S. Gan, K. Chen, J. Zhang, L. Xiang, D. W. K. Ng, and X. Ge, "Completion Time Minimization for Adaptive Semi-Asynchronous Federated Learning over Wireless Networks," accepted for presentation in *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC) - Workshops*, Valencia, Spain, Sep. 2024.

Completion Time Minimization for Adaptive Semi-Asynchronous Federated Learning over Wireless Networks

Shiyi Gan[1], Kui Chen[1], Jing Zhang[1], Lin Xiang[2], Derrick Wing Kwan Ng[3], and Xiaohu Ge[1]
[1]Huazhong Uni. Science and Technology [2]Technische Universität Darmstadt [3]Uni. New South Wales
Emails:{shiyigan,chenkui,zhangjing,xhge}@hust.edu.cn, l.xiang@nt.tu-darmstadt.de, w.k.ng@unsw.edu.au

***Abstract*—Federated learning (FL) over wireless networks offers a promising approach to enable decentralized machine learning among massive mobile edge nodes while ensuring privacy in training data. However, the convergence speed of FL is limited by the straggler effect, which arises from heterogeneous nodes, wireless fading channels, and non-independently and identically distributed (non-IID) training data. In this paper, we consider an adaptive semi-asynchronous FL to mitigate the straggler effect, by dynamically selecting subsets of nodes over time to synchronize the global model. We jointly optimize the node scheduling and computing/communication resource allocation to minimize the completion time required for convergence of the adaptive semi-asynchronous FL. Leveraging the convergence condition of semi-asynchronous FL, we further propose a greedy heuristic policy for node scheduling while tackling the remaining computing/communication resource allocation problem by exploiting a hidden convexity. Simulation results on open datasets demonstrate that, compared with existing FL algorithms, our proposed adaptive semi-asynchronous algorithm can significantly lower the latency of FL convergence.**

I. Introduction

Federated learning (FL) over wireless networks presents a distributed learning paradigm that enables multiple mobile edge nodes to collaboratively train a model, without sharing their raw data with other nodes or the central server [1]. This approach not only preserves privacy for the edge nodes, but also conserves communication resources, attracting substantial interest from both academia and industry [2], [3].

In the standard *synchronous* FL model, full synchronization between the parameter server and the participating nodes is required. The server waits for all nodes to complete their local training before aggregating these updates to refine the global model. However, the straggler effect poses significant challenges for implementing effective synchronous FL in wireless environments [4], [5]. Specifically, due to heterogeneous nodes, wireless fading channels, and non-independently and identically distributed (non-IID) training data, local models are sent to the parameter server at varying speeds. Consequently, under the synchronous FL setting, the server can only update the global model once it has received all local models.

To mitigate the straggler effect in FL, a novel method called *semi-asynchronous* FL was introduced in [6]. In this approach, the server updates the global model upon receiving local models from a predetermined number of nodes. This substantially shortens the time required for global model aggregation, thereby alleviating the straggler effect. Since [6], various node scheduling policies have been developed for semi-asynchronous FL, to enhance its performance in wireless networks. For example, FedSA was proposed in [7] to adaptively adjust the learning rate based on node participation frequency. Additionally, a priority function was introduced in [8] to select nodes based on data volume and computational capability, aiming to accelerate the convergence of semi-asynchronous FL. Furthermore, considering devices with unreliable network connections in [9], a cache-based latency tolerance mechanism was implemented by the aggregation server to improve round efficiency and enhance convergence.

Despite the fruitful development in [6]–[9], several unsolved issues continue to impede the efficacy of semi-asynchronous FL in wireless networks. One primary challenge involves determining the optimal number of nodes that should transmit their trained local models, while taking into account varying data distributions and edge heterogeneity. Typically, data collected from nodes exhibits distinct characteristics, resulting in different convergence rates when local models are trained on such data. As such, selecting a fixed number of nodes for model uploading may not promote the most effective convergence in semi-asynchronous FL. As another critical issue, the efficiency of global model aggregation can deteriorate when an excessive number of nodes upload their trained local models. In such scenarios, due to limited and heterogeneous radio resources at the wireless nodes, increasing the number of participating nodes can add to the latency in model uploading.

To address the challenges associated with semi-asynchronous FL in wireless networks, in this paper, we propose a novel node scheduling and resource allocation algorithm for minimizing completion time, namely the time required for semi-asynchronous FL to converge. Unlike existing works [7]–[9], our algorithm dynamically selects a subset of nodes per communication round, based on the volume of untrained data at each node, to implement global model aggregation. Given the node selection policy, we further jointly optimize the number of nodes for uploading trained local models and their order of scheduling and resource allocation to accelerate the convergence of semi-asynchronous FL under finite power and energy budget for each node. Our contributions are:

- We propose a novel node selection and resource allocation algorithm to minimize the completion time required for semi-asynchronous FL to converge. This algorithm utilizes a greedy heuristic policy for node selection and jointly optimizes the order of scheduling and computation/communication resources for the selected nodes.
- We reveal a hidden convexity underlying the resource allocation optimization and derive properties of the optimal scheduling order. Leveraging these results, we propose an iterative algorithm to solve the completion time minimization problem.

The work is supported by the National Natural Science Foundation of China (No.U2001210), the National Key Research and Development Program of China (No.2020YFB1806605). The work of L. Xiang has been funded by the LOEWE initiative (Hesse, Germany) within the emergenCITY center under grant LOEWE/1/12/519/03/ 05.001(0016)/72 and has been supported by the BMBF project Open6GHub under grant 16KISKO14.

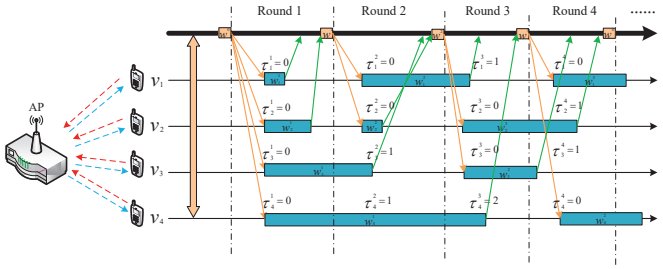

Fig. 1: Illustration of semi-asynchronous FL over a wireless network consisting of an AP and $M = 4$ nodes, with $N = 2$ nodes selected per round to upload locally trained models.

- Simulation results on open datasets demonstrate that, compared with existing FL algorithms, our proposed semi-asynchronous algorithm can significantly lower the completion time required for FL to converge.

In the remainder of this paper, Section II introduces the system model for semi-asynchronous FL over wireless. The completion time minimization problem is formulated and solved in Section III and Section IV, respectively. Simulation results are presented in Section V and finally, Section VI concludes the paper.

