Wireless Distributed Learning: A New Hybrid Split and Federated Learning Approach

Xiaolan Liu *Member, IEEE,* Yansha Deng *Member, IEEE,* and Toktam Mahmoodi *Senior Member, IEEE,*

Abstract

Cellular-connected unmanned aerial vehicle (UAV) with flexible deployment is foreseen to be a major part of the sixth generation (6G) networks. The UAVs connected to the base station (BS), as aerial users (UEs), could exploit machine learning (ML) algorithms to provide a wide range of advanced applications, like object detection and video tracking. Conventionally, the ML model training is performed at the BS, known as centralized learning (CL), which causes high communication overhead due to the transmission of large datasets, and potential concerns about UE privacy. To address this, distributed learning algorithms, including federated learning (FL) and split learning (SL), were proposed to train the ML models in a distributed manner via only sharing model parameters. FL requires higher computational resource on the UE side than SL, while SL has larger communication overhead when the local dataset is large. To effectively train an ML model considering the diversity of UEs with different computational capabilities and channel conditions, we first propose a novel distributed learning architecture, a hybrid split and federated learning (HSFL) algorithm by reaping the parallel model training mechanism of FL and the model splitting structure of SL. We then provide its convergence analysis under non-independent and identically distributed (non-IID) data with random UE selection scheme. By conducting experiments on training two ML models, Net and AlexNet, in wireless UAV networks, our results demonstrate that the HSFL algorithm achieves higher learning accuracy than FL and less communication overhead than SL under IID and non-IID data, and the learning accuracy of HSFL algorithm increases with the increasing number of the split training UEs. We further propose a Multi-Arm Bandit (MAB) based best channel (BC) and best 2-norm (BN2) (MAB-BC-BN2) UE

This work was submitted and accepted in part at IEEE International Communication Conference (ICC2022).

This work has funded by the European Union's Horizon 2020 research and innovation programme, and 'Primo-5G : virtual presence in moving objects, e.g. drones, through 5G' under grant agreement No. 815191.

Xiaolan Liu is with the Institute for Digital Technologies, Loughborough University, London, E20 3BS, U.K. (Email: xiaolan.liu@lboro.ac.uk) ; Yansha Deng and Toktam Mahmoodi are with the Department of Engineering, King's College London, London, WC2R 2LS, U.K. (Email: { yansha.deng, toktam.mahmoodi }@kcl.ac.uk).

selection scheme to select the UEs with better wireless channel quality and larger local model updates for model training in each round. Numerical results demonstrate it achieves higher learning accuracy than BC, MAB-BC and MAB-BN2 UE selection scheme under non-IID, Dirichlet-nonIID and Dirichlet-Imbalanced data.

Index Terms

Wireless unmanned aerial vehicles (UAV) Networks, Federated learning (FL), Multi-Arm Bandit (MAB), Split learning (SL), User (UE) selection

I. Introduction

Cellular-connected unmanned aerial vehicle (UAV) network is becoming an integral component of the beyond fifth generation (5G) and upcoming sixth generation (6G) networks [1], [2] to provide a variety of advanced applications ranging from real-time video streaming to surveillance. In this new network, the aerial users (UEs), i.e., UAVs, fly over the target area with the control of the base stations (BSs) to collect data (e.g., images and videos), and then they collaborate with the BSs to perform data processing for supporting those applications. Recently, machine learning (ML) algorithms, like deep neural network (DNN) and convolutional neural network (CNN), have been effectively used to provide efficient data processing for those applications through extracting the features and insights from a large dataset. However, each UE is only able to collect a sub-dataset that only contains partial information of the target area. The conventional approach is to gather the sub-datasets from all the UEs to the BSs for centralized ML model training, known as centralized learning (CL). In this case, the UEs require wide bandwidth and large amount of energy to transmit their sub-datasets to the BSs, and may potentially reveal their private information through the transmission process [3]. In practice, the transmission processes from UAVs to the BS always suffer from limited bandwidth and dynamic wireless channels, and the UAVs are powered by energy-limited batteries, hence, transmitting raw data to the BS is challenging. Due to the growing computational capability of computing engines, such as the CPU, GPU and DSP (e.g., Qualcomm Hexagon Vector extensions on Snapdragon 835 [4], and the possibility of equipped GPU on UAVs), the UAVs are able to perform ML model training locally using their own sub-datasets and then only share model parameters instead of raw data with the BSs. Therefore, distributed learning algorithms are emerged to provide ML

model training in a distributed manner, which becomes a more attractive solution for supporting advanced applications in cellular-connected UAV networks.

The two state-of-art distributed learning algorithms, federated learning (FL) and split learning (SL), have different learning architectures and therefore are suitable for different application scenarios. In FL, all the UEs collaboratively train an entire ML learning model (e.g., DNN) with the help of a central parameter server collecting and performing model aggregation with the received local model updates from the UEs [5]. FL architectures rely on the fact that all of the UEs are capable of performing gradient descent and having powerful computational capabilities. Different from FL, SL was recently proposed in [6], [7] by splitting the ML model (e.g., DNN) into several sub-models (e.g., a few layers of the entire DNN) with the cut layer and distributing them to different entities (e.g., the UE-side model at the UEs or the server-side model at the server), which facilitates distributed learning via sharing the smashed data of the cut layer. In this case, SL limits the UE-side model down to a few layers, thus reduces the computational overhead of the UEs compared to FL. Interestingly, the researches in [7], [8] have shown that FL is more communication or computation efficient with small model size and large dataset size, whereas SL is more efficient with increasing the number of UEs or the model size. However, in practical UAV networks, the UAVs have diverse computational capabilities, own different datasets (e.g., imbalanced and non-independent and identically distributed (non-IID) data distribution over them), and heterogeneous communication and energy resources, either deploying FL or SL may be not efficient. Motivated by this, a splitfed learning (SFL) has been proposed in [9], which exploits the parallel model training mechanism in FL and model splitting structure of SL. By doing so, the SFL shortens the training time in SL and becomes more communication efficient than FL when the number of UEs is large. However, the SFL algorithm still exhibits high communication overhead similar to SL when the number of UEs is small and the dataset over UEs is highly imbalanced. To address this, there is an urgent need to propose a hybrid solution that can well leverage the advantages from both FL and SL even for small number of UEs and the highly imbalanced datasets.

While deploying distributed learning algorithms in wireless networks, not all the UEs can access to the BS in each communication round due to unreliable and randomly fading wireless channels from the UEs to the BSs, so it's essential to develop efficient UE selection schemes to select reliable and informative UEs to participate in distributed learning in each round. In FL, UE selection schemes have been widely studied [10]–[15], where the parameter server

determines which UEs should participate in FL according to their channel conditions and resource information (e.g., throughput, computational resource) . Generally, the UE selection in FL has been studied either based on channel qualities [10]–[12] or the importance of local model updates [13]–[15]. In [10], the proportional fair UE selection policies based on the instantaneous channel qualities were developed. In [11], a joint learning, wireless resource allocation and UE selection problem was formulated and optimized to minimize the FL loss function. The authors in [12] studied the UE selection scheme that maximizes the number of selected UEs in each round based on their wireless and computational resource conditions. The authors in [13] proposed a reliable UE selection scheme by considering the reliability of the dataset owned by UEs. The reliability of dataset has a great impact on the importance of local model updates while training a ML model with the dataset, the user selection policy taking into account both channel conditions and the importance of local model updates at the UEs was proposed in [14], [15].

Nevertheless, the above studies [10]–[15] assumed that the UE information, including channel conditions and the importance of local model updates, is known in advance. In practice, it is difficult to obtain accurate UE information before the execution of the learning procedure, it also consumes extra computation and communication resources to estimate each UE's local model updates before UE selection. To address this, the dynamic UE selection scheme for FL based on Multi-Arm Bandit (MAB) has been proposed [16]–[18], in which the parameter server selects the UEs through exploration and exploitation processes according to the estimated local model updates of UEs [16] or the estimated channel qualities of UEs [17], [18]. Moreover, from [10]–[15], when deploying FL in wireless networks, both channel qualities and the importance of local model updates are significant to select UEs for global model aggregation in each round. However, as far as we know, there are rare existing works that have considered both factors together to design UE selection schemes using MAB algorithm. Different from considering any of them, the exploration process of exploiting MAB algorithm needs to maximize a weighted sum of both channel qualities and the importance of local model updates, which causes a challenge of finding a trade-off between those two parameters when selecting UEs.

Motivated by the above, we will study distributed learning architecture to train ML models for supporting advanced applications, like fire tracking and flood monitoring, in wireless UAV networks. We consider a group of aerial UEs are flying over a target area under the control of the BS to collect image data with the equipped camera. Here, each UAV is carried with a powerful processing unit (e.g., NVIDIA JETSON) [19] that may have different computational

capabilities, and it can only capture a sub-dataset that observes partial information of the target area, and thus the whole dataset collected by all the UAVs may be on imbalanced and non-IID distribution. By transmitting the sub-datasets to the BS, an immediate data aggregation can be performed to enable each UE access to the complete environment information captured by other UEs. However, the transmission process of raw data is expensive in terms of energy and bandwidth, and possibly introduces infringements of UE privacy. To address these challenges, we first propose a novel distributed learning architecture, namely the hybrid split and federated learning (HSFL) algorithm, that encompasses the parallel model training mechanism of FL and the model splitting structure of SL. We conduct the experiments on the image recognition task using MINST dataset and perform training on two different ML models, Net and AlextNet, with the goal of improving the learning accuracy and communication efficiency in wireless UAV networks using the proposed distributed learning algorithms.

The main contributions of this work are summarized as follows.

We propose an HSFL algorithm that allows a portion of UEs to train an entire ML model locally, namely federated training, and other portions of UEs to train the ML model collaboratively with the BS, namely split training. Our results show that the learning accuracy results follows: CL$>$ SL $>$ HSFL $>$ FL/SFL, in wireless UAV networks under IID and non-IID data.We perform fundamental analysis on an expression for the convergence rate of HSFL algorithm in wireless UAV networks under non-IID data with random UE selection scheme. Our analytical result reflects that the convergence speed increases with increasing the number of split training UEs. We exploit MAB algorithm to design UE selection scheme based on the discounted upper confidence bound (UCB) policy. Then, an MAB-BC-BN2 UE selection scheme is proposed to select the UEs with better wireless channel quality and larger local model updates to participate in ML model training in each round by designing a trade-off function of both factors as the UCB score. Our results demonstrate that our proposed HSFL algorithm achieves around half less communication overhead and faster convergence compared to SL and SFL. Additionally, the communication efficiency of HSFL is better than that of FL, and improves with the increasing number of UEs. We also show that our proposed MAB-BC-BN2 UE selection scheme achieves better learning accuracy performance than BC, MAB-BC and MAB-BN2 UE selection schemes under non-IID, Dirichlet-nonIID and Dirichlet-Imbalanced data.

The organization of this paper is presented as follows. In Section II, we present the system

model and learning model, as well as the learning problem formulation in wireless networks. Section III introduces the proposed HSFL algorithm including its learning procedure and convergence analysis in wireless networks. Then the UE selection schemes are illustrated in Section IV. The experiments and simulation results are demonstrated in Section V, finally the conclusions are drawn in Section VI.

II. System Model and Problem Formulation

As illustrated in Fig. 1, we consider a single-cell wireless UAV network, consisting of a BS located at the center of the cell, and a set of aerial UEs $\mathcal{N} = \{u_1, ..., u_N\}$ distributed in the BS coverage area as predefined flight paths and remained spatially static during the process of ML model training. In this network, the total system bandwidth is equally divided into M radio access channels, where $M < N$. The BS S is assumed to have a single antenna and equipped with high computational capability, and located at the origin of the 3D coordinates system with the antenna installed at the altitude h_s above the ground. Each UE is also equipped with a single antenna and a lightweight GPU. We assume that all the UEs transmit data with a constant power $P_n = P$. The location of UE u_n is denoted as (x_n, y_n, h_n). Each UE is assumed to fly at the fixed altitude h_n above the ground while the horizontal coordinates (x_n, y_n) of each UE vary over time.

