul}(g)},$$

with $r_{ij}^{ul}(g) = \beta_{ij}(g)W^{ul}\log\left(1 + \mathsf{SNR}_{ij}^{ul}(g)\right)$ (18)

where $\mathsf{SNR}_{ij}^{ul}(g) = \frac{p_{ij}(g)K_i\gamma_{ij}(g)}{W^{ul}N_0}$, and $p_{ij}(g)$ is the transmit power coefficient of UE (i,j) during the local model uploading phase, subject to the power constraint $p_{ij}(g) \leq P_{ij}^{\max}$.

2) Computation and energy consumption models at UEs: Denote by c_{ij} the number of CPU cycles required for executing 1 data bit of UE (i,j), which assumes to be known *a prior* by an offline measurement [47]. Let $\mathsf{f}_{ij}(g)$ be the CPU clock speed of UE (i,j), which can be chosen in the range $[\mathsf{f}_{ij}^{\min},\mathsf{f}_{ij}^{\max}]$. Then, the computation delay for local training updates at UE (i,j) over L local iterations at round g can be expressed as

$$t_{ij}^{cp}(g) = L\frac{c_{ij}\mathsf{S}_B}{\mathsf{f}_{ij}(g)} \quad (19)$$

where S_B denotes the mini-batch size.

The total energy consumed by UE (i,j) at round g can be formulated as:

$$E_{ij}(g) = \underbrace{p_{ij}(g)t_{ij}^{ul,co}(g)}_{E_{ij}^{co}(g)} + \underbrace{L\frac{\alpha_{ij}}{2}c_{ij}\mathsf{S}_B\mathsf{f}_{ij}^2(g)}_{E_{ij}^{cp}(g)} \quad (20)$$

where $E_{ij}^{co}(g)$ is the energy consumption required to transmit the trained local model via the UL, and $E_{ij}^{cp}(g)$ is the energy consumed for local executions; The constant $\alpha_{ij}=2$ represents the average switched capacitance and the average activity

[1]A more sophisticated beamformer is beyond the scope of this paper.

factor of UE (i,j) [48], [49].

B. Network-Aware Optimization Problem

We assume that UEs communicate asynchronously with BSs [23], [24]. The delay of one global round (say, round g) of FedFog is

$$T(g) = \max_{\forall i,j}\left\{t_{ij}^{dl,co}(g) + t_{ij}^{cp}(g) + t_{ij}^{ul,co}(g)\right\}. \quad (21)$$

The completion time for implementing FedFog over G global rounds is thus $T = \sum_{g\in G}T(g)$. The simplest way to compute T is to set a sufficiently large value of G [19], [23], [24], which ensures the convergence of FedFog. However, such a solution may require redundant transmissions of training models between UEs and CS, resulting in an extra cost. Therefore, it is necessary to design a stopping criteria to output an optimal value $G^* \leq G$, which helps to achieve a lower communication cost $T^* = \sum_{g=0}^{G^*-1}T(g)$. Inspired by [25], we solve (3) by FedFog, taking into account the iterative costs. [25].

Cost function and performance measure: The general cost function should capture both the global loss function and the completion time, which are two prime objectives in FL algorithms, to provide a more precise measurement on the system performance. We first introduce a cost function over G global rounds based on the multi-objective optimization method [50]:

$$C(G) \triangleq \lambda\frac{F(\mathbf{w}^G)}{F_0} + (1-\lambda)\frac{\sum_{g\in G}T(g)}{T_0} \quad (22)$$

where $\lambda \in [0,1]$ is the priority parameter. In addition, $F_0 > 0$ and $T_0 > 0$ denote the references of the loss value and the completion time, respectively, which are used to deal with the different dimensions of the two quantities. We can see that minimizing both the global loss function and the completion time are conflicting. In particular, the lower the completion time, the higher the global loss, resulting in low accuracy of the learning model. Therefore, the priority parameter λ is imposed to create a trade-off between the two objective functions. The higher the value of λ, the higher the completion time that the FedFog is willing to spend to achieve a better accuracy of the learning model.

Remark 2. *We note that minimizing the cost function $C(G)$ requires complete information about the network across G global rounds, which is obviously unaffordable in hierarchical FL-supported wireless fog-cloud networks. In particular, to compute the cost function $C(G)$ in (22) over G global rounds in an offline manner, we need to have the sequences $\{F(\mathbf{w}^g)\}_{\forall g}$ and $\{T(g)\}_{\forall g}$ in advance, which is unrealistic since the future values at round $g+1$ (i.e., $F(\mathbf{w}^{g+1})$ and $T(g+1)$) are not revealed at the beginning of round g. This calls for an alternating procedure, solely based on the network information in each global round. The training time will be accumulated after each global round.*

Based on the above discussions, we consider the following minimization problem of joint learning and communication at round g:

$$\min_{\mathbf{w}^g,\mathbf{p}(g),\mathbf{f}(g),\beta(g)} C(g) \triangleq \lambda\frac{F(\mathbf{w}^g)}{F_0} + (1-\lambda)\frac{\sum_{g'=0}^{g}T(g')}{T_0} \quad (23a)$$

s.t.:
$$E_{ij}(g) \le E^{\max}, \forall i \in I, j \in J_i; \quad (23b)$$
$$SNR_{ij}^{ul}(g) \ge SNR^{\min}, \forall i \in I, j \in J_i; \quad (23c)$$
$$p_{ij}(g) \le P_{ij}^{\max}, \forall i \in I, j \in J_i; \quad (23d)$$
$$f_{ij}^{\min} \le f_{ij}(g) \le f_{ij}^{\max}, \forall i \in I, j \in J_i; \quad (23e)$$
$$\sum_{i\in I}\sum_{j\in J_i} \beta_{ij}(g) \le 1 \quad (23f)$$

where $\mathbf{p}(g) \triangleq \{p_{ij}(g)\}_{\forall i,j}$, $\mathbf{f}(g) \triangleq \{f_{ij}(g)\}_{\forall i,j}$ and $\boldsymbol{\beta}(g) \triangleq \{\beta_{ij}(g)\}_{\forall i,j}$. Constraints (23b) and (23c) indicate the maximum energy consumption requirement $E_{\max}$ and the minimum SNR requirement $SNR^{\min}$ for performing one round of FedFog, respectively. As discussed previously, (23d), (23e) and (23f) are the transmit power, CPU-frequency and bandwidth constraints for UE (i,j), respectively.

We can see that it is not possible to minimize the two quantities in (23a) simultaneously since they are optimized on different time slots. Intuitively, problem (23) can be decomposed into two sub-problems at round g as follows:
$$\min_{\mathbf{w}^g \in \mathbb{R}^q} F(\mathbf{w}|D) \quad (24)$$
which is the FL problem solved by Algorithm 1, solely based on the UEs' local datasets, and
$$\min_{\mathbf{p}(g),\mathbf{f}(g),\boldsymbol{\beta}(g)} C(g) \text{ s.t.: } (23b)-(23f) \quad (25)$$
which is the resource allocation sub-problem of joint computation and communication resources solved at CS for a given $F(\mathbf{w}^g)$ obtained in the last global round. We note that minimizing the cost function $C(g)$ in (25) is equivalent to minimizing the delay of one global round, i.e. $T(g)$, since the total delay of the previous round is already revealed at round g.

C. Proposed Path-Following Algorithm to Solve (25)

In what follows, we treat the loss function $F(\mathbf{w}^g)$ as a constant and rewrite (23a) equivalently as
$$C(g) = (1-\lambda)\frac{T(g)}{T_0} + \bar{C}(g) \quad (26)$$
where $\bar{C}(g) \triangleq \lambda\frac{F(\mathbf{w}^g)}{F_0} + (1-\lambda)\frac{\sum_{g'=0}^{g-1} T(g')}{T_0}$ is also a constant at round g since $\sum_{g'=0}^{g-1} T(g')$ is already computed in the previous rounds.

Problem (25) is nonconvex due to the non-concavity of (23a) and non-convexity of (23b). By introducing new variables $t(g)$ and $\gamma(g) \triangleq \{\gamma_{ij}(g)\}_{\forall i,j}$, problem (25) is rewritten as
$$\min_{\substack{\mathbf{p}(g),\mathbf{f}(g),\boldsymbol{\beta}(g) \\ t(g),\gamma(g)}} C(g) \triangleq (1-\lambda)\frac{t(g)}{T_0} + \bar{C}(g) \quad (27a)$$
s.t.:
$$t_{ij}^{dl,co}(g) + L\frac{c_{ij}S_B}{f_{ij}(g)} + \frac{S_{ul}}{\gamma_{ij}(g)} \le t(g), \forall i,j; \quad (27b)$$
$$r_{ij}^{ul}(g) \ge \gamma_{ij}(g), \forall i \in I, j \in J_i; \quad (27c)$$
$$S_{ul}\frac{p_{ij}(g)}{\gamma_{ij}(g)} + L\frac{\kappa_{ij}}{2}c_{ij}S_B f_{ij}^2(g) \le E^{\max}, \forall i,j; \quad (27d)$$
$$p_{ij}(g) \ge SNR^{\min}\frac{W^{ul}N_0}{K_i \kappa_{ij}(g)}, \forall i,j; \quad (27e)$$
$$(23d),(23e),(23f) \quad (27f)$$
where (27d) and (27e) are transformed from (23b) and (23c), respectively. The equivalence between (25) and (27) is due to the fact that constraints (27b) and (27c) must hold with

equality at optimum for at least some of the slowest UEs. In problem (27), the nonconvex parts include (27c) and (27d), which can be convexified by IA framework [39].

