This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3425757

Client Selection for Wireless Federated Learning with Data and Latency Heterogeneity

Xiaobing Chen, Xiangwei Zhou, Hongchao Zhang, Mingxuan Sun, and H. Vincent Poor

***Abstract*—Federated learning is a distributed machine learning paradigm that allows multiple edge devices to collaboratively train a shared model without exchanging raw data. However, the training efficiency of federated learning is highly dependent on client selection. Moreover, due to the varying wireless communication environments and various computation latencies among clients, selecting clients randomly or uniformly may not be optimal for balancing data diversity and training efficiency. In this paper, we formulate a new latency-minimization problem that simultaneously optimizes client selection and training procedures in federated learning, which takes into account the data and latency heterogeneity among clients. Given the non-convexity of the problem, we derive a new convergence upper bound for federated learning with probabilistic client selection. To solve the mixed integer nonlinear programming problem, we introduce a hybrid solution that integrates grid search techniques with the polyhedral active set algorithm. Numerical analyses and experiments on real-world data demonstrate that our scheme outperforms existing ones in terms of overall training latency and achieves up to 3 times acceleration over random client selection, especially in scenarios with highly heterogeneous data and latencies among clients.**

***Index Terms*—Federated learning, client selection, optimization, data heterogeneity, latency heterogeneity.**

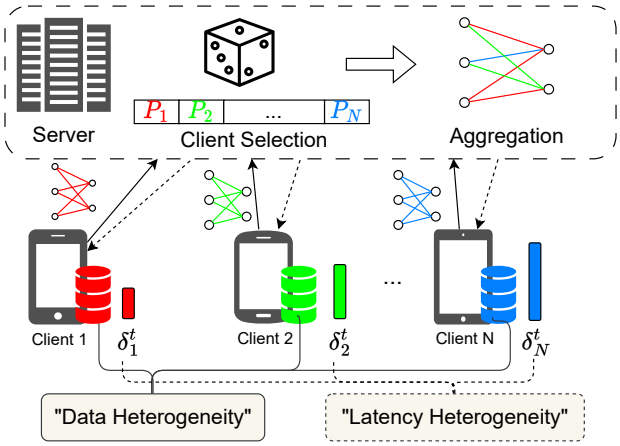


Fig. 1: Data and latency heterogeneity in federated learning. Clients exhibit diverse data qualities and experience various latencies. In each round, participants are selected according to the probability $\{p_i\}$. δ_i^t denotes the latency of the i-th client in the t-th round.

I. Introduction

The ubiquitous presence of edge devices, such as mobile phones and Internet of Things (IoT) sensors, is introducing new paradigms for collaborative machine learning, among which federated learning is gaining significant attention. This approach is particularly appealing in edge networks because of its ability to facilitate decentralized learning over edge devices while helping preserve the data privacy of these devices.

In a typical federated learning setup, selected devices train models on their local data and share model updates with a central server that aggregates these updates to improve a global model. However, environments such as IoT and edge networks are characterized by significant data and latency heterogeneity among devices. Therefore, the efficiency of the training process in federated learning faces several challenges [1], such as high communication costs between clients and the server, data heterogeneity among clients [2], and substantial differences in the response times of clients, as shown in Figure 1.

The overall training time of federated learning is determined by two factors: *the number of global rounds required for model convergence* and *the time cost for each training round*. To reduce the overall training time of federated learning, recent studies have explored various strategies, including model compression [3–9], model convergence acceleration [10–12], and judicious participant selection [13–15]. Given the server's limited communication resources and the imperative of maintaining efficient training, only a subset of clients are selected to participate in each round of training. Hence, the client selection strategies hold a critical role in determining both the performance of the model and the efficiency of the training process. On one hand, the diversity and quality of the data of selected clients are crucial to the generalizability of the global model. On the other hand, the various training latencies of the clients should be taken into account to improve the training efficiency of the federated learning system.