Notations: Throughout this paper, vectors and sets are denoted in lower-case boldface and upper-case calligraphic letters, respectively. $\|\mathbf{x}\|$ and $|\mathcal{X}|$ denote the norm of vector $\mathbf{x}$ and the cardinality of set $\mathcal{X}$, respectively. $\langle \mathbf{x}, \mathbf{y} \rangle$ denotes the inner product of vectors $\mathbf{x}$ and $\mathbf{y}$. $\lceil \cdot \rceil$ is the ceiling operator and $\mathbb{E}\{\cdot\}$ is the expectation operator. For FL, local models and loss functions are distinguished from their global counterparts by including the node indices in the subscripts. Specifically, $\mathbf{w}_i^k$ and $\mathbf{w}^k$ denote local model at node i and the global model in round k, respectively. Besides, $F_i(\mathbf{w}, \mathcal{D}_i)$ and $F(\mathbf{w})$ are the loss functions for local training at node i and the empirical risk function for global model updating. Finally, we adopt T to denote time duration, while t for time instance.

II. System Model

A. FL over Wireless Network

As shown in Fig. 1, we consider FL in a wireless network comprising an access point (AP) and M heterogeneous wireless nodes indexed by set $\mathcal{V} = \{1, 2, ..., i, ..., M\}$. The AP is equipped with a computing server and maintains bidirectional communications with the nodes over wireless links. Each node owns a local dataset $\mathcal{D}_i \triangleq \{(\mathbf{x}_1^i, y_1^i), ..., (\mathbf{x}_j^i, y_j^i), ..., (\mathbf{x}_{D_i}^i, y_{D_i}^i)\}$ with $D_i \triangleq |\mathcal{D}_i|$ samples. The jth sample in $\mathcal{D}_i$, denoted as $(\mathbf{x}_j^i, y_j^i)$, is composed by the training data $\mathbf{x}_j^i$ and label y_j^i. Without loss of generality, $\mathcal{D}_i, i \in \mathcal{V}$ are considered to be non-IID, with a total volume given by $D = \sum_{i \in \mathcal{V}} D_i$.

Let $\mathbf{w}$ be the global model parameter to be learned. Moreover, $f(\mathbf{w}, \mathbf{x}_j^i, y_j^i)$ is a loss function to measure the error or deviation between the training model on data $\mathbf{x}_j^i$ and the label y_j^i. The FL system aims to find the optimal parameter vector $\mathbf{w}^*$ that minimizes the empirical risk function defined by

$$F(\mathbf{w}) \triangleq \frac{1}{D} \sum_{i=1}^{M} \sum_{j=1}^{D_i} f(\mathbf{w}, \mathbf{x}_j^i, y_j^i), \tag{1}$$

without sharing the local datasets among nodes or with the AP, where

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} F(\mathbf{w}). \tag{2}$$

To this end, each node trains a local model leveraging its local dataset. The loss function for the local training at node i is defined as

$$F_i(\mathbf{w}, \mathcal{D}_i) \triangleq \frac{1}{D_i} \sum_{j=1}^{D_i} f(\mathbf{w}, \mathbf{x}_j^i, y_j^i), \quad i \in \mathcal{V}. \tag{3}$$

The model parameters obtained by the nodes are then uploaded to and aggregated at the AP in a global model aggregation phase. The latter aims to generate a new global model to minimize the empirical risk function $F(\mathbf{w})$. Based on (1) and (3), we have

$$F(\mathbf{w}) = \sum_{i=1}^{M} \frac{D_i}{D} F_i(\mathbf{w}, \mathcal{D}_i). \tag{4}$$

To guarantee the convergence for FL, we assume that the loss functions F_i, $i \in \mathcal{V}$ satisfy the following typical assumptions [10].

- *Assumption* 1 *(Smoothness):* F_i is L-smooth with $L > 0$, i.e., for all $\mathbf{w}_1, \mathbf{w}_2$, we have $F_i(\mathbf{w}_2, \mathcal{D}_i) - F_i(\mathbf{w}_1, \mathcal{D}_i) \leq \langle \nabla_{\mathbf{w}} F_i(\mathbf{w}_1, \mathcal{D}_i), \mathbf{w}_2 - \mathbf{w}_1 \rangle + \frac{L}{2} \| \mathbf{w}_2 - \mathbf{w}_1 \|^2$.
- *Assumption* 2 *(Strong Convexity):* F_i is μ-strongly convex with $\mu \geq 0$, i.e., we have $F_i(\mathbf{w}_2, \mathcal{D}_i) - F_i(\mathbf{w}_1, \mathcal{D}_i) \geq \langle \nabla_{\mathbf{w}} F_i(\mathbf{w}_1, \mathcal{D}_i), \mathbf{w}_2 - \mathbf{w}_1 \rangle + \frac{\mu}{2} \| \mathbf{w}_2 - \mathbf{w}_1 \|^2$, $\forall \mathbf{w}_1, \mathbf{w}_2$.

Here, L and μ are hyperparameters for the FL. Assumption 1 prevents drastic changes in the gradient $\nabla_{\mathbf{w}} F_i(\cdot, \cdot)$ to enhance the robustness of FL. Assumption 2 ensures that a global optimum exists for the loss function $F_i(\cdot)$.

B. Semi-Asynchronous FL

Due to heterogeneous nodes and fading wireless channels, the completion time of local model training may vary significantly across the edge nodes, slowing down the overall global model aggregation. This is known as the straggler effect and defines a performance bottleneck for FL. In this paper, we consider semi-asynchronous FL to mitigate the straggler effect, by dynamically selecting different subsets of nodes of a given size for implementing global model aggregation.

$\mathcal{K} = \{1, 2, ..., k, ..., K\}$ denotes the round set for FL training, and the system operates in multiple rounds indexed by k. At the beginning of round $k = 1$, the AP broadcasts the initial global model $\mathbf{w}^0$ to all nodes to start the local training at each node. At the end of each round $k \geq 1$, a subset of nodes, denoted by $\mathcal{V}^k \subseteq \mathcal{V}$ with size $|\mathcal{V}^k| = N \leq M$, are selected to upload their locally trained models to the AP for further global model aggregation. Let $L_{i,k} \in \{0, 1\}$ be a binary variable. We set $L_{i,k} = 1$ for each selected node $i \in \mathcal{V}^k$. We assume that each selected node removes or inactivates the trained data from its dataset after successfully uploading the trained local model. These selected nodes then receive the updated global model from the AP in round $k + 1$ and adopt it together with their remaining local data to start a new local training. On the other hand, we set $L_{i,k} = 0$ for node $i \in \mathcal{V} \backslash \mathcal{V}^k$, which continues to train its local models without uploading in round k. This process continues until the FL converges.