A. Channel Propagation Model

From Fig. 1, in the considered cellular-connected UAV networks, only the information including the BS's and UAVs' locations, and the type of environment (e.g. rural, suburban, urban, highrise urban, etc.) is available. Noted that, in such practical scenarios, one may not have any additional information about the exact locations, heights, and the number of obstacles. Therefore, to consider the possibility of occurrence of LoS link affected by the environment, we adopt the channel model for ait-to-ground (ATG) communication in urban environment presented in [20], [21]. Here, we consider the randomness of LoS communication links using an LoS probability $\mathbb{P}_{n,s}^{LoS}$, which depends on the environment, the location of UE and BS, as well as the elevation angle. Thus, the LoS probability is given as

$$\mathbb{P}_{n,s}^{LoS} = \frac{1}{1 + a \cdot \exp(-b[\theta_{n,s} - a])}, \tag{1}$$

where a and b are the environmental parameters indicating the type of environment, like rural, urban or dense urban, $\theta_{n,s}$ is the elevation angle of the UE-BS communication link. In (1),

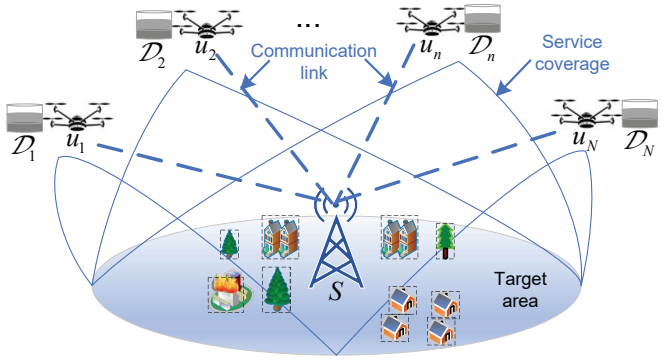


Fig. 1. System model

$\theta_{n,s} = \frac{180}{\pi} \times sin^{-1}(\frac{h_n - h_s}{dist_{n,s}})$, where $dist_{n,s}$ is the Euclidean distance between UE u_n and BS S, calculated by $dist_{n,s} = \sqrt{x_n^2 + y_n^2 + (h_n - h_s)^2}$. The LoS probability increases with increasing the elevation angle and the UE's altitude.

As stated in [22], the communication paths of ATG channels depend on both LoS and NLoS propagations, and it's impossible to determine the exact LoS/ NLoS status of the UE-BS link. Thus, we consider the spatial expectation of the pathloss for LoS and NLoS groups as the pathloss model to describe the UE-BS communication channel, which is given by

$$\bar{\xi}_{ij} = \mathbb{P}_{n,s}^{LoS} \varphi_l (\frac{4\pi f \cdot dist_{n,s}}{c})^{\alpha} + \mathbb{P}_{n,s}^{NLoS} \varphi_n (\frac{4\pi f \cdot dist_{n,s}}{c})^{\alpha}, \tag{2}$$

where $\mathbb{P}_{n,s}^{NLoS} = 1 - \mathbb{P}_{n,s}^{LoS}$ is the NLoS probability, f is the system carrier frequency, c is the light speed, and α denotes the path loss exponent, φ_l and φ_n are the additional path loss coefficients of LoS and NLoS, respectively.

B. Problem Formulation

At the BS, the goal is to learn a statistical model over the dataset distributed among N UEs, that is, the BS aims to obtain an optimal vector $\boldsymbol{\omega}$ to minimize an empirical loss function $L(\boldsymbol{\omega})$ (e.g., $L(\mathbf{x}^T\boldsymbol{\omega}) = \frac{1}{2}\left\|y - \phi(\boldsymbol{\omega}^T\mathbf{x})\right\|$) by using the dataset distributed over all the UEs under its service. The local loss function of the u_n that measures the prediction error of its local dataset $\mathcal{D}_n, d_n = |\mathcal{D}_n|$ denoting the data size, can be defined as

$$L_n(\boldsymbol{\omega}) = \frac{1}{d_n} \sum_{i=1}^{d_n} l(\boldsymbol{\omega}, \mathrm{x}_n^i), \quad \forall n \in \mathcal{N}. \tag{3}$$

where $l(\boldsymbol{\omega}, \mathrm{x}_n^i)$ is an empirical loss function defined by the learning task, which quantifies the loss of the ML model at sample x_n^i.

The objective of the considered learning task is to find the optimal model weights $\boldsymbol{\omega}^*$ that minimize the global loss function $L(\boldsymbol{\omega})$ [23] as

$$\mathcal{OP}_1 \quad \min_{\boldsymbol{\omega} \in R} \; L(\boldsymbol{\omega}). \tag{4}$$

To solve the optimization problem $\mathcal{OP}_1$, two distributed learning approaches, FL and SL, can be used to train the ML model by exploiting the computational capabilities of the UEs in a distributive manner. However, FL has higher requirements on the computational resource at the UEs and SL has higher communication overhead when the dataset is large at UE. To efficiently obtain the solution to $\mathcal{OP}_1$ with the dataset distributed over the heterogeneous UEs, we propose a novel distributed learning architecture, namely the HSFL algorithm, which keeps the parallel model training mechanism of FL and the model splitting structure of SL.

C. The FL and the SL Preliminaries

In this section, we present the learning procedures of using FL or SL to solve the optimization problem $\mathcal{OP}_1$.

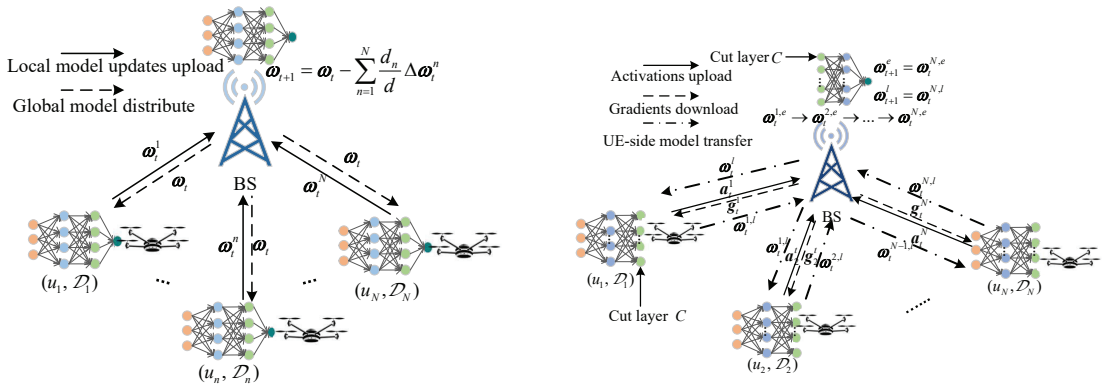
1) FL Algorithm: To solve the optimization problem $\mathcal{OP}_1$ using FL, we can convert $\mathcal{OP}_1$ to the following

$$\mathcal{OP}_2 \quad \min_{\boldsymbol{\omega} \in R} \{L(\boldsymbol{\omega}) = \frac{1}{d} \sum_{n=1}^{N} d_n L_n(\boldsymbol{\omega})\}, \tag{5}$$

where $d = \sum_{n=1}^{N} d_n$ is the size of the whole dataset. By applying the Federated Averaging algorithm proposed in [5] to solve $\mathcal{OP}_2$, the general learning procedure of this algorithm is illustrated in Fig. 2 (a). In Fig. 2 (a), each UE receives a global model, $\boldsymbol{\omega}_t$, from the BS and trains it with the local dataset by minimizing the local loss function (3), it then performs gradient descent, such that the global model $\boldsymbol{\omega}_t$ is updated at UE u_n to $\boldsymbol{\omega}_{t+1}^n$, thus the local model updates can be defined as

$$\Delta \boldsymbol{\omega}_t^n = \boldsymbol{\omega}_{t+1}^n - \boldsymbol{\omega}_t. \tag{6}$$

The BS periodically collects the local model updates from the UEs and then performs model aggregation to generate the improved global model and sends it back to the UEs. The whole process, defined as one communication round, repeats a sufficient amount of rounds until the objective function converges to the global optima.



(a) Federated learning (b) Split learning

Fig. 2. The illustration of FL [5] and SL [7]

2) SL Algorithm: SL algorithm is another state-of-art distributed learning techniques without need of directly accessing the raw data. Unlike FL, where each UE trains the entire ML model, SL divides the ML model into at least two sub-models, and trains them separately at the UE and the BS. As shown in Fig. 2 (b), the SL framework with multiple UEs in centralized mode [6] is illustrated, where each UE holds a fraction of dataset $\mathcal{D}_n$, to participate in training the model aiming to minimize the global loss function $L(\boldsymbol{\omega})$ sequentially. From Fig. 2 (b), the ML model is divided into two sub-models by the cut layer C, the first sub-model is trained at the UEs, termed as UE-side model $\boldsymbol{\omega}_t^l$, whereas the second sub-model is trained at the BS, termed as BS-side model $\boldsymbol{\omega}_t^e$. As such, each UE only needs to train a sub-model, consisting of a few layers and the rest of layers reside at the BS, which can reduce the computational load of each UE.

To solve the optimization problem $\mathcal{OP}_1$ with SL, we can convert it to

$$\mathcal{OP}_3 \quad \min_{\boldsymbol{\omega}\in R} \{L(\boldsymbol{\omega}) = \frac{1}{d}\sum_{n=1}^{N} d_n L_n(\boldsymbol{\omega})\}. \tag{7}$$

where the full model $\boldsymbol{\omega}$ includes two sub-models $\boldsymbol{\omega}_t^l$ and $\boldsymbol{\omega}_t^e$, it can be denoted by

$$\boldsymbol{\omega} = \{\boldsymbol{\omega}_t^l; \boldsymbol{\omega}_t^e\} \tag{8}$$

If we use the classical sequential SL mechanism to solve $\mathcal{OP}_3$, the learning procedure is presented as the following steps: 1) the BS initializes the global BS-side model $\boldsymbol{\omega}_t^e$ and the global UE-side model $\boldsymbol{\omega}_t^l$, then sends $\boldsymbol{\omega}_t^l$ to the UE u_1; 2) the UE u_1 trains $\boldsymbol{\omega}_t^l$ over its local dataset $\mathcal{D}_1$ and then sends the output of the cut layer C, $\boldsymbol{a}_t^1$, to the BS; 3) the BS receives and

feed forwards $\boldsymbol{a}_t^1$ to the BS-side model $\boldsymbol{\omega}_t^e$, and then it calculates and back propagates the loss to the cut layer C, where its gradients, $\boldsymbol{g}_1^t$, are computed; 4) the BS sends $\boldsymbol{g}_1^t$ back to UE u_1 for back propagation and updating the UE-side model $\boldsymbol{\omega}_t^{1,l}$, UE u_1 then updates the UE-side model and sends it back to the BS; and 5) the UE u_2 receives the UE-side model $\boldsymbol{\omega}_t^{1,l}$ from the BS and then starts training on its local dataset. This repeats until the training of the last UE is finished, then one communication round is finished.

D. UE Selection

When applying FL algorithm in wireless networks, the limited bandwidth and dynamic communication channels make the BS unable to access all the UEs in each round. Additionally, different local model updates are of dissimilar importance to the model convergence [15], [24]. Therefore, it's essential to develop efficient UE selection schemes to select a subgroup of UEs that provide the most useful information in each round. Due to the parallel model training mechanism of FL, applying the proposed HSFL algorithm in wireless networks also requires efficient UE selection scheme. Inspired by [24], we propose a MAB-BC-BN2 UE selection scheme by jointly taking into account both channel qualities and the importance of local model updates. For comparisons, we also implement the BC and BN2 UE selection schemes proposed in [24] as the benchmark schemes, and also propose the MAB-BC and MAB-BN2 UE selection schemes.

III. A Novel Distributed Learning Architecture: HSFL Algorithm

In this section, We present our proposed novel distributed learning architecture, an HSFL algorithm, by exploiting the advantageous learning mechanisms of FL and SL. In the following, we first introduce its learning procedure, then propose a wireless HSFL algorithm with its convergence analysis.

A. HSFL Learning Procedure

Inspired by [9], [25], we propose a novel HSFL algorithm with the detailed learning procedure as illustrated in Fig. 3. Let us consider the UE set $\mathcal{U} = \{u_1, ..., u_n, u_{n+1}, ..., u_N\}$ with diverse computational capabilities, channel qualities and energy resource, and the dataset owned by UEs is imbalanced and non-IID. In our proposed HSFL algorithm, we allow a portion of UEs to implement split training method with lower computational capability at the UE side,

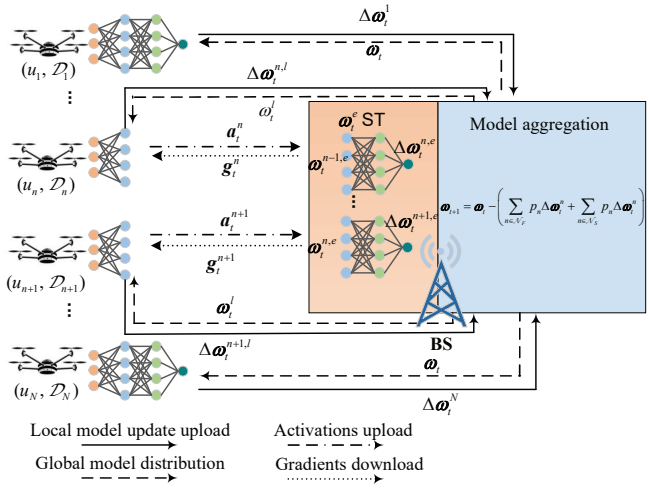


Fig. 3. The illustration of HSFL algorithm

while allowing the rest portion of the selected UEs to use federated training method with less communication overhead when the dataset is large at the UE. Then, the UEs perform local model training in parallel and send the local model updates to the BS where it performs model aggregation and generates new global models.