Given the data heterogeneity among clients, a straightforward strategy to facilitate faster model convergence is to

This paper was supported in part by the National Science Foundation under Grants 1943486, 2110722, 2246757, 2309549, and 2332011. *(Corresponding author: Xiangwei Zhou.)*

X. Chen and X. Zhou are with the Division of Electrical and Computer Engineering, Louisiana State University, Baton Rouge, LA, 70803, USA (e-mail: {xchen87, xwzhou}@lsu.edu).

H. Zhang is with the Department of Mathematics, Louisiana State University, Baton Rouge, LA, 70803, USA (e-mail: zhc@lsu.edu).

M. Sun is with the Division of Computer Science and Engineering, Louisiana State University, Baton Rouge, LA, 70803, USA (e-mail: msun11@lsu.edu).

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

prioritize participants who have high-quality data and can substantially influence model training. Existing studies have widely adopted importance sampling techniques, in which participant selection dynamically evolves based on defined criteria throughout the training process [16–21]. Data-based criteria and model-based criteria are two major categories that determine the importance of clients. Data-based criteria leverage the intrinsic properties of local datasets, considering elements such as the volume of data samples and the divergence in gradients or losses across data sets [16–18, 22]. Instead, model-based criteria prioritize clients with high model divergence from the global model [19], high angle divergence from the global model [20], or high magnitude of the local model [21].

While importance sampling methods have indeed facilitated enhancements in both performance and training efficiency in comparison with random client selection [18], they have some intrinsic constraints and pose serious challenges. Because most of the importance sampling methods are heuristic, the effectiveness of sampling designs can only be evaluated by empirical experiments. It is challenging to provide the performance guarantee for these algorithms with respect to the convergence rate and model accuracy [23]. Moreover, manually defined criteria make it harder to strike the tradeoff between exploitation, which focuses on selecting important clients, and exploration, which aims to involve a diverse range of clients. Another issue is the latency heterogeneity; clients with valuable data might exhibit slow training times, potentially extending the overall training time. To alleviate this, some strategies establish deadlines for model submissions, such as enforcing a cut-off [24], instituting a soft deadline [18], and applying a dynamic deadline [25]. Other strategies schedule the model transmission of participants in consideration of latency heterogeneity, such as round Robin scheduling [26], latency clustering [27] and multi-armed bandit scheduling [28].

In contrast to potentially biased importance sampling methods, probabilistic client selection strategies have demonstrated stronger convergence guarantees and remain unbiased in comparison with full participation methods [12, 29]. The foundational studies, including the federated optimization [30] and the Federated Averaging algorithm (FedAvg) [31], have presented a uniform selection scheme, where a subset of clients are randomly selected in each round of training. However, this typically exhibits a suboptimal convergence rate [10]. To improve FedAvg, extra prior knowledge has been used in determining client selection probabilities, taking into account factors like data volume [10], employing clustered sampling [32, 33], and analyzing the norms of the clients' model parameters [15]. However, most of these efforts neglect the impact of latency heterogeneity on the probability of client selection. A notable exception is a recent study that proposes an adaptive client sampling algorithm, which factors in both data and system heterogeneity to address latency issues [29]. However, this method is based on convergence analysis suited for strongly convex situations, which does not fully align with the non-convex characteristics commonly found in real-world federated learning applications.

To address the limitations of existing client selection methods, we formulate a nonlinear optimization problem with the aim of reducing overall training time. This produces an optimal probabilistic client selection scheme grounded in non-convex convergence analysis that accounts for both data and latency heterogeneity among clients. The main contributions of our paper are as follows:

1) We formulate a new latency-minimization problem that simultaneously optimizes client selection and training procedures in federated learning. This optimization problem consists of key variables such as the participant selection probability, the number of global rounds, and the number of participants. Our problem formulation incorporates both system and data diversity to ensure its comprehensiveness. Furthermore, we study an unbiased participant selection scheme, guaranteeing the model convergence.
2) We derive a new convergence upper bound for federated learning with probabilistic client selection in non-convex settings. Our results reflect a convergence rate of $\mathcal{O}(\frac{1}{T})$. This completes the convergence analysis for federated learning in both convex and non-convex cases.
3) We derive the analytical expression of the overall latency of federated learning with probabilistic participant selection and build the analytical relationship between the latency, convergence constraint, system and data heterogeneity, and control variables. To address the complexities of the mixed integer nonlinear programming problem, we introduce a hybrid solution that integrates grid search techniques with the polyhedral active set algorithm.
4) Through numerical analyses and experiments on real-world data, we demonstrate that our proposed client selection scheme is more efficient in reducing overall training time in comparison with existing methods.

In the remainder of this paper, we first introduce the system model and problem formulation in Section II. Next, Section III delves into the convergence upper bound of our algorithm and the solution to the optimization problem. Following this, we present numerical analyses and experimental results on real-world datasets in Section IV. We conclude this paper in Section V.

II. System Model

A. Federated Learning

We consider a federated learning system with a server and N clients with index set $\mathcal{N} = \{1, 2, ..., N\}$, where the i-th client owns its private dataset $\mathcal{D}_i = \{\xi_j^i \mid j = 1, 2, ..., |\mathcal{D}_i|\}$ with size $|\mathcal{D}_i|$. Here, ξ_j^i denotes the j-th data sample at the i-th client. The whole dataset across clients is denoted as $\mathcal{D} = \bigcup_{i\in\mathcal{N}} \mathcal{D}_i$ with size $|\mathcal{D}|$.

The goal of federated learning is to find an optimal model parameter x to minimize a global objective function $f(x)$ over dataset $\mathcal{D}$, which can be formulated as

$$\min_x f(x) \coloneqq \sum_{i=1}^{N} d_i F_i(x), \tag{1}$$

where $d_i = |\mathcal{D}_i|/|\mathcal{D}|$ denotes the ratio of the size of the local dataset at the i-th client to the whole dataset and $\sum_{i=1}^{N} d_i = 1$. $F_i(x)$ is the local objective function for the i-th client, computed over dataset $\mathcal{D}_i$ as

$$F_i(x) \coloneqq \frac{1}{|\mathcal{D}_i|} \sum_{\xi_j^i \in \mathcal{D}_i} F_i(x; \xi_j^i). \tag{2}$$

Client selection typically occurs during each round of federated learning to prevent network congestion due to full participation. This partial participation was proposed as FedAvg [31] by uniformly sampling participants.

We generalize FedAvg as shown in Algorithm 1, which allows probabilistic client selection. Moreover, we will show that the aggregated model in Algorithm 1 is unbiased to the one with full participation later. Specifically, the algorithm executes for T rounds in total after the stopping criterion is satisfied. For each round, there are three major stages:

1) Global model broadcasting. The sever samples M clients to participate in the t-th round of training to form $\mathcal{M}^t$, according to the client selection probability $\mathbf{p} = [p_1, p_2, ..., p_N]$. The global model weights, x^t, are broadcast to the selected participants.
2) Local updating. The i-th participant initializes its local model $y_{t,0}^i$ with the global one and performs model training on the private dataset $\mathcal{D}_i$ by applying the stochastic gradient descent (SGD) algorithm as follows:

$$y_{t,j+1}^i \leftarrow y_{t,j}^i - \gamma \nabla F_i(y_{t,j}^i, \xi_j^i), \tag{3}$$

where $j = 0, 1, 2, ..., I-1$ denotes the epoch index of local updating and γ denotes the step size. $\xi_j^i \in \mathcal{D}_i$ is a randomly selected data sample. After I epochs, the local model weights will be submitted to the server.
3) Aggregation. After the server successfully receives all the model updates from the selected participants, aggregation is performed to update the global model:

$$x_{t+1} = \sum_{i \in \mathcal{M}^t} \frac{d_i}{M p_i} y_{t,I}^i. \tag{4}$$

The stopping criterion is as follows:

$$\mathbb{E} \left\| \nabla f(x_T) \right\|^2 \leq \epsilon, \tag{5}$$

where ϵ is a small number.