Due to node selection, staleness is incurred for edge nodes that have received the latest global model in round $k - 1$ but cannot upload their local models in round k, for $k > 1$. Let τ_i^k be the staleness of node $i \in \mathcal{V}$ in round k, which is an integer representing the number of rounds elapsed since it lastly received the global model. The updating rule for staleness is $\tau_i^{k+1} = (1 + \tau_i^k)(1 - L_{i,k})$. The AP records the staleness of each node. Fig. 1 illustrates the considered semi-asynchronous FL

with $M = 4$ and $N = 2$. For instance, after node v_1 receives the global model $\mathbf{w}^1$, it performs a local update, obtains the updated local model $\mathbf{w}_1^2$, and sends $\mathbf{w}_1^2$ to the AP in round 3. The resulting staleness of node 1 in round 3 is $\tau_1^3 = 3-1-1 = 1$.

When $\tau_i^k > 0$ in round k, node i has been continuing an ongoing local training for $(\tau_i^k + 1)$ rounds. Note that both k and τ_i^k increase at the same pace for all continuing rounds of local training at node i before uploading the trained model, such that the value of $k - \tau_i^k$ remains unchanged. To capture the impact of staleness in semi-synchronous FL, we denote the local training parameter of node i in round k by $\mathbf{w}_i^{k-\tau_i^k}$, which is updated according to

$$\mathbf{w}_i^{k-\tau_i^k} = \mathbf{w}^{k-\tau_i^k-1} - \eta_i \nabla_{\mathbf{w}} F_i \left(\mathbf{w}^{k-\tau_i^k-1}, \mathcal{D}_i\right). \quad (5)$$

In (5), a learning rate η_i is used for node i. In the special case of $\tau_i^k = 0$, i.e., when a selected node i receives the latest global model $\mathbf{w}^{k-1}$ in round k, it updates its local model in round k according to $\mathbf{w}_i^k = \mathbf{w}^{k-1} - \eta_i \nabla_{\mathbf{w}} F_i \left(\mathbf{w}^{k-1}, \mathcal{D}_i\right)$.

Upon receiving the local models from the selected N nodes, the AP updates the global model in round k according to

$$\mathbf{w}^k = \left(1 - \sum\nolimits_{i\in\mathcal{V}^k} \frac{D_i}{D}\right) \mathbf{w}^{k-1} + \sum\nolimits_{i\in\mathcal{V}^k} \frac{D_i}{D} \mathbf{w}_i^{k-\tau_i^k}, \quad (6)$$

where $\mathbf{w}_i^{k-\tau_i^k}$ is the local model uploaded from node i with a staleness of τ_i^k in round k. According to (6), the AP updates the global model $\mathbf{w}^k$ in round k through a convex combination of the received N local models $\mathbf{w}_i^{k-\tau_i^k}, i \in \mathcal{V}_k$ and the global model $\mathbf{w}^{k-1}$ in the previous round $k-1$, in order to decrease $F(\mathbf{w}^k)$ in (4) such that $F(\mathbf{w}^k) \leq F(\mathbf{w}^{k-1})$. The AP further transmits the updated global model $\mathbf{w}^k$ to the N selected nodes to continue FL, until $F(\mathbf{w}^k)$ obtains a minimum value, where the global model converges.

Note that a small N may result in large staleness at the edge nodes. In this case, the nodes would use significantly different global model parameters than the other nodes in local training, slowing down the convergence of semi-asynchronous FL. On the other hand, a large N may not effectively mitigate the straggler effect. Therefore, the value of N should be judiciously chosen to improve the convergence of semi-asynchronous FL.

C. Time and Energy Consumption of Edge Nodes

Let $T_{i,k}^b$ be the time (duration) required for broadcasting/multicasting the global model from the AP to the nodes. We assume that $T_{i,k}^b$ is constant to facilitate a tractable problem. Let d_i^k be the number of data samples trained at node i in round k. The time required for computing the local model at node i is given by

$$T_{i,k}^c = \frac{d_i^k C_i}{g_{i,k}}, \quad (7)$$

where $g_{i,k}$ represents the computing capability of node i in round k, measured in number of CPU cycles per second, and C_i denotes the number of CPU cycles required for processing one data sample at node i during local training. Note that $T_{i,k}^c$ is independent of the scheduling decision $L_{i,k}$. Meanwhile, the energy consumption for local computation at node i in round k is

$$E_{i,k}^c = \kappa_i d_i^k C_i g_{i,k}^2, \quad (8)$$

where κ_i is the effective switching capacitance of the CPU at node i [11].

Assume that the scheduled nodes upload their trained local models over a shared spectrum of B_u Hz utilizing time division multiple access (TDMA). When $L_{i,k} = 1$, the time consumed for uplink transmission from node i to the AP is given as

$$T_{i,k}^u = \frac{u_i}{r_{i,k}}, \quad (9)$$

where u_i and $r_{i,k}$ are the size of the local model parameters and the achievable data rate of node i, respectively. We consider block fading channels between the node i and the AP. Let $h_{i,u}$ be the instantaneous channel gain between node i and the AP. We assume that $h_{i,u}$ remains constant during updating in each round but may vary from one round to another. We have $r_{i,k} = B_u \log_2(1 + \frac{p_{i,k}|h_{i,u}|^2}{B_u N_0})$, where $p_{i,k}$ is the transmit power of node i and N_0 is the power spectral density of noise. Hence, the energy consumption of node i for the uplink transmission is given as

$$E_{i,k}^u = T_{i,k}^u p_{i,k} = \frac{p_{i,k} u_i}{B_u \log_2(1 + \frac{p_{i,k}|h_{i,u}|^2}{B_u N_0})}. \quad (10)$$

Under the TDMA scheme, nodes have to wait for spectrum access before they can upload their trained models. Let $T_{i,k}^w$ be the time that node i spends waiting to be scheduled in round k under TDMA. We have $T_{i,k}^w = t_{i,k}^u - (t_k^0 + T_{i,k}^b + T_{i,k}^c)$, where t_k^0 marks the beginning of round k, $t_{i,k}^u$ is the time instance when node i begins to upload its local model. Therefore, the time consumed by node i after K rounds is given as

$$T_i = \sum\nolimits_{k=1}^{K} (T_{i,k}^c + L_{i,k} T_{i,k}^u + L_{i,k} T_{i,k}^b + L_{i,k} T_{i,k}^w). \quad (11)$$