The detailed steps are given in Fig. 3. Here, if u_1 and u_N are scheduled for federated training, they receive the global model parameters $\boldsymbol{\omega}_t$ of the entire ML model from the BS and perform local training only at the UE sides. On the other hand, if u_n and u_{n+1} are scheduled for split training, they receive the global UE-side model parameters $\boldsymbol{\omega}_t^l$, i.e., a sub-model of the entire ML model from the BS, at the same time, the BS initializes the global BS-side model parameters $\boldsymbol{\omega}_t^e$ (adopts sequential training at the BS). Then, the UEs perform local training in parallel at both UE and BS sides. Noted that the BS undertakes two tasks, including the BS-side model training for the split training UEs and model aggregation for all the UEs. When each UE finishes its local training, it sends the local model updates to the BS where the model aggregation is performed as in FL and the global model $\boldsymbol{\omega}_{t+1}$, the global UE-side model $\boldsymbol{\omega}_{t+1}^l$ and the global BS-side model $\boldsymbol{\omega}_{t+1}^e$ are generated.

Next, the local model updates of the federated training UEs and split training UEs in the HSFL algorithm will be discussed, respectively.

1) Federated Training: The federated training UEs $u_n, n \in \mathcal{N}_{F_t}$ follow the same local model update rule as in (6),

$$\Delta \boldsymbol{\omega}_t^n = \boldsymbol{\omega}_{t+1}^n - \boldsymbol{\omega}_t = -\eta_t \boldsymbol{g}_t^n, \tag{9}$$

where η_t denotes the learning rate, and $\boldsymbol{g}_t^n$ denotes the gradients computed at the UE.

2) Split Training: The split training UEs $u_n, n \in \mathcal{N}_{S_t}$ train the global UE-side model ω_t^l over the local datasets to the cut layer C in parallel, and then send the output of the cut layer, the activations $\boldsymbol{a}_t^n$, to the BS. The BS is supposed to be super resourceful and can provide fast model training, such that it sequentially performs forward propagation to the BS-side model ω_t^e with the received activations $\boldsymbol{a}_t^n, n \in \mathcal{N}_{S_t}$ to calculate the loss function $L_n(\boldsymbol{\omega}_t^n)$. Then the gradients of the cut layer are computed and sent back to the UEs for the back propagation and updating the UE-side models, respectively.

Specifically, let us consider a split training UEs set $u_n \in \{u_1, u_2, ..., u_{N_S}\}, n \in \mathcal{N}_{S_t}$. At first, the activations of user u_1 are fed forward to the BS-side model for calculating the gradients, and thus the BS-side model and the UE-side model of user u_1 are updated based on the those gradients. Then, the actiations of user u_2 are fed forward to the updated BS-side model by the activations of user u_1, and thus the updated BS-side model and the UE-side model of user u_1 are updated. This process continues to update the BS-side model and the UE-side models with the activations received from all the split training users. Since the model updates of BS-side model are based on the updated BS-side model before, more local model updates will be obtained by the UEs starting from u_2.

Therefore, the local model updates of the split training UE u_n are given by

$$\begin{aligned}
&\Delta \boldsymbol{\omega}_t^1 = \boldsymbol{\omega}_{t+1}^1 - \boldsymbol{\omega}_t^1 = -\eta_t \boldsymbol{g}_t^1, \\
&\Delta \boldsymbol{\omega}_t^2 = \boldsymbol{\omega}_{t+1}^2 - \boldsymbol{\omega}_t^1 = -\eta_t \boldsymbol{g}_t^1 - \eta_t \boldsymbol{g}_t^2, \\
&\text{.....}, \\
&\Delta \boldsymbol{\omega}_t^{N_S} = \boldsymbol{\omega}_{t+1}^{N_S} - \boldsymbol{\omega}_t^1 = -\eta_t \boldsymbol{g}_t^1 - \eta_t \boldsymbol{g}_t^2 - ... - \eta_t \boldsymbol{g}_t^{N_S},
\end{aligned} \tag{10}$$

where $\boldsymbol{\omega}_t^1 = \boldsymbol{\omega}_t$, and the gradients of each UE $u_n, \forall n \in \mathcal{N}_{S_t}$ is calculated by $\boldsymbol{g}_t^1 = \nabla L_1(\boldsymbol{\omega}_t), \boldsymbol{g}_t^2 = \nabla L_2(\boldsymbol{\omega}_t - \boldsymbol{g}_t^1), ..., \boldsymbol{g}_t^{N_S} = \nabla L_{N_S}(\boldsymbol{\omega}_t - \boldsymbol{g}_t^1 -, ..., -\boldsymbol{g}_t^{N_S - 1})$.

3) Model Aggregation of HSFL: Accordingly, the new global models are updated at the BS by performing model aggregation of all the local model updates obtained from both federated

training UEs and split training UEs as

$$\boldsymbol{\omega}_{t+1} = \boldsymbol{\omega}_t - \Delta\boldsymbol{\omega}_t = \boldsymbol{\omega}_t - (\sum_{n\in\mathcal{N}_{F_t}} p_n \Delta\boldsymbol{\omega}_t^n + \sum_{n\in\mathcal{N}_{S_t}} p_n \Delta\boldsymbol{\omega}_t^n). \tag{11}$$

The average local model updates of the federated training UEs $u_n, n \in \mathcal{N}_{F_t}$ are given by

$$\begin{aligned}\sum_{n\in\mathcal{N}_{F_t}} p_n \Delta\boldsymbol{\omega}_t^n &= p_1\Delta\boldsymbol{\omega}_t^1 + p_2\Delta\boldsymbol{\omega}_t^2, ..., +p_{N_F}\Delta\boldsymbol{\omega}_t^{N_F} \\ &= -p_1\eta_t\boldsymbol{g}_t^1 - p_2\eta_t\boldsymbol{g}_t^2, ..., -p_{N_F}\eta_t\boldsymbol{g}_t^{N_F}.\end{aligned} \tag{12}$$

The average local model updates of the split training UEs $u_n, n \in \mathcal{N}_{S_t}$ are calculated by

$$\begin{aligned}\sum_{n\in\mathcal{N}_{S_t}} p_n \Delta\boldsymbol{\omega}_t^n &= p_1\Delta\boldsymbol{\omega}_t^1 + p_2\Delta\boldsymbol{\omega}_t^2, ..., +p_{N_S}\Delta\boldsymbol{\omega}_t^{N_S} \\ &= p_1(-\eta_t\boldsymbol{g}_t^1) + p_2(-\eta_t\boldsymbol{g}_t^1 - \eta_t\boldsymbol{g}_t^2)+, ..., +p_{N_S}(-\eta_t\boldsymbol{g}_t^1 - \eta_t\boldsymbol{g}_t^2 - ... - \eta_t\boldsymbol{g}_t^{N_S}) \\ &= -p_1\eta_t\boldsymbol{g}_t^1 - p_2\eta_t\boldsymbol{g}_t^2, ..., -p_{N_S}\eta_t\boldsymbol{g}_t^{N_S} - p_2\eta_t\boldsymbol{g}_t^1 - p_3(\eta_t\boldsymbol{g}_t^1 + \eta_t\boldsymbol{g}_t^2)-, ..., \\ &- p_{N_S}(\eta_t\boldsymbol{g}_t^1 + \eta_t\boldsymbol{g}_t^2+, ..., +\eta_t g_{N_S-1}^t) \\ &= -p_1\eta_t\boldsymbol{g}_t^1 - p_2\eta_t\boldsymbol{g}_t^2-, ..., -p_{N_S}\eta_t\boldsymbol{g}_t^{N_S} - p_2\Delta_{\boldsymbol{g}_2} - p_3\Delta_{\boldsymbol{g}_3}-, ..., -p_{N_S}\Delta_{\boldsymbol{g}_{N_S}},\end{aligned} \tag{13}$$

where $\Delta_{\boldsymbol{g}_2} = \eta_t\boldsymbol{g}_t^1, \Delta_{\boldsymbol{g}_3} = \eta_t\boldsymbol{g}_t^1 + \eta_t\boldsymbol{g}_t^2, ..., \Delta_{\boldsymbol{g}_{N_S}} = \eta_t\boldsymbol{g}_t^1 + \eta_t\boldsymbol{g}_t^2+, ..., +\eta_t\boldsymbol{g}_{N_S-1}^t$.

Since the BS-side models can be trained sequentially at the BS with the received activations from the split training UEs, the UEs $u_n, n \in \{2, ..., N_S\}$ will receive more local model updates $\Delta_{\boldsymbol{g}_n}$ than they are trained in parallel. This means if the same number of UEs are trained with federated training or split training, the later one will provide more local model updates in each round. Therefore, the model aggregation in the HSFL algorithm is derived as

$$\boldsymbol{\omega}_{t+1} = \boldsymbol{\omega}_t - (\sum_{n\in\mathcal{N}_F} p_n\eta_t\boldsymbol{g}_t^n + \sum_{n\in\mathcal{N}_S} p_n\eta_t\boldsymbol{g}_t^n + \sum_{n=2}^{N_S} p_n\Delta_{g_n}). \tag{14}$$

B. Wireless HSFL Algorithm

In the considered wireless UAV networks, the UEs and BS collaboratively train the ML model for accomplishing the object recognition task based on the transmission of model parameters with dynamic and randomly fading wireless channels. Considering the diversity of the UEs with different computational capabilities, dataset distribution and channel conditions, the HSFL algorithm is needed. The general procedure of the wireless HSFL algorithm is summarized in **Algorithm 1**.

Algorithm 1 Wireless HSFL Algorithm

1): Initialize global model $\boldsymbol{\omega}_t$, global UE-side model $\boldsymbol{\omega}_t^l$ and global BS-side model $\boldsymbol{\omega}_t^e$, set $t = 0$
2: **Repeat**
3: The BS selects a subset of UEs $\mathcal{K}$, then schedules UE set $\mathcal{K}_S$ on split training and UE set $\mathcal{K}_F$ on federated training.
4: The BS distributes $\boldsymbol{\omega}_t$ to the UE set $\mathcal{K}_F$ and distributes $\boldsymbol{\omega}_t^l$ to the UE set $\mathcal{K}_F$.
5: **for** UE $n \in \mathcal{K}$ **do**
6: **if** $n \in \mathcal{K}_F$ **then**
7: UE u_n computes $\Delta\boldsymbol{\omega}_t^n$ as in (9)
8: **else if** $n \in \mathcal{K}_S$ **then**
9: UE u_n collaborating with the BS computes $\Delta\boldsymbol{\omega}_t^n$ as in (10) with transmitting the activations and gradients of the cut layer in the uplink and downlink.
10: **end if**
11: **end for**
12: The BS computes the new global model as in (11)
13: Set $t = t + 1$
14: **Until** the desired convergence performance is achieved or the final iteration arrives

C. Convergence Analysis

In this section, we perform fundamental convergence analysis for our proposed wireless HSFL algorithm, where only a subset of UEs K are selected to participate in the global model training in one communication round due to the limited bandwidth and unreliable wireless communication links. We analyze the convergence performance of the proposed HSFL algorithm under non-IID data [26] with the random UE selection scheme. We first present the preliminaries and assumptions, and then the convergence result is obtained.

1) Preliminaries: The optimal solution of the global loss function $L(\boldsymbol{\omega})$ in (4) is defined as

$$\boldsymbol{\omega}^* \triangleq \arg\min_{\boldsymbol{\omega}} L(\boldsymbol{\omega}), \tag{15}$$

so the minimum loss is $L^* \triangleq L(\boldsymbol{\omega}^*)$. Similarly, the minimum loss of UE u_n is denoted by $L_n^* = L_n(\boldsymbol{\omega}^*)$. Then the local-global objective gap is defined as

$$\Phi \triangleq L^* - \sum_{n=1}^{N} p_n L_n^*, \tag{16}$$

where Φ is nonzero, which quantifies the degree of non-IID data, its magnitude reflects the heterogeneity of the data distribution, that is, larger Φ implies higher data heterogeneity over the UEs. If the data is IID, then Φ obviously goes to zero as the number of samples grows.