Let us treat (27c) first. We make the variable change $\tilde{\gamma}_{ij}(g) = \frac{1}{\gamma_{ij}(g)} \ge 1, \forall i,j$ to equivalently rewrite (27c) as
$$(27c) \Leftrightarrow \frac{1}{\tilde{\gamma}_{ij}(g)}\log\left(1 + \frac{1}{\psi_{ij}(g)}\right) \ge \frac{\gamma_{ij}(g)}{W^{ul}} \quad (28a)$$
$$\ge \frac{p_{ij}(g)K_i \kappa_{ij}(g)}{W^{ul}N_0} \ge \frac{1}{\psi_{ij}(g)} \quad (28b)$$
where $\tilde{\gamma}(g) \triangleq \{\tilde{\gamma}_{ij}(g)\}_{\forall i,j}$ and $\psi(g) \triangleq \{\psi_{ij}(g)\}_{\forall i,j}$ are newly introduced variables. Constraint (28b) is convex and can be cast into a second-order cone (SOC) one. In (28a), the function $\frac{1}{\tilde{\gamma}_{ij}(g)}\log\left(1 + \frac{1}{\psi_{ij}(g)}\right)$ is convex which can be verified by checking the Hessian matrix. Applying the inequality [51, Appendix A], we iteratively convexify constraint (28a) at iteration $\kappa + 1$ as
$$R_{ij}^{ul,(\kappa)}(\tilde{\gamma}_{ij}(g),\psi_{ij}(g)) \triangleq a_{ij}^{(\kappa)} - b_{ij}^{(\kappa)}\psi_{ij}(g) - c_{ij}^{(\kappa)}\tilde{\gamma}_{ij}(g)$$
$$\ge \frac{\gamma_{ij}(g)}{W^{ul}}, \forall i \in I, j \in J_i \quad (29)$$
where $a_{ij}^{(\kappa)} \triangleq 2\frac{1}{\tilde{\gamma}_{ij}^{(\kappa)}(g)}\log\left(1 + \frac{1}{\psi_{ij}^{(\kappa)}(g)}\right) + \frac{1}{\tilde{\gamma}_{ij}^{(\kappa)}(g)(\psi_{ij}^{(\kappa)}(g)+1)}$, $b_{ij}^{(\kappa)} \triangleq \frac{1}{\tilde{\gamma}_{ij}^{(\kappa)}(g)\psi_{ij}^{(\kappa)}(g)(\psi_{ij}^{(\kappa)}(g)+1)}$ and $c_{ij}^{(\kappa)} \triangleq \frac{1}{(\tilde{\gamma}_{ij}^{(\kappa)}(g))^2}\log\left(1 + \frac{1}{\psi_{ij}^{(\kappa)}(g)}\right)$ are positive constants. Here $\tilde{\gamma}_{ij}^{(\kappa)}(g)$ and $\psi_{ij}^{(\kappa)}(g)$ are the feasible points of $\tilde{\gamma}_{ij}(g)$ and $\psi_{ij}(g)$ obtained at iteration κ, respectively. It is clear that the function $R_{ij}^{ul,(\kappa)}(\tilde{\gamma}_{ij}(g),\psi_{ij}(g))$ is concave lower bound of $\frac{1}{\tilde{\gamma}_{ij}(g)}\log\left(1 + \frac{1}{\psi_{ij}(g)}\right)$. Next, applying [51, Eq. (B.1)] to $\frac{p_{ij}(g)}{\gamma_{ij}(g)}$ in (27d) yields
$$\frac{S_{ul}}{2}\left(\frac{1}{\gamma_{ij}^{(\kappa)}(g)p_{ij}^{(\kappa)}(g)}p_{ij}^2(g) + \frac{p_{ij}^{(\kappa)}(g)}{\gamma_{ij}(g)\gamma_{ij}^{(\kappa)}(g)}\right)$$
$$+ L\frac{\kappa_{ij}}{2}c_{ij}S_B f_{ij}^2(g) \le E^{\max}, \forall i \in I, j \in J_i \quad (30)$$
which is the convex constraint.

Bearing all the above in mind, we solve the following inner convex approximate program at iteration $\kappa + 1$:
$$\min_{\substack{\mathbf{p}(g),\mathbf{f}(g),\tilde{\gamma}(g) \\ t(g),\boldsymbol{\beta}(g),\psi(g)}} C(g) \triangleq (1-\lambda)\frac{t(g)}{T_0} + \bar{C}(g) \quad (31a)$$
s.t.:
$$t_{ij}^{dl,co}(g) + L\frac{c_{ij}S_B}{f_{ij}(g)} + \frac{S_{ul}}{\gamma_{ij}(g)} \le t(g), \forall i,j; \quad (31b)$$
$$\left\|\begin{array}{c} 0.5p_{ij}(g) + \psi_{ij}(g) \\ \sqrt{\frac{W^{ul}N_0}{K_i\kappa_{ij}(g)}} \end{array}\right\| \le 0.5(p_{ij}(g) - \psi_{ij}(g)), \forall i,j; \quad (31c)$$
$$\sum_{i\in I}\sum_{j\in J_i}\frac{1}{\tilde{\gamma}_{ij}(g)} \le 1; \quad (31d)$$
$$(23d),(23e),(27e),(29),(30) \quad (31e)$$
where the SOC constraint (31c) is derived from (28b). We successively solve (31) and update the optimization variables $(\mathbf{p}^{(\kappa)}(g),\tilde{\gamma}^{(\kappa)}(g),\gamma^{(\kappa)}(g),\psi^{(\kappa)}(g))$ until convergence. The proposed path-following procedure to solve (25) is summarized in Algorithm 2. The initial feasible values for

---

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

Initialization: Set $\kappa := 0$ and choose initial feasible values for $(\mathbf{p}^{(0)}(g), \boldsymbol{\beta}^{(0)}(g), \boldsymbol{\tau}^{(0)}(g), \boldsymbol{\omega}^{(0)}(g))$ to constraints in (31)

1: **repeat**
2:     Solve (31) to obtain the optimal solutions $\mathbf{p}^*(g), \mathbf{f}^*(g), \boldsymbol{\beta}^*(g), \mathbf{t}^*(g), \boldsymbol{\tau}^*(g), \boldsymbol{\omega}^*(g)$
3:     Update $\mathbf{p}^{(\kappa+1)}(g), \boldsymbol{\beta}^{(\kappa+1)}(g), \boldsymbol{\tau}^{(\kappa+1)}(g), \boldsymbol{\omega}^{(\kappa+1)}(g) := \mathbf{p}^*(g), \boldsymbol{\beta}^*(g), \boldsymbol{\tau}^*(g), \boldsymbol{\omega}^*(g)$
4:     Set $\kappa := \kappa + 1$
5: **until** Convergence
6: **Output:** The optimal solutions $\mathbf{p}^*(g), \mathbf{f}^*(g), \boldsymbol{\beta}^*(g)$ where $\alpha_{ij}(g) = 1 = \tilde{\tau}_{ij}(g), \forall i, j$

$(\mathbf{p}^{(0)}(g), \tilde{\tau}^{(0)}(g), \gamma^{(0)}(g), \mathbf{l}^{(0)}(g))$ are required for starting the IA procedure. We first randomly generate $p_{ij}^{(0)}(g) \geq \text{SNR}^{\min} \frac{W^{\mathrm{ul}} N_0}{K_i \gamma_{ij}(g)}, P_{ij}^{\max}, \forall i, j$ and then set $\tilde{\tau}_{ij}^{(0)}(g) = J, \gamma_{ij}^{(0)}(g) = \frac{1}{J} W^{\mathrm{ul}} \log\left(1 + \frac{p_{ij}^{(0)}(g) K_i \gamma_{ij}(g)}{W^{\mathrm{ul}} N_0}\right), l_{ij}^{(0)}(g) = \frac{W^{\mathrm{ul}} N_0}{p_{ij}^{(0)}(g) K_i \gamma_{ij}(g)}, \forall i, j.$

Convergence and complexity analysis: The path-following Algorithm 2 is based on the IA framework [39], where all approximate functions in (31) are satisfied IA properties in [52]. In particular, Algorithm 2 produces better solutions after each iteration, which converge to at least a local optimal solution when $\kappa \to \infty$, satisfying the Karush-Kuhn-Tucker (KKT) conditions [39, Theorem 1]. Problem (31) includes $7J + 1$ linear and conic constraints and $5J + 1$ scalar decision variables. By a general interior-point method [53, Chapter 6], the worst-case of per-iteration complexity of Algorithm 2 is $O\left(\sqrt{7J}(5J)^3\right)$.