In Algorithm 1, participation probability $\mathbf{p}$ should be properly chosen in consideration of data heterogeneity and latency differences among clients. Furthermore, the number of participants M and the number of global rounds T also affect the overall training time. Intuitively, a larger M and a greater T enhance the convergence of the global model. Nevertheless, a larger M increases the likelihood of selecting clients with the poorest response time. This straggler effect may consequently prolong the federated learning training process. The selection of T follows similar considerations.

Therefore, to achieve time-efficient training and model convergence, we explore a novel training scheduling policy by solving a joint optimization problem of total time cost and model convergence, which yields the optimal client-selection probability $\mathbf{p}$, number of participants M, and number of global rounds T.

Algorithm 1: Generalized FedAvg with probabilistic participation

Input: $x_0, \gamma, I, \mathbf{p}$
Output: $\{x_r : \forall r\}$
for $t = 0, 1, ..., T-1$ **do**
  select M participants according to $\mathbf{p}$ with replacement to form $\mathcal{M}^t$;
  for $i \in \mathcal{M}^t$ *in parallel* **do**
    $y_{t,0}^i \leftarrow x_t$;
    $y_{t,I}^i \leftarrow$ **ClientUpdate**$(y_{t,0}^i, \gamma)$;
  end
  $x_{t+1} \leftarrow \sum_{i \in \mathcal{M}^t} \frac{d_i}{M p_i} y_{t,I}^i$;
end
ClientUpdate $(y_{t,0}^i, \gamma)$:
for $j = 0, 1, ..., I-1$ **do**
  $y_{t,j+1}^i \leftarrow y_{t,j}^i - \gamma \nabla F_i(y_{t,j}^i, \xi_j^i)$;
end

B. Latency Model

For each training round, there are four phases where latency occurs: global model broadcasting, local computing, model uploading, and model aggregation. However, the time cost of model aggregation is negligible given the simple operation.

Similar to previous work in wireless federated learning [34], we make an assumption that each edge device possesses a comparatively consistent computational capability yet operates within varying communication environments. Therefore, we use δ_i^t to denote the response time of the i-th client in the t-th round, resulting from global model broadcasting, local computing, and model uploading. Different clients may exhibit various latencies. Inspired by the scheduling optimization theory [35], we adopt the short-processing time first rule to schedule the model transmission of participants. In the t-th round, we can sort the latency of each client in ascending order as $\delta_1^t \leq \delta_2^t \leq ... \leq \delta_N^t$.

In synchronous federated learning, the server waits for all the model updates from participants to be received and then starts to aggregate the models. As a result, the latency for the t-th round is determined by the client with the longest response time, i.e.,

$$\Delta^t = \max_{i \in \mathcal{M}^t} \delta_i^t. \tag{6}$$

Accordingly, the overall latency for T rounds of training is

$$\Delta = \sum_{t=1}^{T} \Delta^t. \tag{7}$$

C. Client Selection Model

We assume that all the clients are willing to participate in every round of federated learning training once they are selected. We consider general probabilistic client selection according to $\mathbf{p} = [p_1, p_2, ..., p_N]$ in each round, where $\sum_{i=1}^{N} p_i = 1$. Similar to [29, 36, 37], the t-th round of participant set $\mathcal{M}^t$ is

generated by selecting for M times from N clients according to $\mathbf{p}$ with replacement, which means that the same client could appear in $\mathcal{M}^t$ multiple times. The sampling scheme without replacement can also be extended from our work, but for ease of exposition, we only discuss sampling with replacement.