2) Assumptions: We make the following assumptions on the loss function and the stochastic gradients.

Assumption 1. $L_1, ..., L_n$ are all ℓ-smooth, i.e., for all $\boldsymbol{v}$ and $\boldsymbol{\omega}$, $L_n(\boldsymbol{v}) \leq L_n(\boldsymbol{\omega}) + (\boldsymbol{v} - \boldsymbol{\omega})^T \nabla L_n(\boldsymbol{\omega}) + \frac{\ell}{2} \|\boldsymbol{v} - \boldsymbol{\omega}\|_2^2$

Assumption 2. $L_1, ..., L_n$ are all μ-strongly convex, i.e., for all $\boldsymbol{v}$ and $\boldsymbol{\omega}$, $L_n(\boldsymbol{v}) \geq L_n(\boldsymbol{\omega}) + (\boldsymbol{v} - \boldsymbol{\omega})^T \nabla L_n(\boldsymbol{\omega}) + \frac{\mu}{2} \|\boldsymbol{v} - \boldsymbol{\omega}\|_2^2$

Assumption 3. Let ξ_t^n present the random sample dataset from the UE u_n. The variance of stochastic gradients in each UE is bounded: $\mathbb{E} \|\nabla L_n(\boldsymbol{\omega}_t^n, \xi_t^n) - \nabla L_n(\boldsymbol{\omega}_t^n)\|^2 \leq \delta_n^2, for\ n = 1, ..., N$

Assumption 4. The expected squared norm of the stochastic gradients is uniformly bounded, i.e., $\mathbb{E} \|\boldsymbol{g}_n(\boldsymbol{\omega}_n, \xi_n)\|^2 \leq G^2, for\ n = 1, ..., N$

3) Convergence Result: As discussed before, only a subset of UEs $\mathcal{K}_t$ is selected to join in the global model training in each communication round t. To establish the convergence bound, we need to make the assumption on the selected UEs first.

Assumption 5. Assuming that $\mathcal{K}_t$ is a subset of UEs including K UEs randomly sampled from the available UE set $\mathcal{N}_t$ including N UEs without replacement, so that the probability of each UE being selected to contribute global training is $\mathbb{P} = \frac{K}{N}$. Assuming that the data set is on non-IID and balanced in the sense that $p_1 = ... = p_N = \frac{1}{N}$, thus the model aggregation at the BS is fulfilled as $\boldsymbol{\omega}_{t+1} = \boldsymbol{\omega}_t - \frac{N}{K}(\sum_{n \in \mathcal{K}_{F_t}} p_n \Delta \boldsymbol{\omega}_t^n + \sum_{n \in \mathcal{K}_{S_t}} p_n \Delta \boldsymbol{\omega}_t^n)$.

Theorem 1. Let **Assumptions 1,2,3,4** and **5** hold, we assume $\varrho = \frac{2}{\mu}$ with $\iota = \frac{4\ell}{\mu}$ and let $\kappa = \frac{\ell}{\mu}$, then the proposed HSFL algorithm with K UEs selected for participation satisfies

$$\mathbb{E}\left[L(\boldsymbol{\omega}_T)\right] - L^* \leq \frac{\kappa}{\iota + T - 1} \left(\frac{2W}{\mu} + \frac{\mu\iota}{2} \mathbb{E} \|\boldsymbol{\omega}_1 - \boldsymbol{\omega}^*\|^2 \right), \tag{17}$$

where $W = \sum_{n=1}^{N} p_n^2 \delta_n^2 + 6\ell\Phi + 8\eta_t^2 G^2 (1 - \frac{N_S(N_S-1)}{2N}) + \frac{N-K}{K(N-1)}(4N - 2N_S(N_S-1))G^2$.

Proof. The proof is presented in Appendix 1. □

From **Theorem 1**, we can conclude that the increment of total communication rounds leads to the convergence of our proposed HSFL algorithm. Moreover, the convergence performance has a weak dependence on the number of selected UEs K, but the convergence speed increases with the increasing number of UEs on split training N_S.

IV. User Selection

In this section, we present the details of our proposed UE selection schemes. To train the ML model with the dataset distributed over the diverse UEs in wireless UAV networks, the limited bandwidth and dynamic communication channels make the BS cannot access all the UEs in each round. Additionally, different local model updates are of dissimilar importance to the model convergence [15], [24]. Therefore, it's essential to develop efficient UE selection schemes to select a subgroup of UEs that provide the most useful information in each round.

A. UE Selection Scheme

The channel quality and importance of local model updates are two key concerns when developing UE selection schemes, the authors in [24] developed the BC and BN2 UE selection schemes.

1) BC UE Selection Scheme: In this scheme, the BS does not need any information about the local model updates of the UEs, and only selects $K \leq N$ UEs with the best channel qualities from the available UE set $\mathcal{N}$ in round t,

$$\mathcal{K}_t = max_{[K]}\{\gamma_t^1, ..., \gamma_t^N\}, \tag{18}$$

where γ_t^1 denotes the SNR of user u_1 in round t.

2) BN2 UE Selection Scheme: This scheme requires an extra estimation phase, the BS requires all the UEs to compute their local model updates $\Delta\boldsymbol{\omega}_t^n$ and send back $\|\Delta\boldsymbol{\omega}_t^n\|_2$ representing the importance of local model updates in the first estimation phase. Then the BS selects K devices with the largest $\|\Delta\boldsymbol{\omega}_t^n\|_2$ in round t, which is given as

$$\mathcal{K}_t = max_{[K]}\{\left\|\Delta\boldsymbol{\omega}_t^1\right\|_2, ..., \left\|\Delta\boldsymbol{\omega}_t^N\right\|_2\}. \tag{19}$$

The authors in [24] then proposed a UE selection scheme that jointly considering channel qualities and the importance of local model updates, which provides a better long-term performance than scheduling policies based only on either of the two metrics individually. However, in

practice, it is difficult to obtain accurate channel conditions and local model update information before learning procedure is conducted, it also consumes extra computation and communication resources to estimate each UE's local model updates in the extra estimation phase. Fortunately, the dynamic MAB-based UE selection scheme [16] can address this problem by selecting UEs according to the estimated information using trail-and-error rule. In this case, we do not need the pre-estimation step to estimate UE information in each training round. Hence, in this section, we will exploit MAB algorithm to solve the dynamic UE selection problem by jointly considering channel qualities and the importance of local model updates.

B. MAB-based UE Selection Scheme

In this section, we present our proposed MAB-based UE selection scheme, which formulates the UE selection in the wireless HSFL algorithm as a MAB problem, and uses the discounted UCB policy to estimate the UEs with expected larger local model updates and better channel quality. This scheme provides an exploitation-exploration trade-off to select UEs with both larger local model update and better channel quality (i.e., exploitation) as that leads to faster convergence [27], and also to ensure UE diversity (i.e., exploration) [16].

Knowing that the importance of local model updates $\|\Delta\boldsymbol{\omega}_t^n\|_2$ and the channel quality γ_t^n of UE u_n are non-stationary during communication rounds, we apply the discounted MAB algorithm [28]. The discounted UCB algorithm has been modified for UE selection in [16] by measuring the local loss values of the UEs and received good performance. Therefore, we first propose MAB-BC and MAB-BN2 UE selection schemes by modifying the discounted UCB algorithm taking into account the channel qualities and the local model updates, respectively. To jointly consider both of them, we propose a novel MAB-BC-BN2 UE selection scheme.

Our proposed MAB-BC-BN2 UE selection scheme is based on UCB policy which makes decisions depending on the UCB score. It performs exploration by selecting UEs that are selected less often, and exploitation by selecting the UEs with the largest reward. We view the UEs as the arms in the MAB problem and separately compute discounted cumulative values of the $\|\Delta\boldsymbol{\omega}_t^n\|_2$ and the SNRs, i.e., $\Omega_t^n(\lambda)$ and $\Gamma_t^n(\lambda)$, as the cumulative rewards, and a discounted count of the number of times each UE has been selected, $M_t^n(\lambda)$, till communication round t.

Thus, the discounted UCB score for each UE u_n in communication round t is defined as

$$A_t^n(\lambda) = p_n f(K, n), \tag{20}$$

where p_n is the dataset size ratio of UE u_n, λ denotes the discount factor, and $f(K,n)$ is the UCB index function.

*1) **MAB-BC UE selection scheme**:* If we only consider UE's channel quality as the reward in the considered MAB problem, the UCB index function $f(K,n)$ is given by

$$f(K,n) = \bar{\Gamma}_t^n(\lambda_c). \tag{21}$$

*2) **MAB-BN2 UE selection scheme**:* If we only consider the importance of UE's local model updates as the reward in the considered MAB problem, the UCB index function $f(K,n)$ is given by

$$f(K,n) = \bar{\Omega}_t^n(\lambda_l). \tag{22}$$

*3) **MAB-BC-BN2 UE selection scheme**:* In this scheme, by jointly considering channel conditions and the importance of local model updates as the reward, the UCB index function $f(K,n)$ is defined as

$$f(K,n) = \beta\bar{\Omega}_t^n(\lambda_l) + (1-\beta)\bar{\Gamma}_t^n(\lambda_c), \tag{23}$$

In (23), two terms represent the importance of local model updates and the channel quality, respectively, and β is the balance factor between them.

$$\begin{aligned} \bar{\Omega}_t^n(\lambda_l) &= \frac{\Omega_t^n(\lambda_l)}{M_t^n(\lambda_l)} + \sqrt{2\sigma_t^2\frac{\log(T_t(\lambda_l))}{M_t^n(\lambda_l)}} \\ \bar{\Gamma}_t^n(\lambda_c) &= \frac{\Gamma_t^n(\lambda_c)}{M_t^n(\lambda_c)} + \sqrt{2\sigma_t^2\frac{\log(T_t(\lambda_c))}{M_t^n(\lambda_c)}} \end{aligned} \tag{24}$$

$$\begin{aligned} \Omega_t^n(\lambda_l) &= \sum_{\tau=1}^{t}\lambda_l^{t-\tau}\mathbb{I}_{\{n\in\mathcal{K}_{t-1}\}}\Delta\boldsymbol{\omega}_t^n \\ \Gamma_t^n(\lambda_c) &= \sum_{\tau=1}^{t}\lambda_c^{t-\tau}\mathbb{I}_{\{n\in\mathcal{K}_{t-1}\}}\gamma_t^n \\ M_t^n(\lambda_i) &= \sum_{\tau=1}^{t}\lambda_i^{t-\tau}\mathbb{I}_{\{n\in\mathcal{K}_{t-1}\}}, \quad T_t(\lambda_i) = \sum_{\tau=1}^{t}\lambda_i^{t-\tau}, i\in\{l,c\} \end{aligned} \tag{25}$$

Here, the discount factor $0 \le \lambda_i \le 1$ indicates the significance of stale values, $\lambda_i = 1$ means all the past rewards contribute equally to the calculation of $\Omega_t^n(\lambda_l)$ and $\Gamma_t^n(\lambda_c)$, and $\lambda_i = 0$ indicates that only the latest reward is used to estimate the value. Thus, $0<\lambda_i<1$ means putting less weight of stale rewards to calculate the $\Omega_t^n(\lambda_l)$ and $\Gamma_t^n(\lambda_c)$. This can avoid the noise in the latest evaluation and the discounted stale rewards computed in the past while computing the

Algorithm 2 MAB-BC-BN2 UE Selection Algorithm

Initialization

Input: K, K_S, K_F, β λ, p_n for $n \in N$

Initialization: Randomly select $\mathcal{K}_0$, $\mathcal{K}_{S_0}$ and $\mathcal{K}_{F_0}$; a list $\mathcal{A}$ of length N; $t = 1$

Learning:

for $t \leq T$ **do**
  for $i \in K$ **do**
    The BS distributes $\boldsymbol{\omega}_t$ to $u_n, n \in \mathcal{K}_{F_{t-1}}$ and $\boldsymbol{\omega}_{t,l}$ to $u_n, n \in \mathcal{K}_{S_{t-1}}$.
    UEs respectively train the global model with respect to their local dataset.
    UEs compute the l_2-norm $\|\Delta\boldsymbol{\omega}_t^n\|_2$ of the local model update as (9) and (10), and then upload them to the BS.
  end for
  The BS receives the local model updates and measures the received SNR γ_t^n of each UE.
  The BS calculates the UCB score $A_t^n(\lambda)$ and updates list $\mathcal{A}[n] = A_t^n(\lambda)$.
  The BS generates a UE set $\mathcal{K}_t$ = { K clients with the largest values in $\mathcal{A}$ }, and assign the UE set $\mathcal{K}_{S_t}$ and $\mathcal{K}_{F_t}$
  Update the elements in $\mathcal{A}$ by $\mathcal{A} = \lambda\mathcal{A}$
end for
Return selected UE set $\mathcal{K}_t$, $\mathcal{K}_{S_t}$ and $\mathcal{K}_{F_t}$.

estimated values. In practice, the discount factor λ_i in two terms, i.e., $\Omega_t^n(\lambda_l)$ and $\Gamma_t^n(\lambda_c)$, can be set as different values, which means the past rewards may have different impacts on channel qualities and the significance of local model updates. The λ_c can be set based on the empirical fluctuation of wireless channels, while λ_l can be set based on the dataset distributions over the UEs. In the exploration term $\sqrt{2\sigma_t^2 \frac{\log(T_t(\lambda_i))}{M_t^n(\lambda_i)}}$, σ_t is a hyper-parameter controlling the degree of exploration, which is defined as the maximum standard deviation in the reward computed over the latest update of the UEs. If the UE has not been selected very often, or not at all, then $M_t^n(\lambda_i)$ will be small so that the exploration term will be large, making this UE more likely to be selected. As time progresses, the exploration term gradually decreases (due to $(logn)/n$ goes to zero as n goes to infinity) until eventually UEs are selected based only on the exploitation term. Therefore, we propose a MAB-BC-BN2 UE selection algorithm to accomplish the MAB-based

UE selection scheme with the detailed process provided in **Algorithm 2**.