D. Proposed Network-Aware Optimization Algorithms

We note that (31) is a discrete convex program of (25) in each global round $g \in G$. Towards a practical application, we use the theoretical results above to develop the network-aware optimization algorithms in distributed environments due to the causal setting.

Assumption 4. *In addition to Assumption 1, we further assume that the local loss function $F_{ij}(\mathbf{w})$, $\forall i \in I$ and $j \in J_i$ is μ-strongly convex and non-increasing.*

Proposition 1. *Let Assumption 4 hold. Since the cost function $C(g)$ is a discrete convex function, there always exists $G^* > 0$ as a minimizer of the problem: $G^* := \arg\min_{g \in G} C(g)$, where $C(G^* - 1) \geq C(G^*)$ and $C(G^*) \leq C(G^* + 1)$.*

From Assumption 4, we can show that the global loss function $F(\mathbf{w}^g)$ is non-increasing while the completion time function $\sum_{g \in G} T(g)$ is non-decreasing over time. It implies that we can stop FedFog at round G^* once the stopping condition $C(G^*) - C(G^* - 1) > 0$ is met, without incurring in extra costs. The optimal solution G^* can be found by tracking the sign of two consecutive values of the cost function $C(g)$. However, the non-increasing sequence of the global loss function may not hold true in all global rounds due to non-i.i.d. data and stochastic noise of the random sampling of mini-batches. In this case, a few more rounds are needed to avoid

an improper early convergence of FedFog. This phenomenon will be empirically justified by numerical results.

1) Full User Aggregation

Algorithm 3 Proposed FedFog-based Network-Aware Optimization Algorithm with Full User Aggregation

---

1: **Input:** $L, G, I, J_i, D_{ij}, \bar{k}, B, E^{\max}, \epsilon, \text{SNR}^{\min}, P_{ij}^{\max}, f_{ij}^{\min},$ and $f_{ij}^{\max}, \forall i, j$
2: **Initial parameters at CS:** Initialize $\mathbf{w}^0, \delta^0$, and set $G^* = G, g = 0$
3: **while** $g \leq G - 1$ **do**
4:     Run Algorithm 2 and then broadcast the optimal solutions to UEs using dedicated control channel //(S1)
5:     CS broadcasts $\mathbf{w}^g$ to all FSs //(S2-1)
6:     **for** $i \in I$ in parallel **do**
7:       FS i broadcasts $\mathbf{w}^g$ to J_i UEs //(S2-2)
8:       **for** $j \in J_i$ in parallel **do**
9:         Overwrite $\mathbf{w}_{ij,0}^g := \mathbf{w}^g$
10:         **for** $\ell = 0, 1, \cdots, L - 1$ **do**
11:           UE (i, j) randomly samples a new mini-batch $B_{ij,\ell}^g$ with size B, and computes $\nabla F_{ij}(\mathbf{w}_{ij,\ell}^g, B_{ij,\ell}^g)$ and $F_{ij}(\mathbf{w}^g)$ //(S3)
12:         **end for**
13:         UE (i, j) sends $\mathbf{w}_{ij}^g, \sum_{\ell \leq L} \nabla F_{ij}(\mathbf{w}_{ij,\ell}^g, B_{ij,\ell}^g)$ and $F_{ij}(\mathbf{w}^g)$ to FS i //(S4-1)
14:       **end for**
15:       FS i calculates $\mathbf{w}_i^g := \sum_{j \in J_i} \mathbf{w}_{ij}^g$ and $F_i(\mathbf{w}^g) := \sum_{j \in J_i} F_{ij}(\mathbf{w}^g)$, and then forwards them to CS //(S4-2 & S4-3)
16:     **end for**
17:     CS performs global training update $\mathbf{w}^{g+1} := \mathbf{w}^g - \eta_g \frac{\sum_{i \in I} \mathbf{w}_i^g}{J}$, and calculates the cost function $C(g) = \frac{\sum_{i \in I} F_i(\mathbf{w}^g)}{J F_0} + (1 - \delta) \frac{\sum_{g'=0}^g T(g')}{T_0}$ //(S5)
18:     **if** $C(g) \geq C(g - 1)$ **then**
19:       **if** $(k \geq \bar{k} \ \&\& \ g \geq \bar{G})$ **then**
20:         Set $G^* = g - k$; Break and go to step 28
21:       **end if**
22:       Set $k := k + 1$
23:     **else**
24:       $k \leftarrow 0$
25:     **end if**
26:     Set $g := g + 1$
27: **end while**
28: **Output:** $\mathbf{w}^*, G^*, F(\mathbf{w}^{G^*})$ and $T^* = \sum_{g=0}^{G^*+\bar{k}+1} T(g)$

The complete algorithm with the full user aggregation is summarized in Algorithm 3, where ϵ in Step 18 is a small positive constant. In Step 19, the condition $g \geq \bar{G}$ is added to guarantee a comparable accuracy of the learning model, where $\bar{G}$ is the required minimum number of global rounds. The actual value of $\bar{G}$ may depend on the specific ML applications, FL algorithms and datasets. The large variance of the global loss value, which is due to non-i.i.d. data and stochastic noise of the random sampling of mini-batchs, may lead to an improper early stop in Step 18. To tackle this issue, CS may wait for some more global rounds to ensure the convergence of FedFog. If the condition $C(g) - C(g - 1) \geq \epsilon$ is met for $\bar{k} > 0$ consecutive rounds, we terminate Algorithm 3. We can see that to calculate the last cost value, an additional round of global and local updates is carried out at the end. As a result, the effective completion time for implementing FedFog in Algorithm 3 is given as $T^* = \sum_{g=0}^{G^*+\bar{k}+1} T(g)$.

2) Flexible User Aggregation

We can see that in Algorithm 3, CS needs to wait for the slowest UEs (i.e., due to low computing capability, low battery level and unfavorable links) to perform the global training update in each round, which may result in higher training delay (so-called "straggler effect"). As shown in [35], each UE is only required to activate sometime but still guarantees the convergence of FedAvg. Thus, our next endeavor is to propose a flexible user aggregation to reduce completion time. The key idea is to train strong UEs first to obtain a certain accuracy level, and then more UEs will be allowed to join the training process until convergence.

To achieve the above goal, we relax problem (27) as

$$\underset{\substack{\mathbf{p}(g),\mathbf{f}(g),\boldsymbol{\gamma}(g) \\ \mathbf{t}(g),\boldsymbol{\eta}(g)}}{\text{minimize}} \ \hat{C}(g) \triangleq (1-\beta) \frac{\sum_{i\in I}\sum_{j\in J_i} t_{ij}(g)}{J T_0} + \bar{C}(g) \quad (32a)$$

$$\text{s.t.:} \quad t_{ij}^{\text{dl},co}(g) + L \frac{c_{ij} S_B}{f_{ij}(g)} + \frac{S_{\text{ul}}}{\gamma_{ij}(g)} \le t_{ij}(g), \forall i,j, \quad (32b)$$

$$(23d),(23e),(23f),(27c),(27d),(27e) \quad (32c)$$

where $t_{ij}(g)$ is considered as a soft-latency of UE (i,j) and $\mathbf{t}(g) \triangleq \{t_{ij}(g)\}_{\forall i,j}$. For the objective (32a), CS will favor UEs with better conditions by allocating more resources to them, and thus achieving lower latency than other UEs. This problem can be directly solved by Algorithm 2. Let $S(g)$ be the set of $S(g) = |S(g)|$ UEs selected at round g. Given the optimal solution $\{t_{ij}(0)\}_{\forall i,j}$ obtained from solving (32) at the first round, CS determines a time threshold $T(0) := T_{\text{min}}$ to allow the first $S(0) = J_{\text{min}}$ responded UEs (i.e., J_{min} UEs with the lowest delay) to participate in global updates, given as

$$T(0) := T_{\text{min}} = \max_{(i,j)\in S(0)} \{t_{ij}(0)\} \quad (33)$$

where $J_{\text{min}} \in (0, J]$ should be large enough to guarantee the quality of learning. CS then synchronizes T_{min} to all FSs, and any UE (i,j) with higher latency (i.e., $t_{ij}(0) > T_{\text{min}}, \forall i,j$) will be ignored from the local aggregations at FSs. When the certain accuracy level is obtained at round g, i.e.,

$$\frac{1}{S(g)} \sum_{(i,j)\in S(g)} \|\Delta \mathbf{w}_{ij}^g\| < \epsilon \quad (34)$$

we increase the time threshold $T(g)$ by ΔT to allow weaker UEs to join the global update, i.e., $S(g) := S(g-1) \cup \{t_{ij}(g) \le T(g)\}$, where ϵ is a small positive constant. This procedure is repeated untill all UEs are joined the training process. We summarize the complete procedure of flexible user aggregation in Algorithm 4.