Meanwhile, the energy consumption of node i in computation and communication after K rounds is given as

$$E_i = \sum\nolimits_{k=1}^{K} (E_{i,k}^c + L_{i,k} E_{i,k}^u). \quad (12)$$

III. Optimization Problem Formulation

For deploying FL in wireless networks, the time required for achieving convergence in the global model is a key performance metric. However, an inherent trade-off between the convergence time and the incurred energy consumption exists in FL systems. As the edge nodes are usually powered by batteries, it is crucial to accelerate the FL without draining the batteries of the edge nodes. To strike an effective balance between both objectives, we jointly optimize the node selection and scheduling $\{N, L_{i,k}\}$ as well as the allocation of computing and communication resources at each node $\{g_{i,k}, p_{i,k}\}$. This approach aims to minimize the completion time for all nodes implementing FL, under finite power and energy budget for each node, while ensuring that the global model converges within K rounds. The resulting the optimization problem is formulated as

$$\begin{aligned}
\text{P1}: & \min_{N, L_{i,k}, g_{i,k}, p_{i,k}} \max_{i\in\mathcal{V}} \{T_i\} \\
\text{s.t. } & C_1: g_i^{\min} \leq g_{i,k} \leq g_i^{\max}, i \in \mathcal{V},\ k \in \mathcal{K}, \\
& C_2: 0 \leq p_i \leq p_{i,k}^{\max}, i \in \mathcal{V}, k \in \mathcal{K}, \\
& C_3: F\left(\mathbf{w}^K\right) \leq F\left(\mathbf{w}^*\right) + \varepsilon, \\
& C_4: N \in \{1, 2, ..., M\},
\end{aligned}$$

$$C_5: L_{i,k} \in \{0,1\}, \quad i \in \mathcal{V}, k \in \mathcal{K},$$
$$C_6: L_{i,k}E^u_{i,k} + E^c_{i,k} \leq E_b,\ i \in \mathcal{V}, k \in \mathcal{K}. \quad (13)$$

In problem P1, constraints C_1 and C_2 limit the computing capability and the maximum transmit power at node i, respectively. C_3 ensures the convergence of the global model after K rounds with a training accuracy of $\varepsilon \geq 0$. C_4 requires the number of scheduled nodes in each round, N, to be any positive integer not exceeding the total number of nodes. C_5 is the binary variable for scheduling the transmission of locally trained models in round k. Finally, C_6 denotes a total energy consumption constraint for computation and communication at each node in each round.

Note that P1 is a nonconvex mixed-integer nonlinear programming (MINLP) problem due to the integer optimization variables N and $L_{i,k}$ for node scheduling in constraints C_4 and C_5, and the nonconvex constraint C_6. Moreover, constraint C_3 is difficult to tackle due to the implicit dependence of function $F(\cdot)$ on parameters K and N and FL hyperparameters. This type of optimization problem is generally NP-hard [12]. To facilitate a real-time solution to problem P1, we analyze the properties of optimal scheduling policy in Section IV and further propose a suboptimal node scheduling and resource allocation algorithm based on convex optimization combined with a greedy-based heuristic approach.

IV. Proposed Solution

To tackle the aforementioned challenges for solving P1, Section IV-A first discusses the convergence condition of semi-asynchronous FL, cf. Lemma 1. Based on Lemma 1, we then propose in Section IV-B a greedy heuristic policy for node selection and transform the resulting computing/communication resource allocation problem into a convex problem for solution. Given these results, it remains to optimize the uploading order for the selected nodes, which is further tackled by deriving the optimality conditions in Lemmas 2 and 3 in Section IV-C.

A. Convergence of Semi-Asynchronous FL

To facilitate the analysis, we define $\widetilde{\eta} \triangleq \max_{i \in \mathcal{V}} \eta_i$ as the maximum learning rate among nodes and $\widetilde{\eta}_{\max} \triangleq \max_{i \in \mathcal{V}_k} \eta_i$ is the maximum learning rate among the chosen nodes. Moreover, $\widetilde{\beta} \triangleq \min_{v_i \in \mathcal{V}}\{\beta_i\}$ is the minimum proportion of trained data among all nodes, relative to the total data volume and $\widetilde{\beta}_{\min} \triangleq \min_{k=1,\ldots,K}\{\sum_{i \in \mathcal{V}_k} \beta_i\}$ denotes the minimum portion of data trained by the chosen nodes per round over K rounds, relative to the total amount of data, where $\beta_i \triangleq D_i/D$. Furthermore, $\tau_{\max} = \max_{\substack{k=1,\ldots,K \\ i \in \mathcal{V}}} \tau_i^k$ represents the maximum staleness for nodes throughout the training process.

Lemma 1. *If $\widetilde{\eta} < \mu/L^2$, $\lambda < [2N\widetilde{\beta}_{\min}(\mu - \widetilde{\eta}L^2)/M]^{-1}$ and $\alpha\lambda\widetilde{\beta}_{\min}\left(\mu - \widetilde{\eta}L^2\right) \in (0, \frac{1}{2})$, after K rounds of global model aggregation, the obtained global model $\mathbf{w}^k$ satisfies*

$$\mathbb{E}\left\{F(\mathbf{w}^K)\right\} - F(\mathbf{w}^*) \leq \rho^K\left(F(\mathbf{w}^0) - F(\mathbf{w}^*)\right) + \delta, \quad (14)$$

where $\mathbb{E}\{F(\mathbf{w}^K)\}$ is the expected risk function after K communication rounds. $\rho = [1 - 2N\widetilde{\beta}_{\min}(\mu - \widetilde{\eta}L^2)/M]^{\frac{1}{1+\tau_{\max}}} < 1$ is the convergence factor and characterizes the convergence rate of the loss function in one round. $\delta = \frac{\widetilde{\eta}L}{2\widetilde{\beta}_{\min}(\mu - \widetilde{\eta}L^2)} \sum_{i \in \mathcal{V}} \beta_i |F_i(\mathbf{w}^*)|^2$ is the residual error.