V. Experiments and Numerical Results

In this section, the learning performance of our proposed HSFL algorithm is provided, which is compared with the CL algorithm and the state-of-art distributed learning algorithms, including FL, SL and SFL algorithm, by simulating the learning task, image recognition, in wireless UAV networks using classical MINST dataset [29]. This image classification task relying on aerial UEs, i.e., UAVs, to collect dataset, has been investigated in many practical scenarios, such as mapping applications [30] and damage assessment for post disaster analysis [31]. We then compare the performance of different UE selection schemes for selecting UEs to join in model training with wireless HSFL algorithm under IID, non-IID, Dirichlet-nonIID and Dirichlet-ImD data.

A. Experiment Environment

The experiments are conducted by the laptop with one NVIDIA RTX 2070 GPU and Intel i7-10750H CPUs, where the BS's programming code is running on the GPU while the UEs' programming codes are running on the CPU. We consider training two different DNN models, Net and AlextNet, on MINST dataset, the architectures of which are shown in Table I. For all the experiments using SL, SFL and HSFL algorithm, the DNN network is split in the second layer, i.e., after the first *conv1* layer. In this paper, to verify the learning performance of the proposed HSFL framework, we simulate a wireless UAV network with one BS located at the origin of the cell and multiple UAVs uniformly distributed within the cell. The cell radius is 500 m, the height of the BS antenna is 20 m and the UAV's flying height is in the range of 20-80 meters. The detailed simulation parameters of the UAV networks are provided in Table II.

B. Learning Performance Comparisons

In this section, the learning performance of our proposed wireless HSFL algorithm is studied in terms of test accuracy, training time and communication overhead. We adopt BC UE selection scheme to select $K = 10$ UEs from $N = 100$ UEs for training in each round, and set $K_s = K_F = 5$ for HSFL algorithm. The IID and non-IID data follow the same settings in [5]. We set the local training rounds $\tau = 5$ and the batch size $b = 10$. The local learning (LL) and CL algorithms are also simulated as the benchmarks. In LL algorithm, each user is training the DNN

TABLE I

DNN MODEL ARCHITECTURE

Architecture	No. of parameters	layers	Kernel size
Net	60 thousands	4	(5 × 5), (5 × 5)
AlexNet	60 million	8	(3 × 3), (3 × 3), (3 × 3), (3 × 3), (3 × 3)

TABLE II

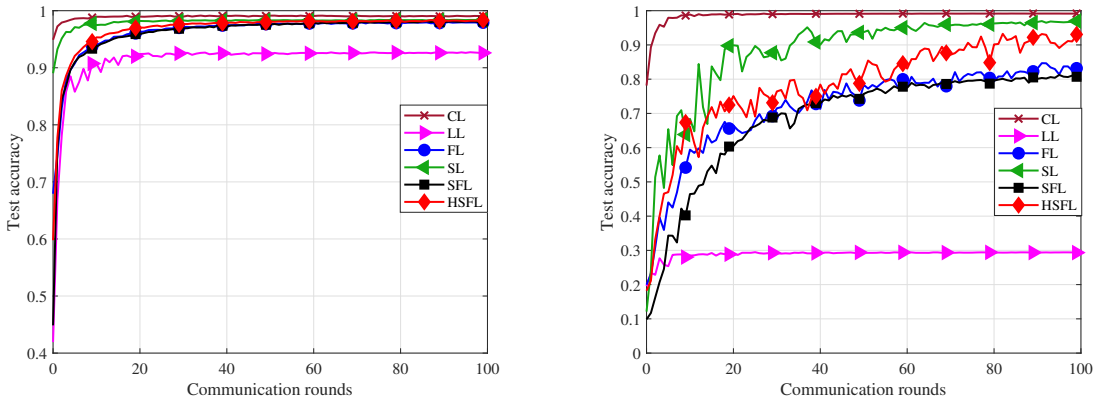
SIMULATION PARAMETERS OF THE WIRELESS UAV NETWORK

$Parameters$	$Value$
φ_l, φ_n	$21, 1$
a, b	$5.0188, 0.3511$
Rician factor Kdb	2 dB
system carrier frequency	2 GHz
Noise power σ^2	-130 dBm
P_s, P_n	40 dBm, 23 dBm
B_s, B_w	5 MHz, 1MHz

model locally without sharing any raw data or model parameters to the BS. In CL algorithm, all the users have to send their raw data to the BS for performing centralized training.

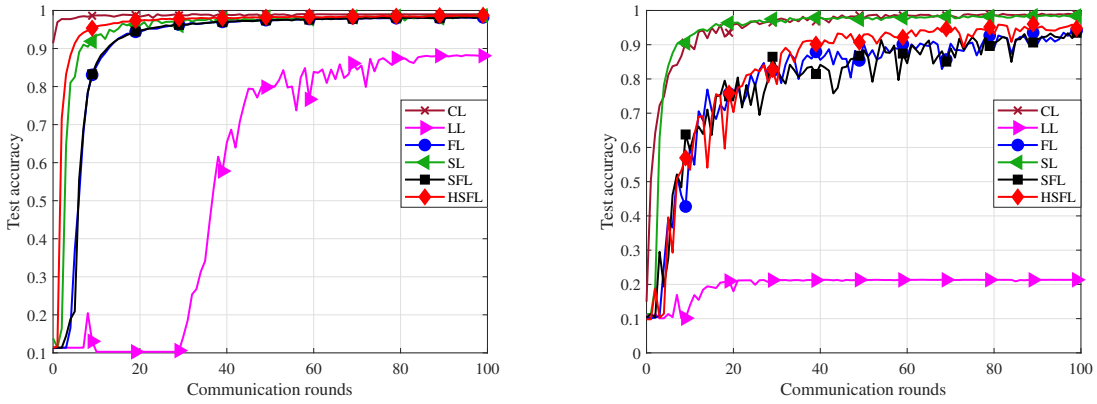
1) Learning Accuracy Performance: In Fig. 4 and 5, the learning accuracy performance of Net and AlexNet is presented, respectively. The CL has the highest learning accuracy while training both ML models. From Fig. 4, we can observe that the HSFL algorithm provides similar test accuracy performance as SL (sequentially SL can be viewed as optimal as the centralized learning) and better test accuracy performance than FL and SFL under both IID and non-IID data. This is because in HSFL algorithm, half number of the UEs perform split training which brings the superiority in test accuracy performance. In Fig. 5, it shows that the learning accuracy performance of using different learning algorithms to train AlexNet has similar trend to train Net shown in Fig. 4, and it takes less communication rounds to converge. From Fig. 5, the HSFL algorithm also has better learning accuracy performance than FL and SFL, and it converges faster.

2) Training Time and Communication Overhead: The training time is calculated to include two parts, computation and communication time. The computation time is monitored by using *time* module in Python, while the UE's local training is simulated by running on the CPU and the



(a) IID data (b) non-IID data

Fig. 4. Test accuracy performance of different wireless distributed learning algorithms on IID and non-IID data with BC UE selection scheme, $K = 10, N = 100, K_s = 5$.

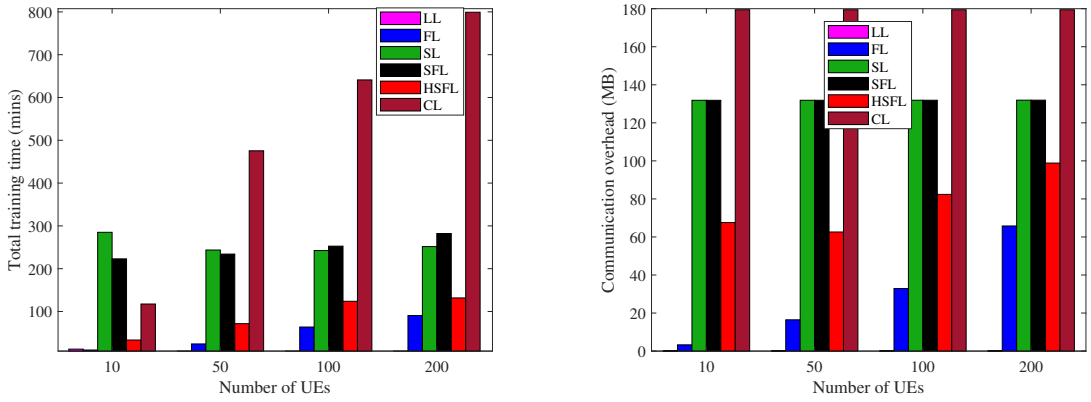


(a) AlexNet with IID data (b) AlexNet with non-IID data

Fig. 5. Test accuracy performance of different wireless distributed learning algorithms on IID and non-IID data with BC UE selection scheme and using AlexNet model, $K = 10, N = 100, K_s = 5$.

BS's model aggregation or training are running on the GPU on my laptop. On the other side, the communication time is calculated by simulating the transmission process of model parameters through the wireless channels in wireless UAV networks, the simulation parameters related to the wireless transmission links are shown in Table II.

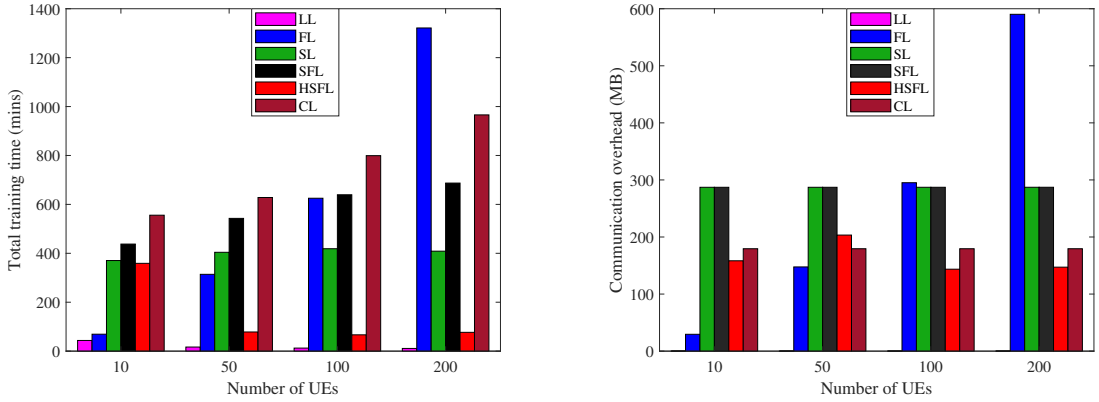
The communication overhead mainly includes two parts of calculation, model size and smashed data including activations and gradients of the cut layer in split learning. The model size is calculated by its model parameters, where each parameter is represented by a standard 32-bit floating point. The size of activations and gradients is computed by calculating the output size



(a) Total training time with Net

(b) Communication overhead per round with Net

Fig. 6. Training time and communication overhead comparisons of different wireless distributed learning algorithms on IID data with BC UE selection scheme, $N = 10, 50, 100, 200$, $K = 1, 5, 10, 20$, $K_S = 0/1, 2, 5, 10$



(a) Total training time with AlexNet

(b) Communication overhead per round with AlexNet

Fig. 7. Training time and communication overhead comparisons of different wireless distributed learning algorithms on IID data with BC UE selection scheme, $N = 10, 50, 100, 200$, $K = 1, 5, 10, 20$, $K_S = 0/1, 2, 5, 10$

of the cut layer.