V. NUMERICAL RESULTS

In this section, we numerically evaluate our proposal algorithms in several scenarios. We first present the simulation setup in Section V-A and validate the performance of FedFog in Section V-B. The performance comparison of Algorithms 3 and 4 over a wireless fog-cloud network will be provided in Section V-C.

A. Simulation Setup

ML Model and Data Samples: We consider an image classification task using a multinomial logistic regression with a convex loss function. The regularization parameter is fixed to 10^{-4}. We evaluate FedFog by training neural networks on

---

Algorithm 4 Proposed FedFog-based Network-Aware Optimization Algorithm with Flexible User Aggregation

1: **Input:** $L, G, I, J_i, D_{ij}, J_{\text{min}}, k, B, \epsilon, \Delta, T, E^{\text{max}}, \text{SNR}^{\text{min}}, P_{ij}^{\text{max}}, f_{ij}^{\text{min}}$, and $f_{ij}^{\text{max}} \forall i,j$
2: **Initial parameters at CS:** Initialize $\mathbf{w}^0, \eta^0$, and set $\hat{G} = G, g = 0$
3: **while** $g \le G - 1$ **do**
4:      Run Algorithm 2 and then broadcast the optimal solutions to UEs using dedicated control channel //(S1)
5:      **if** $g = 0$ **then**
6:          Calculate $T(0) := T_{\text{min}}$ and $S(0) = \{t_{ij}(0) \le T_{\text{min}}, \forall i,j\}$ in (33); Break and go to step 10
7:      **else if** the condition (34) is met **then**
8:          Update $T(g) := T(g-1) + \Delta T$ and $S(g) := S(g-1) \cup \{t_{ij}(g) \le T(g)g\}$
9:      **end if**
10:      CS broadcasts $\mathbf{w}^g$ to all FSs //(S2-1)
11:      **for** $i \in I$ *in parallel* **do**
12:          FS i broadcasts $\mathbf{w}^g$ to J_i UEs //(S2-2)
13:          **for** $j \in J_i$ *in parallel* **do**
14:              Overwrite $\mathbf{w}_{ij,0}^g := \mathbf{w}^g$
15:              **for** $\ell = 0,1,\dots,L-1$ **do**
16:                  UE (i,j) randomly samples a new mini-batch $B_{ij,\ell}^g$ with size B; and computes $\nabla F_{ij}(\mathbf{w}_{ij,\ell}^g, B_{ij,\ell}^g)$ and $F_{ij}(\mathbf{w}^g)$ //(S3)
17:              **end for**
18:              UE (i,j) sends $\mathbf{w}_{ij}^g \triangleq \sum_{\ell \le L} \nabla F_{ij}(\mathbf{w}_{ij,\ell}^g, B_{ij,\ell}^g)$ and $F_{ij}(\mathbf{w}^g)$ to FS i //(S4-1)
19:          **end for**
20:          FS i calculates $\mathbf{w}_i^g := \sum_{j\in J_i(g)} \mathbf{w}_{ij}^g$ and $F_i(\mathbf{w}^g) := \sum_{j\in J_i(g)} F_{ij}(\mathbf{w}^g)$ where $J_i(g)$ is the subset of UEs with $t_{ij}(g) \le T(g)$, and then forwards them to CS //(S4-2 & S4-3)
21:      **end for**
22:      CS performs global training update $\mathbf{w}^{g+1} := \mathbf{w}^g - \eta^g \frac{\sum_{i\in I} \mathbf{w}_i^g}{S(g)}$; and calculates the cost function $\hat{C}(g) = \beta \frac{\sum_{i\in I} F_i(\mathbf{w}^g)}{S(g) F_0} + (1-\beta) \frac{\sum_{g^0=0}^g T(g^0)}{T_0}$ //(S5)
23:      **if** $(\hat{C}(g) \ge \hat{C}(g-1)$ && $S(g) = J)$ **then**
24:          **if** $(k \ge \bar{k}$ && $g \ge \bar{G})$ **then**
25:              Set $\hat{G} = g - k$; Break and go to step 33
26:          **end if**
27:          Set $k := k + 1$
28:      **else**
29:          $k \leftarrow 0$
30:      **end if**
31:      Set $g := g + 1$
32: **end while**
33: **Output:** $\mathbf{w}, \hat{G}, F(\mathbf{w}^{\hat{G}})$ and $\hat{T} = \sum_{g=0}^{\hat{G}+k+1} T(g)$

MNIST and CIFAR-10 datasets.

MNIST [54] contains 70K images of hand-written digits 0-9 with 60K training samples and 10K testing samples. We train a fully-connected Neural Network (FCNN) with a single hidden layer using ReLU activation and a softmax layer at the end. There are $(784+1) \times 10 = 7,850$ optimized parameters, where the input and output sizes of the NN model are $28 \times 28 = 784$ and 10, respectively. The initial learning rate is set to $\eta^0 = 0.001$, which is decayed after every global round as $\eta^g = \frac{\eta^0}{1.01^g}$.

CIFAR-10 [55] consists of 60K colour images in 10 different classes (e.g., airplanes, cars, birds, etc.) with 50K training images and 10K testing images, where each image in CIFAR-10 is 32×32 colour image. We train a convolutional NN (CNN) which has two 3×3 convolution

TABLE II: Simulation Parameters

Parameter	Value
System bandwidth, $W = W^{\text{dl}} = W^{\text{ul}}$	10 MHz
Noise power spectral density, N_0	-174dBm/Hz
SNR threshold, SNR^{min}	1 dB
Number of antennas at BS i, $K_i, \forall i$	8
Power budget at FSs, $P_i^{\text{max}}, \forall i$	40 dBm
Effective capacitance coefficient, $\alpha_{ij} = 2, \forall i, j$	10^{-28}
Priority parameter, β	0.7
Minimum number of global rounds for MNIST, $\bar{G}$	250
Minimum number of global rounds for CIFAR-10, $\bar{G}$	600
Energy consumption requirement, E^{max}, for MNIST	0.01 Joule
Energy consumption requirement, E^{max}, for CIFAR-10	1 Joule
Ref. loss and completion time, (F_0, T_0), for MNIST	(0.1,100)
Ref. loss and completion time, (F_0, T_0), for CIFAR-10	(1,1000)
Threshold for stopping condition, $\bar{k}$	5

layers followed by 2×2 maxPooling, one fully-connected layer (128 units) using ReLU activation and a softmax at the output layer. The learning rate is set to $\eta^g = \frac{\eta^0}{1.005^g}$ with $\eta^0 = 0.001$.

Data Distribution: Due to the limited number of samples on datasets, we consider 100 UEs concurrently participating in the training process. There are 5 BSs (or FSs), each has 20 UEs. We consider non-i.i.d. distributed data across the network, where each UE has the same number of data samples but contains only one of the ten classes. We generate an initial global model as $\mathbf{w}^0 = \mathbf{0}$.
Simulation Parameters and Benchmark Schemes over Wireless Fog-Cloud Systems:
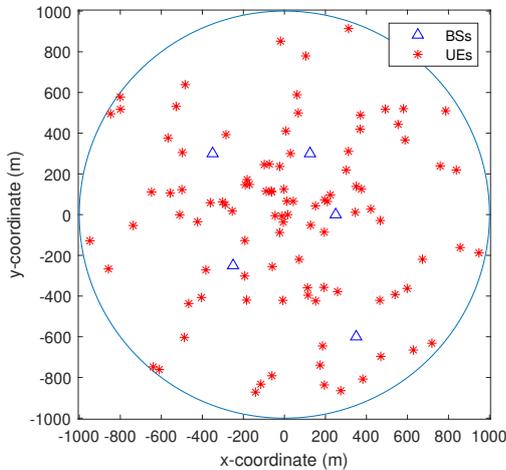
Fig. 4: A system topology with $I = 5$ FSs, $J = 100$ UEs and $J_i = 20$ UEs, $\forall i$.