Proof: The proof follows from [7, Theorem 1] and is ignored here due to page limitation. ■

According to Lemma 1, the global model is guaranteed to converge after K rounds, i.e., satisfying constraint C_3, provided $\mathbb{E}\left\{F\left(\mathbf{w}^K\right)\right\} - F\left(\mathbf{w}^*\right) \leq \rho^K(F(\mathbf{w}^0) - F\left(\mathbf{w}^*\right)) + \delta_{\max} \leq \varepsilon$, where $\delta_{\max} = \frac{\widetilde{\eta}_{\max}L}{2\widetilde{\beta}(\mu - \widetilde{\eta}_{\max}L^2)} \sum_{i \in \mathcal{V}} \beta_i |F_i(\mathbf{w}^*)|^2$ stands for the maximum residual error. Therefore, we rewrite C_3 as

$$\widetilde{C_3}: K \geq (1 + \tau_{\max}) \frac{\log_2 \phi}{\log_2(1 - 2\frac{N}{M}\lambda\widetilde{\beta}_{\min}\left(\mu - \widetilde{\eta}L^2\right))}, \quad (15)$$

where $\phi = \frac{\varepsilon - \delta_{\max}}{F(\mathbf{w}^0) - F(\mathbf{w}^*)}$ represents the target training accuracy relative to the suboptimality of the starting point, namely $F(\mathbf{w}^0) - F(\mathbf{w}^*)$. $\widetilde{C_3}$ ensures convergence in the global model within K rounds, regardless of how the nodes are scheduled for model uploading.

Note that in (15), ϕ, λ, M, μ, and L are all constants, whose values are independent of the optimization variables in problem P1. In contrast, N and $\widetilde{\beta}$ still depend on the scheduling policy of the nodes in K rounds, which are coupled with each other and also with constraints C_1, C_2, and C_6 for resource allocation. To facilitate the solution, in Section IV-B, we consider a greedy-based heuristic policy for node scheduling, which can decouple the node selection from the resource allocation.

B. Greedy Heuristic Policy for Selection of Uploading Nodes

For a given N, we select the N edge nodes having the most amount of untrained data to upload their trained local models. Given the selected uploading nodes, we further show below that the resulting resource allocation problem can be transformed into a convex problem and solved with convenience. Finally, the value of N is optimized via a one-dimensional search over set $\{1, \ldots, M\}$.

Let $D_{i,k}$ be the amount of data remaining at node i at the begining of round k, where $D_{i,1} = D_i, i \in \mathcal{V}$. After node i uploads its local model in round k, the d_i^k amount of trained data is removed/inactivated from node i. We thus have

$$D_{i,k+1} = D_{i,k} - d_i^k. \quad (16)$$

Arranging $D_{i,k}$, $i \in \mathcal{V}$, in a descending order, the first N nodes are scheduled to upload their local models.

For the selected N nodes, we have $L^*_{i,k} = 1$. Otherwise, $L^*_{i,k} = 0$. Based on this node selection, P1 reduces to

$$\begin{aligned} \mathrm{P2}: & \min_{g_{i,k}, p_{i,k}} \max_{i \in \mathcal{V}_k} \{T_{i,k}\} \\ \text{s.t. } & C_1, C_2, \\ & \widetilde{C_6}: L^*_{i,k}E^u_{i,k} + \kappa_i d_i^k C_i g^2_{i,k} \leq E_b, \quad k \in \mathcal{K}, \end{aligned} \quad (17)$$

where $T_{i,k} = \frac{d_i^k C_i}{g_{i,k}} + \frac{L^*_{i,k}u_i}{B_u \log_2(1 + \frac{p_{i,k}|h_{i,u}|^2}{B_u N_0})} + L^*_{i,k}T^b_{i,k} + L^*_{i,k}T^w_{i,k}$. Problem P2 is still nonconvex due to the nonconvex objective function and the nonconvex constraint $\widetilde{C_6}$. However, unlike P1, problem P2 exhibits a hidden convexity, which can be exploited to obtain its global optimal solution. To unveil this, let us introduce auxiliary variables ω_k and χ_k and rewrite problem P2 equivalently as

$$\begin{aligned} \mathrm{P3}: & \min_{g_{i,k}, p_{i,k}, \omega_k, \chi_k} \max_{i \in \mathcal{V}_k} \left(\frac{d_i^k C_i}{g_{i,k}} + L^*_{i,k}\omega_k + L^*_{i,k}T^b_{i,k}\right) \\ \text{s.t. } & C_1, C_2, \end{aligned}$$

$$\widetilde{C_6}: L^*_{i,k}\chi_k + \kappa_i d_i^k C_i g_{i,k}^2 \le E_b, k \in \mathcal{K},$$
$$C_7: \frac{p_{i,k}^2 u_i}{\chi_k} - p_{i,k} B_u \log_2(1 + \frac{p_{i,k}|h_{i,u}|^2}{B_u N_0}) \le 0, k \in \mathcal{K},$$
$$C_8: \frac{u_i}{\omega_k} - B_u \log_2(1 + \frac{p_{i,k}|h_{i,u}|^2}{B_u N_0}) \le 0, k \in \mathcal{K},$$
$$C_9: \omega_k > 0, k \in \mathcal{K},$$
$$C_{10}: \chi_k > 0, k \in \mathcal{K}. \quad (18)$$

Here, $T^w_{i,k}$ is omitted in the objective function of P3, because it depends only on the scheduling order of selected nodes in TDMA, rather than $g_{i,k}$ and $p_{i,k}$. As P3 is a convex problem, it can be solved using off-the-shelf tools such as CVX [13].

C. Optimization of Scheduling Order in Each Round

As discussed in Section IV-B, the scheduling order of computing and communication operations for the selected nodes significantly affects the waiting time $T^w_{i,k}$ in the objective function of P2 or P1. To minimize the overall completion time, it remains to optimize the scheduling order for each individual round. In the following discussions, we consider an arbitrary round and for brevity, omit the round index k. For example, the beginning of the round is denoted by t^0.

As the time required for AP broadcasting, T^b_i, is independent of node scheduling, we only need to schedule the data computation and uploading communication for the N selected nodes in each round. This involves a set of $2N$ operations denoted by $\mathcal{O} \triangleq \{O^c_1, O^c_2, \ldots, O^c_N, O^u_1, O^u_2, \ldots, O^u_N\}$, where O^c_i and O^u_i denote the computation and communication operations for node i, to be completed within duration T^c_i and T^u_i, respectively. A scheduling policy Π maps $\mathcal{O}$ into an operation order, i.e., a series of computation and communication operations for all nodes. Here, O^u_i has to follow O^c_i for each node i in order to guarantee causality.