We consider four scenarios with the number of UEs $N = 10, 50, 100, 200$, and select $K = 1, 5, 10, 20$ UEs in each scenario, respectively, in each round for model aggregation on IID and non-IID data. In HSFL, the number of split training UEs in each scenario is set as $K_S = 0/1, 2, 5, 10$, note that when $K = 1$, the UE either performs split training with $K_S = 1$, or performs federated training with $K_S = 0$. We set $B_w = 1$ MHz, which is shared by the selected UEs in each round with each UE allocated with the same bandwidth. SL adopts sequential training, where only one UE takes up the whole bandwidth in each round in all the considered

scenarios. In contrast, FL, SFL and HSFL are training in parallel, so that all the UEs selected in each round will share the whole bandwidth.

Fig. 6 (a) shows the total training time over UEs $N = 10, 50, 100, 200$ four scenarios when training the learning model Net. The CL has the highest training time over all the scenarios because in which all the UEs have to upload their raw dataset to the BS in each scenario. The training time of FL increases with increasing number of UEs because the bandwidth allocated to each UE is decreasing. The SL and SFL experience similar training time performance since the total bandwidth is fixed and the training time mainly depends on the communication latency. Compared to SL and SFL, HSFL spends less training time because in this case only half of the selected UEs share the total bandwidth while performing split training, which reduces the communication latency for each communication round. Likewise, Fig. 7 (a) shows the total training time over UEs when training on AlexNet. In this case, the total training time of FL is increasing significantly with the increasing number of UEs because it makes the communication overhead increase significantly, Since the model size of AlexNet is larger than Net, communication overhead will increase a lot with the increasing number of UEs to send their model parameters to the BS. However, the proposed HSFL algorithm has the shortest training time compared to the other distributed algorithms, except local learning, which follows the same reason as training on Net shown in last subsection. Moreover, we can observe that the training time of FL is larger than SL, SFL and even CL when the dataset is distributed over 200 UEs.

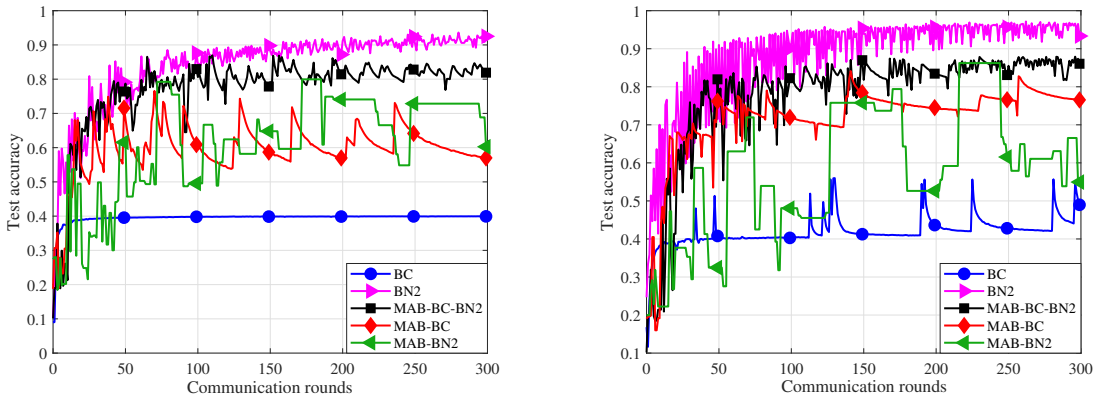
Fig. 6 (b) plots the total communication overhead per round when training on Net. The CL has the highest communication overhead over all the distributed algorithms when training on Net. However, when training on AlexNet, the communication overhead of CL becomes less than most distributed learning algorithms. This is because the model size of ALexNet is larger than Net, which causes large communication overhead for the distributed learning algorithms that include model parameters transmission. The communication overhead of FL is increasing over UEs $N = 10, 50, 100, 200$ because more users need to transmit their model parameters. The communication overhead of SL and SFL are almost the same and keeps unchanged over the increasing number of UEs, this is because the communication overhead of them, i.e., the activations and gradients of the cut layer, is decided by the size of local dataset at each UE. Specially, HSFL has almost half less communication overhead than SL and SFL in each scenario since it only includes half number of UEs for split training and the other half number of UEs for federated training. In Fig. 7(b), the total communication overhead per round when training

on AlexNet with different distributed learning algorithms is shown. In this case, We can see that FL is less communication efficient than SL and SFL when the number of UEs increased to 100, while it is more communication efficient than SL and SFL when training on Net shown in Fig. 6 (b).

C. Performance Comparisons of UE Selection Schemes

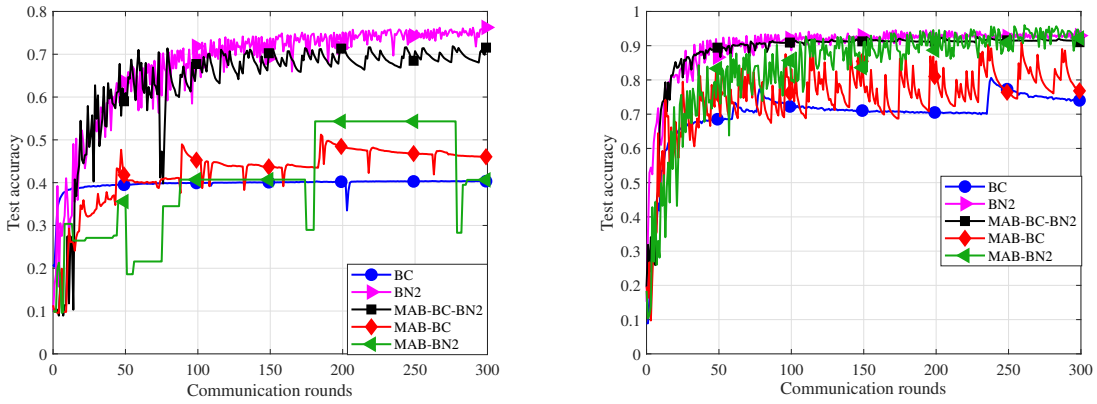
In this section, the learning accuracy performance of our proposed MAB-BC-BN2 UE selection scheme is evaluated over the non-IID and imbalanced data in wireless HSFL algorithm and wireless FL algorithm. We set $N = 30$, $\tau = 5$ and $b = 64$. We consider two non-IID data distribution settings; 1) For non-IID, it follows the similar settings in [5], where the dataset is first sorted by digit label, and it is divided into 60 shards of size 1000, and then each of 30 UEs is assigned with 2 shards. 2) For Dirichlet-nonIID, the whole dataset is partitioned among 30 UEs following the Dirichlet distribution Dir(α_d) [32], where smaller α_d indicates larger data heterogeneity across UEs. We set $\alpha_d = 0.01$. 3) For Dirichlet-ImD, we also construct the imbalanced data partition among 30 UEs using this Dirichlet distribution Dir(α_d, α_{imd}), where smaller α_{imd} indicates the dataset size across UEs is more imbalanced. We set $\alpha_d = 0.1, \alpha_{imd} = 2$.

Fig. 8 plots the test accuracy of different UE selection schemes in wireless FL and HSFL algorithms. In both FL and HSFL, we can observe that the BN2 UE selection scheme achieves the best test accuracy performance because it takes an extra round to estimate the importance of local model updates of all the UEs, so that the BS can select a subset of UEs with the largest local model updates to participate in training in each round. In contrast, the BC UE selection scheme has worst learning performance, since it always selects the UEs with the best channel qualities and neglects the importance of their local model updates. The MAB-BC-BN2 scheme has similar test accuracy performance as the BN2 scheme, which jointly considers both channel conditions and the importance of local model updates. The MAB-BN2 and MAB-BC UE selections schemes are shown with lower test accuracy than the MAB-BC-BN2 scheme. This is because in MAB-BN2, the selected UEs with large local model updates may fail to upload the local model updates due to bad channel conditions. And in MAB-BC scheme, the selected UEs with good channel conditions may have low local model updates. Note that MAB-BC has better performance than BC scheme, this is because MAB-BC adopts exploitation-exploration



(a) FL (b) HSFL, $K_S = 2$

Fig. 8. Performance comparisons of different UE selection schemes, $N = 30, K = 3$, on non-IID data, $\lambda_l = \lambda_c = 0.99, \beta = 0.5$



(a) Dirichlet-nonIID data, $\alpha_d = 0.01$ (b) Dirichlet-ImD data, $\alpha_d = 0.1, \alpha_{imd} = 2$

Fig. 9. Test accuracy of different UE selection schemes with HSFL on Dirichlet-nonIID and Dirichlet-ImD data, $N = 30, K = 3, K_s = 2, \lambda_l = \lambda_c = 0.99, \beta = 0.5$

rule that enables it to explore the UEs with less optimal channel conditions. It then increases the chance to include the UEs with larger local model updates for training.

In Fig. 9, we compare test accuracy performances of different UE selection schemes in wireless HSFL algorithm using Dirichlet-nonIID and Dirichlet-ImD data. The comparisons of different UE selection schemes follow similar trend as Fig. 8 (b) using non-IID data. In Fig. 9 (a), the test accuracy performances of all the UE selection schemes are worse than Fig. 8 (b) due to larger heterogeneity of dataset over all the UEs. However, the test accuracy of UE selection schemes in Fig. 9 (b) shows better performance because the dataset across UEs is less heterogeneous even

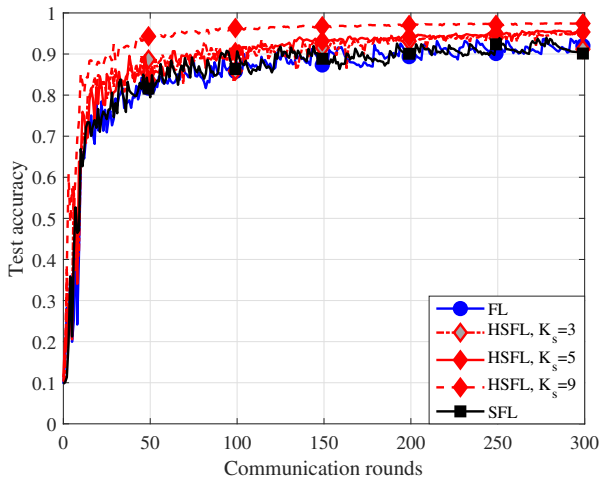


Fig. 10. Test accuracy performance of HSFL with MAB-BC-BN2 scheme on non-IID data over parameters K_S, $N = 100, K = 10, \beta = 0.9$

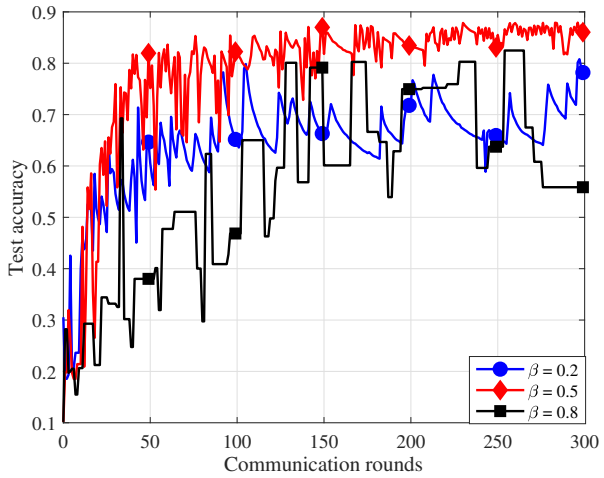


Fig. 11. Test accuracy performance of HSFL with MAB-BC-BN2 scheme on non-IID data over parameters β, $N = 30, K = 3, K_S = 2$.

it's imbalanced.

Fig. 10 plots the test accuracy of wireless HSFL algorithm with MAB-BC-BN2 UE selection scheme using non-IID data for various numbers of split training UEs K_s. Compared to FL and SFL, the HSFL algorithm achieves better test accuracy performance and its superiority is increasing with increasing the number of split training UEs, K_S. In Fig. 11, we examine the impact of balance factor β in MAB-BC-BN2 scheme on HSFL learning performance in terms of

test accuracy. We can see that $\beta = 0.5$ has the best learning performance in our simulated UAV networks, which reveals that selecting the UEs that satisfying the lowest SNR requirements and with larger local model updates will facilitate the improvement of test accuracy performance. In practice, if the dataset over the UEs is on IID, each user would have similar local model updates, in this case, more weight could be put on the channel qualities, that is, β can be set as a smaller value. On the other hand, if the dataset over the UEs is on non-IID, more weight could be put on the significance of local model updates to get better convergence performance.