We consider a system topology shown in Fig. 4, where 5 BSs and 100 UEs are located within a circle of 1-km radius. The locations of BSs are fixed during the simulation. The large-scale fading (in dB) is generated as $\varphi_{ij}(g) = 103.8 + 20.9 \log(d_{ij}(g))$, where $d_{ij}(g)$ (in km) is the distance between BS i and UE (i, j) at round g [23]. By the IEEE 754-2008 standard, we use 32-bit float type to store model weights and the local loss value. To illustrate the heterogeneity of UEs, P_{ij}^{max} is uniformly distributed in $[10, 23]$ dBm, c_{ij} is

uniformly distributed in $[10, 20]$ cycles/bits, f_{ij}^{max} is uniformly distributed in $[10^9, 3 \cdot 10^9]$ cycles/s and $f_{ij}^{\text{min}} = 10^6$ cycles/s. The other parameters are specified in Table II, following [19], [20], [22], [23], [56]. We set $E_{\text{CIFAR-10}}^{\text{max}} > E_{\text{MNIST}}^{\text{max}}$ since the batch size of CIFAR-10 is much larger than that of MNIST. In most cases, Algorithm 2 converges in about 5 iterations. The results are averaged over 100 simulation trials.

For comparison purpose, we consider the following three schemes:

"Equal Bandwidth (EB):" Each UE (i, j) at round g is allocated the fixed portion of bandwidth as $\eta_{ij}(g) = 1/J, \forall i, j$ in uplink.

"Fixed Resource Allocation (FRA):" Since the communication delay is often dominant computation delay, we assume that UE (i, j) uses its maximum transmit power (i.e., $p_{ij}(g) = P_{ij}^{\text{max}}, \forall i, j, g$), and the frequency $f_{ij}(g)$ is then computed by (23b) and (23e).

"Sampling scheme [23], [35]:" At each global round, only a subset $J(g)$ is selected at random to participate in the training process. This scheme allows more bandwidth to be allocated to UEs in the uplink links.

B. Effect of Hyperparameters on FedFog *(Algorithm 1)*

In Fig. 5, we investigate the effect of mini-batch size $B \in \{10, 20, 50\}$ on the performance of FedFog for both MNIST and CIFAR-10 datasets. It can be observed that increasing the size of the mini-batch results in a better convergence rate of FedFog since more data are trained in each iteration. However, a very large mini-batch size (e.g., $B = 50$) slows down the convergence rate of FedFog as it requires more local iterations for the local model training to obtain the same accuracy of the learning model with the medium mini-batch size (e.g., $B = 20$) in each round. In addition, large mini-batch sizes will consume more power and require higher computation at local UEs.

In wireless networks, the communication delay can dominate computation delay, and therefore UEs tends to perform more local updates before sending them to CS, resulting less global model updates. In Fig. 6, we show the convergence rate of FedFog with different values of local iterations L and $B = 20$. In all settings, the larger L has a positive impact of the convergence speed of FedFog; however, a very large number of local iterations also lead to high computation latency and divergent convergence. Hence, it is beneficial to choose the appropriate values of L and B, which not only boosts the convergence speed but also balances trade-off between computations and communications. In the following simulations, we set $B = 20$ and $L = 20$.

C. Numerical Results for FedFog *over Wireless Fog-Cloud Systems (Algorithms 3 and 4)*

Fig. 7 depicts the average $C(G)$ performances of Algorithm 3 with different values of the priority parameter β. As can be seen from this figure that, with small value of β, Algorithm 3 obtains a minimum cost function at small value of G, which may lead to an improper early termination. The reason is that, when β is small, the completion time, which is an increasing function of G, takes more effect on the cost function than the loss function. As expected, a larger value of β provides a

Fig. 5: Effect of mini-batch size B on the convergence of FedFog with L = 10.

Fig. 6: Effect of different numbers of local iterations L on the convergence of FedFog with B = 20.

better balance between the accuracy of the learning model and the running cost. Therefore, we set $= 0.7$ in the following results.

In Fig. 8, we show the performance comparison in terms of completion times among the considered schemes versus the number of global rounds. Clearly, Algorithm 3 outperforms

the baseline schemes in all ranges of G, which is even deeper when G is large. The EB, which fairly allocates the fixed bandwidth to UEs (i.e., $_{ij}(g) = 1 = J; 8i; j; g$), provides the worst performance as the bandwidth allocated to each UE has a great impact on both UL and DL transmission latency, leading to the serious straggler effects. These observations demonstrate the effectiveness of the proposed Algorithm 3 by jointly optimizing the transmit power, CPU-frequency and bandwidth. We can also see that the completion time of CIFAR-10 CNN is much higher than that of MNIST-FCNN since the latter has larger sizes (in bits) of data and model training than the former.

Fig. 7: Average $C(G)$ performances of Algorithm 3 with different values of .

Fig. 9: Trade-off between completion time and maximum energy consumption requirement.

The impact of the maximum energy consumption requirement, E^{max}, on the completion time is plotted in Fig. 9, where we set $k = 5$ for the stopping condition. Increasing the threshold E^{max} results in lower completion times for all schemes. This phenomenon is not surprising because with a larger value of E^{max}, more power and CPU-frequency of UEs can be used for the global training update and local model training subject to constraint (23b). Again, Algorithm 3 still offers the best performance out of the schemes considered.

Fig. 10 characterizes the number of received gradients for flexible user aggregation-based schemes. We note that it is often not beneficial to completely satisfy the condition (34). Therefore, weaker UEs are allowed to participate in global updates earlier after a fixed number of global rounds, say G. Here we set $G = 50$, which is numerically shown to significantly accelerate the convergence rate of FedFog. As can be seen in Fig. 10(a) that increasing G results in higher

Fig. 8: Completion time versus G for different schemes.

(a) Algorithm 4: varying $\bar{T}$

(b) Performance comparison between Algorithm 4 and baseline schemes with $\bar{T} = 0.15s$

Fig. 10: Number of received gradients for flexible user aggregation-based schemes, with $J_{min} = 20$.

Fig. 11: Performance comparison between Algorithm 4 and baseline schemes, with $J_{min} = 20$ and $\bar{T} = 0.15s$.

number of received gradients. However, a large $\bar{T}$ (i.e., $\bar{T} = 0.2s$) will not only bring less benefit in terms of the number of received gradients, but also lead to a higher training time. In addition, the results in Fig. 10(b) show that Algorithm 4 can boost the number of received gradients compared to the baseline schemes with the same completion time. This results in better model training, as demonstrated in Fig. 11.

Lastly, we compare the performance between the proposed algorithms and the sampling scheme in Fig. 12, with $J(g) = 10.8g$ for the latter. As expected, Algorithm 4 requires much less completion time in Fig. 12(a) while still achieving the comparable accuracy of the learning model in Fig. 12(b), compared to Algorithm 3. On the other hand, the sampling scheme has the lowest completion time due to a small number of UEs participated in the training process, but exhibiting a much slower convergence speed than Algorithms 3 and 4. In other words, Algorithm 4 offers a good balance between the quality of learning model and the communication cost.

VI. Conclusion

In this paper, we proposed the network-aware FL algorithm for wireless fog-cloud systems, a novel methodology for optimizing the distribution of ML tasks across users while tackling inherent issues in fog-cloud scenarios. We characterized the overall running cost of implementing the proposed FedFog algorithm in discrete time intervals, taking into account the

Fig. 12: Performance comparison between different schemes, with $J_{min} = 20; J(g) = 10.8g$ and $\bar{T} = 0.15s$.

effects of both computation and communication. The cost function of the formulated problem captures the global loss and overall cost in terms of the training time, which is used to design the iteration-stopping criteria to produce a desirable number of global rounds. The proposed scheme can avoid the redundant cost with negligible negative impact on the convergence rate and accuracy of the learning model. We also developed a flexible user aggregation strategy to mitigate the straggler effect, resulting in less completion time of FedFog over wireless fog-cloud systems. Numerical results with popular ML tasks were provided to validate the effectiveness of the network-aware FL algorithm compared to existing baseline approaches. For future works, it would be interesting to develop and implement prototypes to validate the efficiency of FedFog in real-time environments.