Let t^j_i be the time instance at which node i completes operation O^j_i, $j \in \{c, u\}$. We have

$$t^c_i = t^0 + T^b_i + T^c_i, \quad (19)$$

since a node can execute data computation immediately after receiving the global model, in parallel with other nodes. In contrast, the node can execute uploading communication only after it finishes data computation and other nodes scheduled before it finish uploading communication. Thus, let $\mathcal{I}(i)$ be the set of nodes scheduled for uploading communication before node i. The time instance for completing uploading communication by $\mathcal{I}(i)$ is given as $t^u_{\mathcal{I}(i)} \triangleq \max_{s\in\mathcal{I}(i)}\{t^u_s\}$. Consequently, we have

$$t^u_i = \max\{t^u_{\mathcal{I}(i)}, t^c_i\} + T^u_i. \quad (20)$$

The scheduling policy Π impacts the time needed for all nodes to execute FL, which in turn affects the convergence time of the global model. Let Π^* be the optimal scheduling policy that minimizes the completion time for the round under consideration. We have the following lemmas.

Lemma 2. *The optimal scheduling policy Π^* should implement N data computation operations followed by N upload communication operations.*

Proof: Recall that O^u_i must follow O^c_i for each node i. Any policy other than Π^* can be obtained by interchanging two operations in Π^* or performing such interchange for a finite number of times. In this case, the new scheduling policy will ensure that at least one node undergoes communication operation before another node begins its computation operation. Without loss of generality, we consider scheduling policy $\Pi' = \{\Pi_1, O^u_m, O^c_n, \Pi_2\}$, obtained by interchanging operations O^c_n and O^u_m in Π^*. Consequently, we rewrite the optimal scheduling policy as $\Pi^* = \{\Pi_1, O^c_n, O^u_m, \Pi_2\}$, where Π_1 and Π_2 denote the other operations for Π' and Π^*.

With policy Π', the time instances of completing O^u_m and O^c_n are given by $t^u_m = \max\{t^u_{s\in\mathcal{I}(m)}, t^c_m\} + T^u_m$ and $t^c_n = \max\{t^u_{s\in\mathcal{I}(m)}, t^c_m\} + T^u_m + T^c_n$, respectively. Since node m can perform local training while node n implementing uploading communication, we can rewrite t^c_n as $t^c_n = \max\{t^u_{s\in\mathcal{I}(m)} + T^c_n, t^c_m + T^c_n\} + T^u_m$. Meanwhile, with policy Π^*, the time instances of completing O^c_n and O^u_m are $t^c_n = t^0 + T^c_n$ and $t^u_m = \max\{t^0 + T^c_n, t^c_m\} + T^u_m = \max\{t^c_n, t^c_m\} + T^u_m$, respectively.

We have $\max\{t^u_{s\in\mathcal{I}(m)} + T^c_n, t^c_m + T^c_n\} + T^u_m > \max\{t^c_n, t^c_m\} + T^u_m$. Thus, the operations before Π_2 in Π' are completed later than those in Π^*, and hence, Π' cannot do better than Π^*. Similarly, we can show that other policies are also suboptimal, which completes the proof. ■

Lemma 3. *Assume that the selected nodes complete local training at time instances sorted in an increasing order as $t^c_{\pi(1)} \le t^c_{\pi(2)} \le \ldots \le t^c_{\pi(N)}$, where $\pi(n)$ is the node index in the nth order. Then, the optimal scheduling policy Π^* has to schedule the uploading operations for the selected nodes according to $O^u_{\pi(1)}, O^u_{\pi(2)}, \ldots, O^u_{\pi(N)}$ sequentially.*

Proof: Without loss of generality, assume that nodes m and n satisfy $t^c_m \le t^c_n$. Other than the optimal policy $\Pi^* = \{\Pi_1, O^u_{\pi(m)}, O^u_{\pi(n)}, \Pi_2\}$, let us consider a scheduling policy defined as $\overline{\Pi} = \{\Pi_1, O^u_{\pi(n)}, O^u_{\pi(m)}, \Pi_2\}$. The time instance of completing $O^u_{\pi(n)}$ in Π^* is $t^u_n = \max\{t^c_m + T^u_m, t^c_n\} + T^u_n$. Meanwhile, the time instance of completing $O^u_{\pi(m)}$ in $\overline{\Pi}$ satisfies $t^u_m = \max\{t^c_n + T^u_n, t^c_m\} + T^u_m = t^c_n + T^u_n + T^u_m$, where the second equality is due to $t^c_n \ge t^c_m$, and $T^u_n > 0$.

Now, let us compare t^u_n in Π^* and t^u_m in $\overline{\Pi}$. When $t^c_m + T^u_m > t^c_n$, i.e., $\max\{t^c_m + T^u_m, t^c_n\} = t^c_m + T^u_m$, we have $t^u_n = t^c_m + T^u_m + T^u_n \le t^c_n + T^u_n + T^u_m$ for Π^*, due to $t^c_m \le t^c_n$. Note that $t^c_n + T^u_n + T^u_m$ is the time instance of completing $O^u_{\pi(m)}$ in $\overline{\Pi}$. Thus, t^u_n in Π^* is smaller than t^u_m in $\overline{\Pi}$. On the other hand, when $t^c_m + T^u_m < t^c_n$, i.e., $\max\{t^c_m + T^u_m, t^c_n\} = t^c_n$, we have $t^u_n = t^c_n + T^u_n \le t^c_n + T^u_n + T^u_m$ for Π^*, since $T^u_m > 0$ always holds. Again, t^u_n in Π^* is smaller than t^u_m in $\overline{\Pi}$.

Meanwhile, the time instances of completing the operations before Π_2 in $\overline{\Pi}$ is larger than that in Π^*. Therefore, $\overline{\Pi}$ is suboptimal, which completes the proof. ■

The optimal order for scheduling the local model training and uploading communication of the selected nodes can be thus determined according to Lemmas 2 and 3. The overall greedy node scheduling procedure and resource allocation is outlined in Algorithm 1.

V. Simulation Results

In this section, we evaluate the performance of the proposed scheme via experiments. We consider a U-class classification task with hypothesis space $\mathcal{C} = \{c_1, c_2, \cdots, c_u, \cdots, c_U\}$, and we set the amount of data on node i with class c_u is D_{i,c_u}. The loss function at node i is defined as [14]

$$F_i(\mathbf{w}, \mathcal{D}_i) = \sum\nolimits_{c_u \in \mathcal{C}} -\frac{D_{i,c_u}}{D_i} \mathbb{E}_{\mathbf{x}|y=c_u}[\log p_u(\mathbf{x}, \mathbf{w})], \quad (21)$$

Algorithm 1 Proposed Joint Node Scheduling and Resource Allocation Optimization Algorithm