VI. Conclusion

In this paper, we proposed a novel distributed learning architecture, namely hybrid split and federated learning (HSFL) algorithm, which adopts the parallel model training mechanism of federated learning (FL) and model splitting structure of split learning (SL). By applying our HSFL algorithm in wireless UAV networks, our results demonstrated it achieved higher learning accuracy than FL, and less communication overhead than SL under independent and identically distributed (IID) and non-IID data. Our results also revealed the learning accuracy performance of HSFL algorithm can be improved with increasing the number of split training UEs. We also provided convergence analysis for wireless HSFL algorithm under non-IID data with random UE selection scheme. To improve the learning performance of our proposed HSFL algorithm in wireless networks under limited bandwidth and dynamic channel conditions, we developed a Multi-Arm Bandit (MAB) based best channel (BC) and best 2-norm (BN2) (MAB-BC-BN2) UE selection scheme based on discounted MAB algorithm to select the UEs with larger local model updates and better channel qualities in each round. Our results have shown that MAB-BC-BN2 UE selection scheme achieved better learning accuracy performance compared to BC, MAB-BC and MAB-BN2 under non-IID, Dirichlet-non-IID and Dirichlet-Imbalanced data.

Appendix A

We analyze the proposed HSFL scheme in the setting of partial UEs participation on non-IID data in this Section. In this scenario, the BS randomly selects a subset of UEs K according to the sampling schemes (like BC, BN2, or MAB-based UE selection scheme). We define $\boldsymbol{g}_t = \sum_{n=1}^{N} p_n \boldsymbol{g}_t^n(\boldsymbol{\omega}_t^n, \xi_t^n)$ and $\bar{\boldsymbol{g}}_t = \sum_{n=1}^{N} p_n \bar{\boldsymbol{g}_t^n}(\boldsymbol{\omega}_t^n)$, thus, $\mathbb{E}\boldsymbol{g}_t = \bar{\boldsymbol{g}}_t$.

First, from (26), we bound the average of the terms $A_1, A_2 \; and \; A_3$. They are explained in three **Lemmas** where the proof of each is included.

$$\begin{aligned}\|\boldsymbol{\omega}_{t+1}-\boldsymbol{\omega}^*\|^2 &= \|\boldsymbol{\omega}_{t+1}-\boldsymbol{v}_{t+1}+\boldsymbol{v}_{t+1}-\boldsymbol{\omega}^*\|^2 \\ &= \underbrace{\|\boldsymbol{\omega}_{t+1}-\boldsymbol{v}_{t+1}\|^2}_{A_1}+\underbrace{\|\boldsymbol{v}_{t+1}-\boldsymbol{\omega}^*\|^2}_{A_2}+\underbrace{2\left\langle\boldsymbol{\omega}_{t+1}-\boldsymbol{v}_{t+1},\boldsymbol{v}_{t+1}-\boldsymbol{\omega}^*\right\rangle}_{A_3}\end{aligned} \tag{26}$$

Lemma 1. To bound A_3, Let $\mathbb{E}_{\mathcal{N}_t}$ denote expectation over the UE selection randomness at the round t. We have

$$\mathbb{E}_{\mathcal{N}_t}[\boldsymbol{\omega}_{t+1}]=\boldsymbol{v}_{t+1} \tag{27}$$

from which it follows that

$$\mathbb{E}_{\mathcal{N}_t}[\langle\boldsymbol{\omega}_{t+1}-\boldsymbol{v}_{t+1},\boldsymbol{v}_{t+1}-\boldsymbol{\omega}^*\rangle]=0 \tag{28}$$

Proof. Due to the randomness of the UE selection policy, it has

$$\mathbb{E}_{\mathcal{N}_t}[\frac{1}{K}\sum_{n\in\mathcal{N}_t}\boldsymbol{\omega}_t^n]=\frac{\binom{N-1}{K-1}}{K\binom{N}{K}}\sum_{n=1}^{N}\boldsymbol{\omega}_t^n=\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{\omega}_t^n \tag{29}$$

□

Lemma 2. To bound A_1, we have

$$\mathbb{E}\,\|\boldsymbol{\omega}_{t+1}-\boldsymbol{v}_{t+1}\|^2\leq\frac{N-K}{KN(N-1)}\eta_t^2(4N-2N_S(N_S-1))G^2 \tag{30}$$

Proof. It's following as in (31),

$$\begin{aligned}
&\mathbb{E}\|\boldsymbol{\omega}_{t+1}-\boldsymbol{v}_{t+1}\|^2=\mathbb{E}\left\|\tfrac{1}{K}\sum_{n\in\mathcal{N}_{t+1}}\boldsymbol{\omega}_{t+1}^n-\boldsymbol{v}_{t+1}\right\|^2=\tfrac{1}{K^2}\mathbb{E}\left\|\sum_{n=1}^{N}\mathbb{I}\{n\in\mathcal{N}_{t+1}\}(\boldsymbol{\omega}_{t+1}^n-\boldsymbol{v}_{t+1})\right\|^2\\
&=\tfrac{1}{K^2}\mathbb{E}_{\mathcal{N}_t}\left[\sum_{n\in[N]}\mathbb{P}(n\in\mathcal{N}_{t+1})\left\|\boldsymbol{\omega}_{t+1}^n-\boldsymbol{v}_{t+1}\right\|^2+\sum_{n\neq j}\mathbb{P}(n,j\in\mathcal{N}_{t+1})\left\langle\boldsymbol{\omega}_{t+1}^n-\boldsymbol{v}_{t+1},\boldsymbol{v}_{t+1}^j-\boldsymbol{v}_{t+1}\right\rangle\right]\\
&=\tfrac{1}{KN}\sum_{n=1}^{N}\mathbb{E}\left\|\boldsymbol{\omega}_{t+1}^n-\boldsymbol{v}_{t+1}\right\|^2+\sum_{n\neq j}\tfrac{K-1}{KN(N-1)}\mathbb{E}\left\langle\boldsymbol{\omega}_{t+1}^n-\boldsymbol{v}_{t+1},\boldsymbol{v}_{t+1}^j-\boldsymbol{v}_{t+1}\right\rangle\\
&=\tfrac{N-K}{KN(N-1)}\sum_{n=1}^{N}\mathbb{E}\left\|\boldsymbol{\omega}_{t+1}^n-\boldsymbol{v}_{t+1}\right\|^2\\
&=\tfrac{N-K}{K(N-1)}\mathbb{E}\left[\tfrac{1}{N}\sum_{n=1}^{N}\left\|(\boldsymbol{\omega}_{t+1}^n-\boldsymbol{\omega}_{t_0})-(\boldsymbol{v}_{t+1}-\boldsymbol{\omega}_{t_0})\right\|^2\right]\\
&=\tfrac{N-K}{K(N-1)}\mathbb{E}\left[\tfrac{1}{N}\sum_{n=1}^{N}\left\|\boldsymbol{\omega}_{t+1}^n-\boldsymbol{\omega}_{t_0}\right\|^2\right]\\
&\leq\tfrac{N-K}{K(N-1)}\sum_{\tau=t_0}^{t}\tfrac{1}{N}\sum_{n=1}^{N}\mathbb{E}\|\eta_\tau\boldsymbol{g}_\tau^n\|^2-\tfrac{1}{N}\sum_{n_s=2}^{N_S}\mathbb{E}\|\eta_\tau\Delta\boldsymbol{g}_\tau^{n_s}\|^2\\
&=\tfrac{N-K}{K(N-1)}\sum_{\tau=t_0}^{t}\mathbb{E}\|\eta_\tau\boldsymbol{g}_\tau^n\|^2-\tfrac{1}{N}(\mathbb{E}\|\eta_\tau\boldsymbol{g}_\tau^1\|^2+\mathbb{E}\|\eta_\tau(\boldsymbol{g}_\tau^1+\boldsymbol{g}_\tau^2)\|^2+...+\mathbb{E}\left\|\eta_\tau(\boldsymbol{g}_\tau^1+\boldsymbol{g}_\tau^2,...,+\boldsymbol{g}_\tau^{N_S-1})\right\|^2)\\
&\leq\tfrac{N-K}{K(N-1)}\eta_\tau^2(G^2-\tfrac{1}{N}(\tfrac{N_S(N_S-1)}{2}G^2))\\
&=\tfrac{N-K}{KN(N-1)}\eta_t^2(4N-2N_S(N_S-1))G^2
\end{aligned}\tag{31}$$

by using the conditions (1) $\mathbb{E}\left\|X-\mathbb{E}X\right\|^2=\mathbb{E}\|X\|^2-\|\mathbb{E}X\|^2$ and $\left\|\sum_{n=1}^{N}b_n\right\|^2\leq N\sum_{n=1}^{N}\|b_n\|^2$
(2) $\mathbb{P}(n\in\mathcal{N}_{t+1})=\frac{K}{N}$ and $\mathbb{P}(n,j\in\mathcal{N}_{t+1})=\frac{K(K-1)}{N(N-1)}$,
(3) $\sum_{n\in[N]}\left\|\boldsymbol{v}_{t+1}^n-\boldsymbol{v}_{t+1}\right\|^2+\sum_{n\neq j}\left\langle\boldsymbol{v}_{t+1}^n-\boldsymbol{v}_{t+1},\boldsymbol{v}_{t+1}^j-\boldsymbol{v}_{t+1}\right\rangle=0$

□

Lemma 3. To bound A_2, we have

$$\mathbb{E}\left\|\boldsymbol{v}^{t+1}-\boldsymbol{\omega}^*\right\|^2\leq(1-\eta_t\mu)\|\boldsymbol{\omega}_t^n-\boldsymbol{\omega}^*\|^2+\eta_t^2(\sum_{n=1}^{N}p_n^2\delta_n^2+6\ell\Phi+8\eta_t^2G^2(1-\frac{N_S(N_S-1)}{2N})\tag{32}$$

Proof. From (33), we bound bound three terms B_1, B_2 and B_3.

$$\begin{aligned}
&\mathbb{E}\left\|\boldsymbol{v}_{t+1}-\boldsymbol{\omega}^*\right\|^2=\mathbb{E}\left\|\boldsymbol{\omega}_t-\eta_t\boldsymbol{g}_t-\boldsymbol{\omega}^*\right\|^2=\mathbb{E}\left\|\boldsymbol{\omega}_t-\eta_t\boldsymbol{g}_t-\boldsymbol{\omega}^*-\eta_t\bar{\boldsymbol{g}}_t+\eta_t\bar{\boldsymbol{g}}_t\right\|^2\\
&=\underbrace{\mathbb{E}\left\|\boldsymbol{\omega}_t-\boldsymbol{\omega}^*-\eta_t\bar{\boldsymbol{g}}_t\right\|^2}_{B_1}+\eta_t^2\underbrace{\mathbb{E}\left\|\boldsymbol{g}_t-\bar{\boldsymbol{g}}_t\right\|^2}_{B_2}+\underbrace{2\mathbb{E}\left[\langle\boldsymbol{\omega}_t-\boldsymbol{\omega}^*-\eta_t\bar{\boldsymbol{g}}_t,\eta_t\boldsymbol{g}_t-\eta_t\bar{\boldsymbol{g}}_t\rangle\right]}_{B_3}
\end{aligned}\tag{33}$$

Note that $B_3=0$ because the $\mathbb{E}_{\mathcal{N}_t}[\langle\boldsymbol{\omega}_t-\boldsymbol{\omega}^*-\eta_t\bar{\boldsymbol{g}}_t,\eta_t\boldsymbol{g}_t-\eta_t\bar{\boldsymbol{g}}_t\rangle]=0$.