APPENDIX A: PROOF OF LEMMAS 1-3

A. Proof of Lemma 1

Since the variances of the stochastic gradients of any two UEs are independent, it follows that

$$\frac{1}{J^2}\mathbb{E}\left\{\left\|\sum_{i\in I}\sum_{j\in J_i}\nabla F_{ij}(w_{ij;\ell}^g)-\nabla F_{ij}(w_{ij;\ell}^g)\right\|^2\right\}$$

$$=\frac{1}{J^2}\sum_{i\in I}\sum_{j\in J_i}\mathbb{E}\left\{\left\|\nabla F_{ij}(w_{ij;\ell}^g)-\nabla F_{ij}(w_{ij;\ell}^g)\right\|^2\right\}$$

$$+\frac{1}{J^4}\sum_{(i;j)\ne(i^0;j^0)}\mathbb{E}\left\{\left\|\nabla F_{ij}(w_{ij;\ell}^g)-\nabla F_{ij}(w_{ij;\ell}^g)\right.\right.$$

$$\left.\left.\nabla F_{i^0j^0}(w_{i^0j^0;\ell}^g)-\nabla F_{i^0j^0}(w_{i^0j^0;\ell}^g)\right\|^2\right\}$$

$$\frac{1}{J^2}\sum_{i\in I}\sum_{j\in J_i}\sigma_{ij}^2 \quad \text{(by Assumption 2):} \tag{A.1}$$

B. Proof of Lemma 2

From the definition of w_ℓ^g there always exists $\ell^0 \quad \ell$; that satisfies $\ell \quad \ell^0 \quad L \quad 1$. First, it is true that $\|w^g \quad w_{ij;\ell}^g\|^2 = \|(w^g \quad w_{\ell^0}^g) \quad (w_{ij;\ell}^g \quad w_{\ell^0}^g)\|^2 = \|w^g \quad w_{\ell^0}^g\|^2 \quad 2\langle w^g \quad w_{\ell^0}^g; w_{ij;\ell}^g \quad w_{\ell^0}^g\rangle + \|w_{ij;\ell}^g \quad w_{\ell^0}^g\|^2$. In addition, we have $\langle w^g \quad w_{\ell^0}^g; \sum_{i\in I}\sum_{j\in J_i}w_{ij;\ell}^g \quad w_{\ell^0}^g\rangle = \|w^g \quad w_{\ell^0}^g\|^2$. Thus, it follows that

$$\frac{1}{J}\mathbb{E}\left\{\sum_{i\in I}\sum_{j\in J_i}\left\|w^g \quad w_{ij;\ell}^g\right\|^2\right\}$$

$$=\frac{1}{J}\sum_{i\in I}\sum_{j\in J_i}\mathbb{E}\{\|w_{ij;\ell}^g \quad w_{\ell^0}^g\|^2\} \quad \frac{1}{J}\mathbb{E}\{\|w^g \quad w_{\ell^0}^g\|^2\}$$

$$=\frac{\eta_g^2}{J}\sum_{i\in I}\sum_{j\in J_i}\left\|\sum_{t=\ell^0}^{\ell}\nabla F_{ij}(w_{ij;t}^g)\right\|^2$$

$$\frac{\eta_g^2}{J}\sum_{i\in I}\sum_{j\in J_i}\sum_{t=\ell^0}^{\ell}\left\|\nabla F_{ij}(w_{ij;t}^g)\right\|^2$$

$$\frac{\eta_g^2}{J}\sum_{i\in I}\sum_{j\in J_i}\sum_{t=\ell^0}^{\ell}\left\|\nabla F_{ij}(w_{ij;t}^g)\right\|^2$$

$$\frac{\eta_g^2 L}{J}\sum_{i\in I}\sum_{j\in J_i}\sum_{t=\ell^0}^{\ell}\left\|\nabla F_{ij}(w_{ij;t}^g)\right\|^2$$

$$(\ell \quad \ell^0)L\eta_g^2\varrho^2 \quad (L \quad 1)L\eta_g^2\varrho^2$$

$$\text{(due to } \ell \quad \ell^0 \quad L \quad 1): \tag{A.2}$$

C. Proof of Lemma 3

Given $w_{\ell+1}^g = w_\ell^g \quad \eta_g\nabla F(w_\ell^g)$ and $\nabla F(w_\ell^g)$, $\frac{1}{J}\sum_{i\in I}\sum_{j\in J_i}\nabla F_{ij}(w_{ij;\ell}^g) = \mathbb{E}\{\nabla F(w_\ell^g)\}$, it follows that

$$\|w_{\ell+1}^g \quad w_\star\|^2$$

$$\|w_\ell^g \quad w_\star \quad \eta_g\nabla F(w_\ell^g) + \eta_g(\nabla F(w_\ell^g) \quad \nabla F(w_\ell^g))\|^2$$

$$\underbrace{\|w_\ell^g \quad w_\star \quad \eta_g\nabla F(w_\ell^g)\|^2}_{A} + \underbrace{\eta_g^2\|\nabla F(w_\ell^g) \quad \nabla F(w_\ell^g)\|^2}_{B} \tag{A.3}$$

since $\mathbb{E}\{\langle w_\ell^g \quad w_\star \quad \eta_g\nabla F(w_\ell^g); \nabla F(w_\ell^g) \quad \nabla F(w_\ell^g)\rangle\} = 0$.

We first focus on the expected bound of A by rewriting it as:

$$A = \underbrace{\|w_\ell^g \quad w_\star\|^2}_{A_1} \quad \underbrace{2\eta_g\langle w_\ell^g \quad w_\star; \nabla F(w_\ell^g)\rangle}_{A_2} + \underbrace{\eta_g^2\|\nabla F(w_\ell^g)\|^2}_{A_3}: \tag{A.4}$$

For A_2 and from Assumption 1 on μ-strongly convex, we have

$$A_2 = \quad 2\eta_g\frac{1}{J}\sum_{i\in I}\sum_{j\in J_i}\langle w_\ell^g \quad w_\star; \nabla F_{ij}(w_{ij;\ell}^g)\rangle$$

$$= \quad 2\eta_g\frac{1}{J}\sum_{i\in I}\sum_{j\in J_i}\langle w_\ell^g \quad w_{ij;\ell}^g; \nabla F_{ij}(w_{ij;\ell}^g)\rangle$$

$$2\eta_g\frac{1}{J}\sum_{i\in I}\sum_{j\in J_i}\langle w_{ij;\ell}^g \quad w_\star; \nabla F_{ij}(w_{ij;\ell}^g)\rangle$$

$$\frac{1}{J}\sum_{i\in I}\sum_{j\in J_i}\|w_\ell^g \quad w_{ij;\ell}^g\|^2 + \eta_g^2\|\nabla F_{ij}(w_{ij;\ell}^g)\|^2$$

$$2\eta_g F_{ij}(w_{ij;\ell}^g) \quad F_{ij}(w_\star) + \frac{\mu}{2}\|w_{ij;\ell}^g \quad w_\star\|^2: \tag{A.5}$$

For A_3 and from Assumption 1 on L smooth, we have

$$A_3 = \eta_g^2\|\frac{1}{J}\sum_{i\in I}\sum_{j\in J_i}\nabla F_{ij}(w_{ij;\ell}^g)\|^2 \quad \eta_g^2\frac{1}{J}\sum_{i\in I}\sum_{j\in J_i}\|\nabla F_{ij}(w_{ij;\ell}^g)\|^2$$

$$\frac{2L\eta_g^2}{J}\sum_{i\in I}\sum_{j\in J_i}F_{ij}(w_{ij;\ell}^g) \quad F_{ij}: \tag{A.6}$$

Substituting (A.5) and (A.6) into (A.4), it follows that

$$A \quad (1 \quad 0.5\mu\eta_g)\|w_\ell^g \quad w_\star\|^2$$

$$+ (1 + \mu\eta_g)\frac{1}{J}\sum_{i\in I}\sum_{j\in J_i}\|w_\ell^g \quad w_{ij;\ell}^g\|^2 + A_4 \tag{A.7}$$

where we use the fact that $\eta_g\frac{1}{J}\|w_{ij;\ell}^g \quad w_\star\|^2 \quad \eta_g(0.5\|w_\ell^g \quad w_\star\|^2 \quad \sum_{i\in I}\sum_{j\in J_i}\|w_\ell^g \quad w_{ij;\ell}^g\|^2)$, $L\nabla F_{ij}(w_{ij;\ell}^g)\|^2 \quad 2\nabla F_{ij}(w_{ij;\ell}^g) \quad F_{ij}$, and $A_4 \quad 4\eta_g^2\frac{1}{J}\sum_{i\in I}\sum_{j\in J_i}F_{ij}(w_{ij;\ell}^g) \quad F_{ij}$ $2\eta_g\frac{1}{J}\sum_{i\in I}\sum_{j\in J_i}F_{ij}(w_{ij;\ell}^g) \quad F_{ij}(w_\star)$.