Let X_i^t be a random variable denoting the number of times the i-th client is selected in the t-th round. Then, X_i^t follows a binomial distribution with parameters M and p_i, i.e., $X_i^t \sim \text{Binomial}(M, p_i)$. The expectation of X_i^t is given by

$$\mathbb{E}[X_i^t] = Mp_i. \tag{8}$$

As a result, the expected model updates from the i-th client can be expressed as

$$\mathbb{E}_{\mathbf{p}}(y_{t,I}^i) = \mathbb{E}[X_i^t] y_{t,I}^i = Mp_i y_{t,I}^i. \tag{9}$$

D. Problem Formulation

As discussed, client selection probability $\mathbf{p}$ determines the probability of selecting the straggler, which greatly affects the per-round latency. Furthermore, more training rounds T and a larger number of participants M are beneficial to the model convergence, while they potentially increase the overall latency.

Therefore, we formulate a joint optimization problem with respect to $\mathbf{p}$, T, and M to minimize the overall latency of federated learning training while satisfying the model convergence requirement. The problem can be formulated as follows.

$$\min_{\mathbf{p},T,M} \quad \sum_{t=1}^{T} \max_{i\in\mathcal{M}^t} \delta_i^t \tag{P1}$$

$$\text{s.t.} \quad \mathbb{E}\left\|\nabla f(x_T)\right\|^2 \le \epsilon, \tag{P1a}$$

$$\sum_{i=1}^{N} p_i = 1, \tag{P1b}$$

$$T \in \mathbb{Z}^+, M \in \mathbb{Z}^+, \tag{P1c}$$

where $\mathbb{E}\left\|\nabla f(x_T)\right\|^2$ denotes the expected gradient norm of f after T rounds of training where the randomness is from local SGD and client selection. (P1a) denotes the model convergence constraint and (P1b) is a basic constraint of a probability distribution.

In practice, $\max_{i\in\mathcal{M}^t} \delta_i^t$ is a random variable due to the probabilistic client selection, which makes it impossible to analytically solve the problem. Instead, we change the objective to be the expected overall latency, i.e.,

$$\min_{\mathbf{p},T,M} \quad \mathbb{E}(\Delta). \tag{10}$$

Moreover, we need to explicitly express $\mathbb{E}\left\|\nabla f(x_T)\right\|^2$ w.r.t $\mathbf{p}$, T, and M, which will be discussed in III-C.

III. Training Scheduling with Client-selection Probability

In this section, we aim to transform Problem (P1) into a more tractable form. Firstly, we derive the analytical expression of $\mathbb{E}(\Delta)$. Through convergence analysis, we obtain the upper bound of $\mathbb{E}\left\|\nabla f(x_T)\right\|^2$ and utilize it to reformulate constraint (P1a). Subsequently, we formulate and solve an alternative optimization problem, (P2), to determine the optimal training scheduling in terms of $\mathbf{p}$, T, and M.

A. Analytical Expression of Expected Latency

Let q_i denote the probability of the i-th client being selected in the t-th round and being a straggler. According to (6) and (7), we have the expected overall latency as follows:

$$\begin{aligned}\mathbb{E}(\Delta) &= \sum_{t=1}^{T}\sum_{i=1}^{N} q_i \delta_i^t \\ &= T\sum_{i=1}^{N} q_i \delta_i,\end{aligned} \tag{11}$$

where $\delta_i = \frac{1}{T}\sum_{t=1}^{T}\delta_i^t$ denotes the average respond time of the i-th client.

Moreover, the i-th client being a straggler means that only clients $1, 2, ..., i$ are candidates of participants in this round. Therefore, we have

$$\begin{aligned} q_i &= \mathbb{P}(i \text{ is a straggler}) \\ &= \sum_{m=1}^{M} \binom{M}{m} p_i^m \left(\sum_{j=1}^{i-1} p_j\right)^{M-m} \\ &= \left(\sum_{j=1}^{i} p_j\right)^M - \left(\sum_{j=1}^{i-1} p_j\right)^M, \end{aligned} \tag{12}$$

where the last equality can be obtained according to the Binomial Theorem.