For B_1, we use the similar steps as in [24], [26] and get

$$\begin{aligned} B_1 &= \|\boldsymbol{\omega}_t - \boldsymbol{\omega}^* - \eta_t \bar{\boldsymbol{g}}_t\|^2 \\ &\leq (1-\eta_t\mu)\|\boldsymbol{\omega}_t^n - \boldsymbol{\omega}^*\|^2 + 2\sum_{n=1}^{N} p_n \|\boldsymbol{\omega}_t - \boldsymbol{\omega}_t^n\|^2 + 6\ell\eta_t^2\Phi \end{aligned} \tag{34}$$

For B_2, where $\mathbb{E}\,\|\boldsymbol{g}_t - \bar{\boldsymbol{g}}_t\|^2$ shows the variance of the stochastic gradients in UE u_n and it is bounded by δ_n^2, so it's bounded following the steps in (35)

$$\begin{aligned} \mathbb{E}\,\|\boldsymbol{g}_t - \bar{\boldsymbol{g}}_t\|^2 &= \mathbb{E}\left\|\sum_{n=1}^{N} p_n(\nabla L_n(\boldsymbol{\omega}_t^n, \xi_t^n) - \nabla L_n(\boldsymbol{\omega}_t^n))\right\|^2 \\ &= \sum_{n=1}^{N} p_n^2 \mathbb{E}\,\|(\nabla L_n(\boldsymbol{\omega}_t^n, \xi_t^n) - \nabla L_n(\boldsymbol{\omega}_t^n))\|^2 \\ &\leq \sum_{n=1}^{N} p_n^2 \delta_n^2 \end{aligned} \tag{35}$$

Now, we can bound $\mathbb{E}\,\|\bar{\boldsymbol{\upsilon}}_{t+1} - \boldsymbol{\omega}^*\|^2$ as

$$\begin{aligned} \mathbb{E}\,\|\bar{\boldsymbol{\upsilon}}_{t+1} - \boldsymbol{\omega}^*\|^2 &= \mathbb{E}B_1 + \eta_t^2\mathbb{E}B_2 + B_3 \\ &\leq (1-\eta_t\mu)\mathbb{E}\,\|\boldsymbol{\omega}_t^n - \boldsymbol{\omega}^*\|^2 + \underbrace{2\,\mathbb{E}\sum_{n=1}^{N} p_n\|\boldsymbol{\omega}_t - \boldsymbol{\omega}_t^n\|^2}_{F} + 6\ell\eta_t^2\Phi + \eta_t^2\sum_{n=1}^{N} p_n^2\delta_n^2 \end{aligned} \tag{36}$$

Further, F can be bounded following in (37)

$$\begin{aligned} F &= \mathbb{E}\sum_{n=1}^{N} p_n\|\boldsymbol{\omega}_t - \boldsymbol{\omega}_t^n\|^2 = \mathbb{E}\sum_{n=1}^{N} p_n\|(\boldsymbol{\omega}_t^n - \boldsymbol{\omega}_{t_0}) - (\boldsymbol{\omega}_t - \boldsymbol{\omega}_{t_0})\|^2 \\ &\leq \mathbb{E}\sum_{n=1}^{N} p_n\|(\boldsymbol{\omega}_t^n - \boldsymbol{\omega}_{t_0})\|^2 \\ &\leq \sum_{\tau=t_0}^{t-1}\sum_{n=1}^{N} p_n\mathbb{E}\|\eta_\tau \boldsymbol{g}_\tau^n\|^2 - \sum_{n\in\mathcal{N}_S} p_n\mathbb{E}\|\eta_\tau\Delta\boldsymbol{g}_\tau^{n_s}\|^2 \\ &\leq \sum_{\tau=t_0}^{t-1}(\mathbb{E}\|\eta_\tau\boldsymbol{g}_\tau^n\|^2 - (p_2\mathbb{E}\left\|\eta_\tau\boldsymbol{g}_\tau^1\right\|^2 + p_3\mathbb{E}\left\|\eta_\tau(\boldsymbol{g}_\tau^1 + \boldsymbol{g}_\tau^2)\right\|^2 + ... \\ &+ p_{N_S}\mathbb{E}\left\|\eta_\tau(\boldsymbol{g}_\tau^1 + \boldsymbol{g}_\tau^2, ..., +\boldsymbol{g}_\tau^{N_S-1})\right\|^2)) \\ &\leq 4\eta_t^2G^2(1 - \frac{N_S(N_S-1)}{2N}) \end{aligned} \tag{37}$$

Therefore, $\mathbb{E}A2$ is finally bounded by

$$\begin{aligned}\mathbb{E}A_2 &= \mathbb{E}\left\|\boldsymbol{\upsilon}^{t+1}-\boldsymbol{\omega}^*\right\|^2 \\ &\leq (1-\eta_t\mu)\|\boldsymbol{\omega}_t^n-\boldsymbol{\omega}^*\|^2+\eta_t^2(\sum_{n=1}^{N}p_n^2\delta_n^2+6\ell\Phi+8\eta_t^2G^2(1-\frac{N_S(N_S-1)}{2N})\end{aligned} \tag{38}$$

□

Based on the results of **Lemma** 1, 2 and 3, we can finally get

$$\mathbb{E}\left\|\boldsymbol{\omega}_{t+1}-\boldsymbol{\omega}^*\right\|^2 \leq (1-\eta_t\mu)\mathbb{E}\left\|\boldsymbol{\omega}_t-\boldsymbol{\omega}^*\right\|^2+\eta_t^2W \tag{39}$$

where $W=\sum\limits_{n=1}^{N}p_n^2\delta_n^2+6\ell\Phi+8\eta_t^2G^2(1-\frac{N_S(N_S-1)}{2N})+\frac{N-K}{K(N-1)}(4N-2N_S(N_S-1))G^2$

By defining $\Delta_{t+1}=\mathbb{E}\left\|\boldsymbol{\omega}_{t+1}-\boldsymbol{\omega}^*\right\|^2$, we have

$$\Delta_{t+1}\leq(1-\eta_t\mu)\Delta_t+\eta_t^2W \tag{40}$$

By setting $\Delta_t\leq\frac{\mathrm{v}}{\iota+t}$, $\eta_t=\frac{\varrho}{t+\iota}$ with $\varrho>\frac{1}{\mu}$ and $\iota>0$, this can be proved through induction method.

$$\begin{aligned}\Delta_{t+1} &\leq (1-\eta_t\mu)\Delta_t+\eta_t^2W \\ &\leq \left(1-\frac{\varrho\mu}{t+\iota}\right)\frac{\mathrm{v}}{t+\iota}+\frac{\varrho^2W}{(t+\iota)^2} \\ &= \frac{t+\iota-1}{(t+\iota)^2}\mathrm{v}+\left[\frac{\varrho^2W}{(t+\iota)^2}-\frac{\varrho\mu-1}{(t+\iota)^2}\mathrm{v}\right]\leq\frac{\mathrm{v}}{\iota+t+1}\end{aligned} \tag{41}$$

If we choose $\varrho=\frac{2}{\mu}$, then $\eta_t=\frac{2}{\mu}\frac{1}{t+\iota}$, so we have

$$\begin{aligned}\mathrm{v} &= \max\{\frac{\varrho^2W}{\varrho\mu-1},(\iota+1)\mathbb{E}\left\|\boldsymbol{\omega}_1-\boldsymbol{\omega}^*\right\|\} \\ &\leq \frac{\varrho^2W}{\varrho\mu-1}+(\iota+1)\mathbb{E}\left\|\boldsymbol{\omega}_1-\boldsymbol{\omega}^*\right\| \\ &\leq \frac{4W}{\mu^2}+(\iota+1)\mathbb{E}\left\|\boldsymbol{\omega}_1-\boldsymbol{\omega}^*\right\|\end{aligned} \tag{42}$$

Then by $\ell-smoothness$ of $L(.)$ and let $\kappa=\frac{\ell}{\mu}$,

$$\begin{aligned}\mathbb{E}[L(\boldsymbol{\omega}_t)]-L^* &\leq \frac{\ell}{2}\Delta_t\leq\frac{\ell}{2}\frac{\mathrm{v}}{t+\iota} \\ &\leq \frac{\kappa}{t+\iota}(\frac{2W}{\mu}+\frac{\mu(\iota+1)}{2}\mathbb{E}\left\|\boldsymbol{\omega}_1-\boldsymbol{\omega}^*\right\|)\end{aligned} \tag{43}$$

REFERENCES

[1] U. Challita, W. Saad, and C. Bettstetter, "Interference management for cellular-connected UAVs: A deep reinforcement learning approach," *IEEE Trans. Wireless Commun.*, vol. 18, no. 4, pp. 2125–2140, Mar. 2019.

[2] F. Tang, Y. Kawamoto, N. Kato, and J. Liu, "Future intelligent and secure vehicular network toward 6G: Machine-learning approaches," *Proceedings of the IEEE*, vol. 108, no. 2, pp. 292–307, Feb. 2020.

[3] B. Brik, A. Ksentini, and M. Bouaziz, "Federated learning for UAVs-enabled wireless networks: Use cases, challenges, and open problems," *IEEE Access*, vol. 8, pp. 53 841–53 849, Mar. 2020.

[4] Qualcomm, "We are making on-device AI ubiquitous." *Accessed:*, Dec. 2019. [Online]. Available: https://www.qualcomm.com/news/onq/2017/08/16/we-are-making-device-ai-ubiquitous

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

[6] O. Gupta and R. Raskar, "Distributed learning of deep neural network over multiple agents," *Journal of Network and Computer Applications*, vol. 116, pp. 1–8, Oct. 2018.

[7] P. Vepakomma, O. Gupta, T. Swedish, and R. Raskar, "Split learning for health: Distributed deep learning without sharing raw patient data," *arXiv preprint arXiv:1812.00564*, Dec. 2018.

[8] A. Singh, P. Vepakomma, O. Gupta, and R. Raskar, "Detailed comparison of communication efficiency of split learning and federated learning," *arXiv:1909.09145*, Sep. 2019.

[9] C. Thapa, M. A. P. Chamikara, and S. Camtepe, "Splitfed: When federated learning meets split learning," *arXiv preprint arXiv:2004.12088*, Aug. 2020.

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

[11] 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," *arXiv:1909.07972*, Oct. 2019.

[12] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *IEEE Int. Conf. on Commun. (ICC2019)*, Apr. 2019, pp. 1–7.

[13] J. Kang, Z. Xiong, D. Niyato, Y. Zou, Y. Zhang, and M. Guizani, "Reliable federated learning for mobile networks," *IEEE Wireless Commun.*, vol. 27, no. 2, pp. 72–80, Oct. 2020.

[14] M. M. Amiri, D. Gündüz, S. R. Kulkarni, and H. V. Poor, "Convergence of update aware device scheduling for federated learning at the wireless edge," *arXiv:2001.10402*, May 2020.

[15] J. Ren, Y. He, D. Wen, G. Yu, K. Huang, and D. Guo, "Scheduling for cellular federated edge learning with importance and channel awareness," *arXiv:2004.00490*, Jan. 2020.

[16] Y. J. Cho, S. Gupta, G. Joshi, and O. Yağan, "Bandit-based communication-efficient client selection strategies for federated learning," *arXiv:2012.08009*, Dec. 2020.

[17] N. Yoshida, T. Nishio, M. Morikura, and K. Yamamoto, "MAB-based client selection for federated learning with uncertain resources in mobile networks," *arXiv:2009.13879*, Sep. 2020.

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

[19] NVIDIA, "Nvidia jetson," *online: https://www.nvidia.com/pt-br/autonomous-machines/uavs-drones-technology/*, 2020.

[20] R. I. Bor-Yaliniz, A. El-Keyi, and H. Yanikomeroglu, "Efficient 3-D placement of an aerial base station in next generation cellular networks," in *IEEE Int.Conf. on Commun. (ICC2016)*, May 2016.

[21] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Mobile unmanned aerial vehicles (UAVs) for energy-efficient internet of things communications," *IEEE Trans. Wireless Commun.*, vol. 16, no. 11, pp. 7574–7589, Nov. 2017.

[22] J. Holis and P. Pechac, "Elevation dependent shadowing model for mobile communications via high altitude platforms in built-up areas," *IEEE Trans. on Ant. and Prop.*, vol. 56, no. 4, pp. 1078–1084, Apr. 2008.

[23] F. Haddadpour and M. Mahdavi, "On the convergence of local descent methods in federated learning," *arXiv:1910.14425*, Dec. 2019.

[24] M. M. Amiri, D. Gunduz, S. R. Kulkarni, and H. V. Poor, "Convergence of update aware device scheduling for federated learning at the wireless edge," *arXiv:2001.10402*, May 2020.

[25] A. Singh, P. Vepakomma, O. Gupta, and R. Raskar, "Detailed comparison of communication efficiency of split learning and federated learning," *arXiv preprint arXiv:1909.09145*, Sep. 2019.

[26] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on Non-IID data," *arXiv:1907.02189*, Jun. 2019.

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

[28] A. Garivier and E. Moulines, "On upper-confidence bound policies for non-stationary Bandit problems," *arXiv:0805.3415*, May 2008.

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

[30] V. Yilmaz, B. Konakoglu, C. Serifoglu, O. Gungor, and E. Gökalp, "Image classification-based ground filtering of point clouds extracted from uav-based aerial photos," *Geocarto international*, vol. 33, no. 3, pp. 310–320, 2018.

[31] T. Chowdhury, R. Murphy, and M. Rahnemoonfar, "Rescuenet: A high resolution uav semantic segmentation benchmark dataset for natural disaster damage assessment," *arXiv preprint arXiv:2202.12361*, Feb. 2022.

[32] T.-M. H. Hsu, H. Qi, and M. Brown, "Measuring the effects of non-identical data distribution for federated visual classification," *arXiv:1909.06335*, Sep. 2019.