To bound A_4, we define $\nu_g = 2\eta_g(1 \quad 2\eta_g)$ and assume $\eta_g \quad \frac{1}{4}$. Thus, we have $\nu_g \in [\eta_g \quad 2\eta_g]$ and hence

$$A_4 = \quad 2\eta_g(1 \quad 2\eta_g)\frac{1}{J}\sum_{i\in I}\sum_{j\in J_i}F_{ij}(w_{ij;\ell}^g) \quad F_{ij}$$

$$+ 2\eta_g\frac{1}{J}\sum_{i\in I}\sum_{j\in J_i}F_{ij}(w_\star) \quad F_{ij}$$

$$= \quad \nu_g\frac{1}{J}\sum_{i\in I}\sum_{j\in J_i}F_{ij}(w_{ij;\ell}^g) \quad F_{ij}(w_\star)$$

$$+ (2\eta_g \quad \nu_g)\frac{1}{J}\sum_{i\in I}\sum_{j\in J_i}F_{ij}(w_\star) \quad F_{ij}$$

$$\frac{1}{J_g}\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}_i}\left(\nabla F_{ij}(w_{ij;\kappa}^g)-\nabla F_{ij}(w^*)\right)$$
$$+4\beta_g^2\frac{1}{J}\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}_i}\sigma_{ij}^2 \quad (\text{by Definition 2}): \qquad (A.8)$$

Further, it is noted that
$$\frac{1}{J}\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}_i}\left(\nabla F_{ij}(w_{ij;\kappa}^g)-\nabla F_{ij}(w^*)\right)=$$
$$\frac{1}{J}\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}_i}\left(\nabla F_{ij}(w_{ij;\kappa}^g)-\nabla F_{ij}(w^g)+\nabla F_{ij}(w^g)-\nabla F_{ij}(w^*)\right):$$
$$(A.9)$$

Applying Assumption 1 and $\|\nabla F_{ij}(w_{ij;\kappa}^g)\|^2\le 2\|\nabla F_{ij}(w_{ij;\kappa}^g)-\nabla F_{ij}\|$ to (A.9), after some manipulations we can obtain
$$A_4\le \beta_g(\beta_g-1)\frac{1}{J}\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}_i}\left(\nabla F_{ij}(w^g)-\nabla F_{ij}(w^*)\right)$$
$$+\frac{1}{J}\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}_i}\|w_{ij;\kappa}^g-w^g\|^2+6\beta_g^2\frac{1}{J}\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}_i}\sigma_{ij}^2: \qquad (A.10)$$

Substituting (A.7) and (A.10) into (A.3), we have
$$\|w_{\kappa+1}^g-w^*\|^2\le(1-0.5\beta_g)\|w_\kappa^g-w^*\|^2$$
$$+\beta_g^2\gamma^g+2\beta_g\frac{1}{J}\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}_i}\left(\nabla F_{ij}(w^*)-\nabla F_{ij}(w^g)\right) \qquad (A.11)$$

due to $\beta_g\le\frac{1}{\mu}=\frac{4}{L}$ and $\beta_g(1-\beta_g\mu)\ge 2\beta_g$, where $\gamma^g\triangleq\frac{(2+\mu=4)}{\mu^2}\frac{1}{J}\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}_i}\|\nabla F_{ij}(w^g)-\nabla F_{ij}(w_{ij;\kappa}^g)\|^2+\|\nabla F(w^g)-\nabla F(w^g)\|^2+6\frac{1}{J}\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}_i}\sigma_{ij}^2:$ By Lemmas 1 and 2, the expectation of γ^g is $\Gamma^g=\mathbb{E}\{\gamma^g\}=(2+\mu=4)(L-1)L\Theta^2+\frac{\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}_i}\sigma_{ij}^2}{J^2}+6\frac{1}{J}\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}_i}\sigma_{ij}^2:$

APPENDIX B: PROOF OF THEOREM 1

Similar to [35], [42], let us define $Q_\kappa^g\triangleq\|w_\kappa^g-w^*\|^2$ and $Q^g=\mathbb{E}\{Q_0^g\}$. From (A.11), we have
$$\sum_{\kappa=1}^{L}Q_\kappa^g\le\sum_{\kappa=0}^{L-1}(1-0.5\beta_g)Q_\kappa^g+\sum_{\kappa=0}^{L-1}\beta_g^2\gamma^g$$
$$+2\beta_g\frac{1}{J}\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}_i}\left(\nabla F_{ij}(w^*)-\nabla F_{ij}(w_0^g)\right) \qquad (B.1)$$

where w_0^g is a minimizer of $\frac{1}{J}\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}_i}F_{ij}(w^g)$, leading to $\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}_i}\mathbb{E}\{\nabla F_{ij}(w^*)-\nabla F_{ij}(w_0^g)\}\ge 0$. By $\beta_g\triangleq\beta$, $\sum_{\kappa=0}^{L-1}\gamma^g$, we rewrite (B.1) as
$$Q^{g+1}\le Q^g-0.5\beta_g\sum_{\kappa=0}^{L-1}Q_\kappa^g+\beta_g^2\gamma^g: \qquad (B.2)$$

In addition, $\|w_{\kappa+1}^g-w^*\|\le\|w_\kappa^g-w^*\|+\beta_g\|\nabla F(w^g)\|$ or $\|w^{g+1}-w^*\|\le\|w_\kappa^g-w^*\|+\beta_g\sum_{\kappa=}^{L-1}\|\nabla F(w_\kappa^g)\|$. As a result, we can show that $Q^g\ge 0.5Q^{g+1}-\beta_g^2(\sum_{\kappa=0}^{L-1}\|\nabla F(w_\kappa^g)\|)^2$. Hence, (B.2) is rewritten as
$$(1+0.25\beta_g L)Q^{g+1}\le Q^g$$
$$+0.5\beta_g^3 L\sum_{\kappa=0}^{L-1}\|\nabla F(w_\kappa^g)\|^2+\beta_g^2\gamma^g: \qquad (B.3)$$

Assuming the learning rate $\beta_g\le 4=L$ and taking the expectation of both sides of (B.3), we have
$$Q^{g+1}\le\left(1-\frac{\beta_g L}{8}\right)Q^g+\beta_g^2 2\gamma^g+\phi^g \qquad (B.4)$$

where $\gamma^g\triangleq\sum_{\kappa=0}^{L-1}\|\nabla F(w_\kappa^g)\|^2\gamma^g$, $L\sum_{\kappa=0}^{L-1}\mathbb{E}\{\frac{1}{J}\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}_i}\|\nabla F_{ij}(w_{ij;\kappa}^g)\|^2\gamma^g\}\le L^2\Theta^2$ by Cauchy–Schwarz inequality and Assumption 3, and $\phi^g=(2+\mu=4)(L-1)L\Theta^2+\frac{\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}_i}\sigma_{ij}^2}{J^2}+6\beta_g\frac{1}{J}\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}_i}\sigma_{ij}^2$.

We consider a diminishing learning rate $\beta_g=\frac{16}{(g+1+\psi)}$ with $\psi=\max\{\frac{64}{\mu};4L\}>0$, satisfying $\beta_1\le\min\{\frac{1}{4};\frac{4}{L}\}$. By defining $\vartheta\triangleq\max\{\mu^2\mathbb{E}\|w^0-w^*\|^2\psi;\frac{16^2}{\mu^2}\gamma\}$ for $\gamma\triangleq 2L^2\Theta^2+\beta(2+\mu=4)(L-1)L\Theta^2+\frac{\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}_i}\sigma_{ij}^2}{J^2}+6\beta L\frac{1}{J}\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}_i}\sigma_{ij}^2$, we next will prove that $Q^g\le\frac{\vartheta}{(g+\psi)^2}$, which is done by induction. It follows that
$$Q^{g+1}\le\left(1-\frac{16}{(g+1+\psi)}\frac{L}{8}\right)\frac{\vartheta}{(g+\psi)^2}$$
$$+\frac{16}{(g+1+\psi)}\Omega^2$$
$$\le\frac{g+\psi-L}{(g+\psi-L)(g+\psi+L)}\frac{\vartheta}{g+1+\psi}$$
$$+\frac{1}{(g+1+\psi)^2}\frac{16^2}{\mu^2}\Omega\quad(\text{due to }1-L\ge 0)$$
$$=\frac{1}{(g+1+\psi)^2}\vartheta+\frac{16^2}{\mu^2}\Omega=\frac{\vartheta^{g+1}}{(g+1+\psi)^2} \qquad (B.5)$$

where $\vartheta^g\triangleq\max\{\mu^2\mathbb{E}\|w^0-w^*\|^2g;\frac{16^2}{\mu^2}(g+1)\}$.

REFERENCES

[1] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 854–864, Dec. 2016.

[2] Cisco, Demystifying 5G in Industrial IoT, White Paper, 2019. [Online]. Available: www.cisco.com/c/dam/en-us/solutions/iot/demystifying-5g-industrial-iot.pdf

[3] J. Park, S. Samarakoon, M. Bennis, and M. Debbah, "Wireless network intelligence at the edge," *Proc. of the IEEE*, vol. 107, no. 11, pp. 2204–2239, Nov. 2019.

[4] J. Park, S. Samarakoon, A. Elgabli, J. Kim, M. Bennis, S. L. Kim, and M. Debbah, "Communication-efficient and distributed learning over wireless networks: Principles and applications," *Proc. of the IEEE*, vol. 109, no. 5, pp. 796–819, May 2021.

[5] 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.

[6] J. Konecny, 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.