Initialization : Set the amount of data remaining for node i in round k is $D_{i,k}$, and $D_{i,1} = D_i, i \in \mathcal{V}$. Set $\tau_i^1 = 0, i \in \mathcal{V}$.
for $k = \{1, ..., K\}$ **do**
  $\mathcal{V}^k = \emptyset$
  while card$(\mathcal{V}^k) < N$ **do**
    $D_j = \max_{i \in \mathcal{V}} \{D_{i,k}\}$
    $\mathcal{V} = \mathcal{V} \backslash \{j\}$
    $\mathcal{V}^k = \mathcal{V}^k \cup \{j\}$
  end while
  for $i = \{1, ..., M\}$ **do**
    if $i \in \mathcal{V}^k$ **then**
      Set $L_{i,k} = 1$.
    else
      Set $L_{i,k} = 0$.
    end if
  end for
  For given $L_{i,k}$, optimize resource allocation $\{p'_{i,k}, g'_{i,k}\}$ by solving problem P3 in CVX.
  For given $\{p'_{i,k}, g'_{i,k}\}$, obtain $t^c_{i,k}$ and $t^u_{i,k}$ by (7) and (9), respectively.
  According to Lemma 2, all chosen nodes are scheduled to train local model firstly.
  Based on Lemma 3, chosen nodes upload their trained local model according to the ascending of $t^c_{i,k}$.
  if $L_{i,k} = 1$ **then**
    $D_{i,k+1} = D_{i,k} - d_i^k$
  end if
  $\tau_i^{k+1} = (1 + \tau_i^k)(1 - L_{i,k})$.
end for

where $p_u(\mathbf{x}, \mathbf{w})$ is the probability that model $\mathbf{w}$ predicts that input sample $\mathbf{x}$ belongs to label c_u. In this manner, the empirical risk function of the global model is expressed as $F(\mathbf{w}) = \sum_{c_u \in \mathcal{C}} -\frac{\sum_{i \in \mathcal{V}} D_{i,c_u}}{D} \mathbb{E}_{\mathbf{x}|y=c_u}[\log p_u(\mathbf{x}, \mathbf{w})]$.

In the experiments, we evaluate the proposed scheme using the MNIST dataset [15], which contains 7000 samples of 10 kinds of 28×28 greyscale images, and the CIFAR-10 dataset [16], which contains 10 classes 6000 samples of 32×32 color images. We implement semi-asynchronous FL on a 6-core 12-th Generation Intel(R) Core(TM) i5-12400 CPU, to simulate FL between one AP and $M = 10$ nodes. In order to analyze the training performance of non-IID data, all data in the dataset is divided into two groups based on the parity of the labels of the data. Then the data in the odd group are equally distributed to nodes 1-5, and the data in the even group are equally distributed to nodes 6-10.

Each node processes a local dataset and uses it for local training. The local trained models are further uploaded to the AP for aggregation, thus realizing the simulation environment of the FL. To realize the proposed semi-synchronous FL scheme, we choose the number of nodes to be uploaded in each round. Unless otherwise specified, the simulation parameters

TABLE I: Simulation Parameter Settings [17].

Target training accuracy, ϕ	0.05
Number of CPU cycles, C_i	$[1, 3] \times 10^4$ cycle/sample
Spectrum bandwidth of AP, B_u	160 MHz
Noise power density, N_0	-170 dBm/Hz
Maximum energy consumption, E_b	$500J$
Global learning rate, λ	0.01
Maximum transmit power, $P_{ue}^{\max}$	1 W
Maximum CPU Frequency, g_i^{max}	4×10^9cycle/s
Minimum CPU Frequency, g_i^{min}	1×10^9cycle/s
Size of trained model parameters, u_i	5 KB
Effective switch capacitance, κ_i	10^{-28}

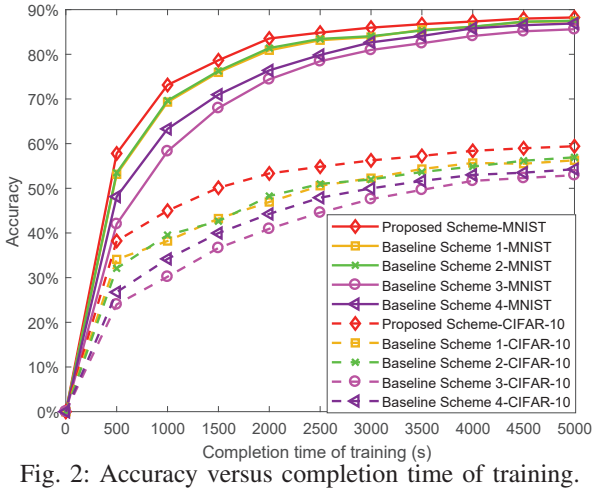

Fig. 2: Accuracy versus completion time of training.

are set according to Table I. For performance comparison, we evaluate four benchmark schemes.

- *Baseline Scheme 1 [18]:* A fixed scheduling policy is adopted, where the local models of the first $\lceil N/2 \rceil$ nodes and the remaining $N - \lceil N/2 \rceil$ nodes are alternately uploaded, for which the staleness does not exceed 1.
- *Baseline Scheme 2 [19]:* N nodes are randomly selected to upload their local models.
- *Baseline Scheme 3:* A fully asynchronous FL is adopted which selects only one node per round to upload local model.
- *Baseline Scheme 4:* All nodes have to upload their local models for synchronized FL in each round.

We compare the benchmarks with our proposed algorithm by training the models for given amount of time on different datasets. Fig. 2 and Fig. 3 show the accuracy and the loss function value of the considered schemes versus total training time under non-IID training data. We observe that semi-asynchronous FL achieves the highest learning accuracy and the smallest loss comparing for various baseline schemes in both the MNIST and the CIFAR-10 datasets. This is because when $N = 1$ in baseline 3, asynchronous FL mechanism is utilized for global model updates. This scheme leads to a large staleness of nodes, which slows down the convergence of FL. When $N = M = 10$ in baseline 4, synchronization mechanism is utilized for global model update and the straggler effect degrades the convergence rate and accuracy. Unlike baselines 1 and 2 that employ fixed and random nodes selection, respectively, the greedy heuristic node selection scheme can

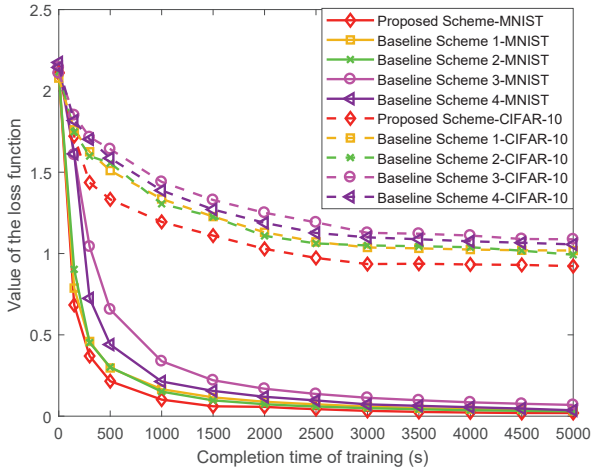

Fig. 3: Value of the loss function versus completion time of training.