[7] J. Wang, S. Wang, R.-R. Chen, and M. Ji, "Local averaging helps: Hierarchical federated learning and convergence analysis," 2020. [Online]. Available: https://arxiv.org/abs/2010.12998

[8] P. Kairouz et al., "Advances and open problems in federated learning," 2019. [Online]. Available: https://arxiv.org/abs/1912.04977

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

[10] 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.

[11] 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, June 2019.

[12] 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, Dec. 2019.

[13] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, "Privacy aware learning," J. ACM, vol. 61, no. 6, pp. 1–57, Dec. 2014. [Online]. Available: https://dl.acm.org/doi/10.1145/2666468

[14] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning differentially private recurrent language models," 2018. [Online]. Available: https://arxiv.org/abs/1710.06963

[15] T. Li, M. Sanjabi, A. Beirami, and V. Smith, "Fair resource allocation in federated learning," in *Proc. International Conference on Learning Representations (ICLR)*, 2020.

[16] C. Xie, S. Koyejo, and I. Gupta, "SLSGD: Secure and efficient distributed on-device machine learning," 2019. [Online]. Available: https://128.84.21.199/abs/1903.06996v1

[17] 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.

[18] S. Zheng, C. Shen, and X. Chen, "Design and analysis of uplink and downlink communications for federated learning," *IEEE J. Select. Areas Commu.*, vol. 39, no. 7, pp. 2150–2167, July 2021.

[19] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 269–283, Jan. 2021.

[20] 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," *IEEE/ACM Trans. Networking*, vol. 29, no. 1, pp. 398–409, Feb. 2020.

[21] 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, Jan. 2020.

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

[23] V.-D. Nguyen, S. K. Sharma, T. X. Vu, S. Chatzinotas, and B. Ottersten, "Efficient federated learning algorithm for resource allocation in wireless IoT networks," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3394–3409, Mar. 2021.

[24] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, "Energy efficient federated learning over wireless communication networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1935–1949, Mar. 2021.

[25] A. Mahmoudi, H. S. Ghadikolaei, and C. Fischione, "Cost-efficient distributed optimization in machine learning over wireless networks," in *Proc. IEEE Inter. Conf. Commun. (ICC)*, Dublin, Ireland, June 2020, pp. 1–7.

[26] L. Liu, J. Zhang, S. Song, and K. B. Letaief, "Client-edge-cloud hierarchical federated learning," in *IEEE Inter. Conf. Commun. (ICC)*, 2020, pp. 1–6.

[27] S. Hosseinalipour, C. G. Brinton, V. Aggarwal, H. Dai, and M. Chiang, "From federated to fog learning: Distributed machine learning over heterogeneous wireless networks," *IEEE Commun. Mag.*, vol. 58, no. 12, pp. 41–47, Dec. 2020.

[28] S. Hosseinalipour, S. S. Azam, C. G. Brinton, N. Michelusi, V. Aggarwal, D. J. Love, and H. Dai, "Multi-stage hybrid federated learning over large-scale D2D-enabled fog networks," 2020. [Online]. Available: https://arxiv.org/abs/2007.09511v4

[29] R. Saha, S. Misra, and P. K. Deb, "FogFL: Fog assisted federated learning for resource-constrained IoT devices," *IEEE Internet of Things Journal*, vol. 8, no. 10, pp. 8456–8463, May 2021.

[30] Y. Tu, Y. Ruan, S. Wagle, C. G. Brinton, and C. Joe-Wong, "Network-aware optimization of distributed learning for fog computing," in *IEEE Conf. Comp. Commun. (IEEE INFOCOM)*, 2020, pp. 2509–2518.

[31] J. Feng, L. Liu, Q. Pei, and K. Li, "Min-max cost optimization for efficient hierarchical federated learning in wireless edge networks," *IEEE Trans. Parallel. Distrib. Syst.*, pp. 1–1, 2021 (Early access).

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

[33] W. Wen, Z. Chen, H. H. Yang, W. Xia, and T. Q. S. Quek, "Joint scheduling and resource allocation for hierarchical federated edge learning," *IEEE Trans. Wirel. Commun.*, pp. 1–1, 2022 (Early access).

[34] C. Zhou, A. Fu, S. Yu, W. Yang, H. Wang, and Y. Zhang, "Privacy-preserving federated learning in fog computing," *IEEE Internet of Things Journal*, vol. 7, no. 11, pp. 10 782–10 793, Nov. 2020.

[35] 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.

[36] M. Chen, H. Vincent Poor, W. Saad, and S. Cui, "Convergence time optimization for federated learning over wireless networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 4, pp. 2457–2471, Apr. 2021.

[37] T. T. Vu, D. T. Ngo, H. Q. Ngo, M. N. Dao, N. H. Tran, and R. H. Middleton, "User selection approaches to mitigate the straggler effect for federated learning on cell-free massive MIMO networks," 2020. [Online]. Available: https://arxiv.org/abs/2009.02031

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

[39] 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.

[40] G. Lee, W. Saad, and M. Bennis, "An online optimization framework for distributed fog network formation with minimal latency," *IEEE Trans. Wireless Commun.*, vol. 18, no. 4, pp. 2244–2258, Apr. 2019.

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

[42] Y. Ruan, X. Zhang, S.-C. Liang, and C. Joe-Wong, "Towards flexible device participation in federated learning for non-iid data," 2020. [Online]. Available: https://arxiv.org/abs/2006.06954

[43] U. Dotsch, M. Doll, H.-P. Mayer, F. Schaich, J. Segel, and P. Sehier, "Optimal and successive approaches to signal design for multiple antenna physical layer multicasting," *Bell Labs Tech. J.*, vol. 18, no. 1, p. 105–128, June 2014.

[44] M. Chen, B. Liang, and M. Dong, "Multi-user multi-task offloading and resource allocation in mobile cloud systems," *IEEE Trans. Wireless Commun.*, vol. 17, no. 10, pp. 6790–6805, Oct. 2018.

[45] J. Du, L. Zhao, J. Feng, and X. Chu, "Computation offloading and resource allocation in mixed fog/cloud computing systems with min-max fairness guarantee," *IEEE Trans. Commun.*, vol. 66, no. 4, pp. 1594–1608, Apr. 2018.

[46] I. H. Kim, D. J. Love, and S. Y. Park, "Optimal and successive approaches to signal design for multiple antenna physical layer multicasting," *IEEE Trans. Commun.*, vol. 59, no. 8, pp. 2316–2327, 2011.

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

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

[49] M. E. T. Gerards, J. L. Hurink, and J. Kuper, "On the interplay between global DVFS and scheduling tasks with precedence constraints," *IEEE Trans. Computers*, vol. 64, no. 6, pp. 1742–1754, June 2015.

[50] R. Marler and J. Arora, "Survey of multi-objective optimization methods for engineering," *Structural Multidisciplinary Optimization*, vol. 26, pp. 369–395, Apr. 2004.

[51] 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.

[52] 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.

[53] A. Ben-Tal and A. Nemirovski, *Lectures on Modern Convex Optimization*. Philadelphia: MPS-SIAM Series on Optimi., SIAM, 2001.

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

[55] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," vol. 1, no. 4, 2009. [Online]. Available: https://www.cs.toronto.edu/ kriz/cifar.html

[56] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3590–3605, Dec. 2016.

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.

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.

Symeon Chatzinotas (SM IEEE) 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.

Trung Q. Duong (Fellow, IEEE) is a Chair Professor of Telecommunications at Queen's University Belfast (UK), where he was a Lecturer (Assistant Professor) (2013-2017), a Reader (Associate Professor) (2018-2020), and Full Professor from August 2020. He also holds a prestigious Research Chair of Royal Academy of Engineering. His current research interests include wireless communications, signal processing, machine learning, realtime optimisation, and data analytic.

Dr. Duong currently serves as an Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE WIRELESS COMMUNICATIONS LETTERS, and an Executive Editor for IEEE COMMUNICATIONS LETTERS. He has served as an Editor/Guest Editor for IEEE TRANSACTIONS ON COMMUNICATIONS, IEEE WIRELESS COMMUNICATIONS, IEEE COMMUNICATIONS MAGAZINES, IEEE COMMUNICATIONS LETTERS, and IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS. He was awarded the Best Paper Award at the IEEE Vehicular Technology Conference (VTC-Spring) in 2013, IEEE International Conference on Communications (ICC) 2014, IEEE Global Communications Conference (GLOBECOM) 2016 and 2019, IEEE Digital Signal Processing Conference (DSP) 2017, and International Wireless Communications & Mobile Computing Conference (IWCMC) 2019. He is the recipient of prestigious Royal Academy of Engineering Research Fellowship (2015-2020) and has won a prestigious Newton Prize 2017. He is a Fellow of IEEE. Foundations and Trends of Signal Processing. He is a fellow of EURASIP.