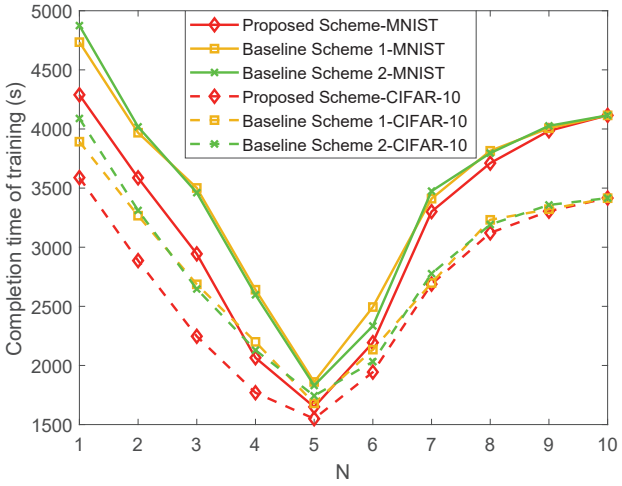

Fig. 4: Completion time of training versus N.

choose the nodes having maximal remaining data sizes and hence, the learning rates of nodes are improved. Combining with the node scheduling policy in Lemma 2 and 3, the convergence rate and accuracy of semi-asynchronous FL can be improved. Besides, we also note that the training accuracy of all schemes for CIFAR-10 dataset is lower than that for MNIST, due to it has higher noise.

Fig. 4 shows the completion time of training versus the number of nodes N uploading the local model in each round for different optimization schemes. Without loss of generality, we set the target training accuracy $\phi = 0.05$ in our experiments. Notably, the completion time of training initially decreases and then increases with N for all considered scenarios. This is because the semi-asynchronous FL allows unscheduled nodes to train their local models concurrently while other nodes are uploading their models. When $N = 5$, the two baseline schemes and our proposed algorithm can achieve minimal completion time. However, the completion time of our proposed algorithm is smaller than the baseline schemes thanks to the adopted greedy heuristic node selection scheme and node scheduling policy based on Lemmas 2 and 3. This result shows that intelligent node selection and scheduling can significantly affect the completion time of semi-asynchronous FL.

VI. Conclusion

This paper investigated a novel semi-asynchronous FL resource allocation framework including node selection, node scheduling, and resource allocation to minimize the completion time of the semi-asynchronous FL. The framework took into consideration the presence of heterogeneous nodes, wireless fading channels, and non-IID training data. To reduce the training time for semi-asynchronous FL to reach convergence, we proposed a greedy heuristic policy for node scheduling. This approach allowed us to effectively solve the nonconvex resource allocation problem. Additionally, we revealed the properties of node scheduling in each round to determine the optimal scheduling order. Simulation results demonstrated that our proposed algorithm reduces the time duration of all nodes implementing FL, ensuring faster convergence in the global model when compared to the four baseline schemes.

References

[1] X. Liu, Y. Deng *et al.*, "Federated learning and meta learning: Approaches, applications, and directions," *IEEE Commun. Surveys Tuts.*, vol. 26, no. 1, pp. 571–618, 2024.

[2] S. Hu, X. Chen *et al.*, "Distributed machine learning for wireless communication networks: Techniques, architectures, and applications," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 3, pp. 1458–1493, 2021.

[3] S. Niknam, H. S. Dhillon *et al.*, "Federated learning for wireless communications: Motivation, opportunities, and challenges," *IEEE Commun. Mag.*, vol. 58, no. 6, pp. 46–51, 2020.

[4] H.-S. Lee and J.-W. Lee, "Adaptive transmission scheduling in wireless networks for asynchronous federated learning," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3673–3687, 2021.

[5] F. Gauthier, V. C. Gogineni *et al.*, "Asynchronous online federated learning with reduced communication requirements," *IEEE Internet Things J.*, vol. 10, no. 23, pp. 20 761–20 775, 2023.

[6] J. Zhang, W. Liu *et al.*, "Semi-asynchronous model design for federated learning in mobile edge networks," *IEEE Trans. Veh. Technol.*, vol. 72, no. 12, pp. 16 280–16 292, 2023.

[7] Q. Ma, Y. Xu *et al.*, "Fedsa: A semi-asynchronous federated learning mechanism in heterogeneous edge computing," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3654–3672, 2021.

[8] Q. Zhuohao, M. Firdaus *et al.*, "A blockchain-based auditable semi-asynchronous federated learning for heterogeneous clients," *IEEE Access*, vol. 11, pp. 133 394–133 412, 2023.

[9] W. Wu, L. He *et al.*, "SAFA: A semi-asynchronous protocol for fast federated learning with low overhead," *IEEE Trans. Computers*, vol. 70, no. 5, pp. 655–668, 2021.

[10] M. Kim, A. L. Swindlehurst *et al.*, "Beamforming vector design and device selection in over-the-air federated learning," *IEEE Trans. Wireless Commun.*, vol. 22, no. 11, pp. 7464–7477, 2023.

[11] Z. Yang, M. Chen *et al.*, "Energy efficient federated learning over wireless communication networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1935–1949, 2021.

[12] A. Farajzadeh, A. Yadav *et al.*, "FLSTRA: Federated learning in stratosphere," *IEEE Trans. Wireless Commun.*, vol. 23, no. 2, pp. 1052–1067, 2024.

[13] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," 2014.

[14] J. Tian, P.-W. Tsai *et al.*, "Synergetic focal loss for imbalanced classification in federated xgboost," *IEEE Trans. Artificial Intelligence*, vol. 5, no. 2, pp. 647–660, 2024.

[15] Y. Saadna, A. Behloul *et al.*, "Speed limit sign detection and recognition system using svm and mnist datasets," *Neural Computing and Applications*, vol. 31, no. 9, pp. 5005–5015, 2019.

[16] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[17] Y. Mao, J. Zhang *et al.*, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3590–3605, 2016.

[18] C. You, D. Feng *et al.*, "Semi-synchronous personalized federated learning over mobile edge networks," *IEEE Trans. Wireless Commun.*, vol. 22, no. 4, pp. 2262–2277, 2023.

[19] S. Lee, A. K. Sahu *et al.*, "Partial model averaging in federated learning: Performance guarantees and benefits," *Neurocomputing*, vol. 556, p. 126647, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231223007701