As a result, the analytical expression of the expected overall latency is given by

$$\mathbb{E}(\Delta) = \sum_{i=1}^{N}\left[\left(\sum_{j=1}^{i} p_j\right)^M - \left(\sum_{j=1}^{i-1} p_j\right)^M\right]\delta_i T. \tag{13}$$

Let $l_i = \left(\sum_{j=1}^{i} p_j\right)^M$, (13) can be rewritten as

$$\begin{aligned}\mathbb{E}(\Delta) &= \sum_{i=1}^{N}(l_i - l_{i-1})\delta_i T \\ &\overset{(a)}{=} \left[\sum_{i=1}^{N} l_i\delta_i - \sum_{i=2}^{N} l_{i-1}\delta_i\right] T \\ &= \left[\sum_{i=1}^{N} l_i\delta_i - \sum_{i=1}^{N-1} l_i\delta_{i+1}\right] T \\ &\overset{(b)}{=} \left[\delta_N - \sum_{i=1}^{N-1} l_i(\delta_{i+1} - \delta_i)\right] T \\ &= \left[\delta_N - \sum_{i=1}^{N-1}\left(\sum_{j=1}^{i} p_j\right)^M (\delta_{i+1} - \delta_i)\right] T, \end{aligned} \tag{14}$$

where (a) uses $l_0 = 0$ and (b) uses $l_N = 1$.

B. Adjusted Aggregation Weights for Probabilistic Selection

Despite only M participants being selected for the t-th round, we demonstrate that probabilistic client selection remains unbiased towards full participation by adjusting aggregation weights, as exhibited in line 7 of Algorithm 1.

Consider FedAvg with full participation. The aggregation rule is given by

$$\bar{x}_{t+1} = \sum_{i=1}^{N} d_i y_{t,I}^i. \tag{15}$$

Let a_i denote the adjusted aggregation weight for the i-th client. The aggregated model can be expressed as $x_{t+1} = \sum_{i\in\mathcal{M}^t} a_i y_{t,I}^i$. Therefore, the expected global model with $\mathbf{p}$ is

$$\mathbb{E}_{\mathbf{p}}(x_{t+1}) = \sum_{i=1}^{N} a_i \mathbb{E}_{\mathbf{p}}(y_{t,I}^i) \overset{(a)}{=} \sum_{i=1}^{N} a_i M p_i y_{t,I}^i, \tag{16}$$

where (a) is using (9).

To make $\mathbb{E}_{\mathbf{p}}(x_{t+1})$ unbiased to $\bar{x}_{t+1}$, we have

$$\sum_{i=1}^{N} a_i M p_i y_{t,I}^i = \sum_{i=1}^{N} d_i y_{t,I}^i. \tag{17}$$

Then we can easily obtain

$$a_i = \frac{d_i}{M p_i}. \tag{18}$$

Therefore, the unbiased aggregation rule in federated learning with client-selection probability $\mathbf{p}$ is

$$x_{t+1} = \sum_{i\in\mathcal{M}^t} \frac{d_i}{M p_i} y_{t,I}^i. \tag{19}$$

C. Convergence Analysis

In the following, we will derive the convergence upper bound for Algorithm 1. To facilitate the convergence analysis, we use some common assumptions [12, 29, 37] about local objective functions $\{F_i\}$:

Assumption 1. *$F_i(x)$ is continuous and differentiable, i.e., $\nabla F_i(x)$ exists. $F_i(x)$ is lower bounded by $F_i(x^*)$.*

Assumption 2. *The gradient of $F_i(x)$ is L-Lipschitz continuous: for any $x, y \in dom(F_i)$, we have $\|\nabla F_i(x) - \nabla F_i(y)\| \leq L\|x - y\|$.*

Assumption 3. *The expected second moment of $\nabla F_i(x)$ is bounded: for any data sample $\xi_j^i \in \mathcal{D}_i$ and when there exists a constant $G_i > 0$, we have $\mathbb{E}(\left\|\nabla F_i(x, \xi_j^i)\right\|^2) \leq G_i^2, \forall x \in dom(F_i)$.*

It is worth noting that there are some related studies on the convergence analysis of FedAvg with client selection [29, 37]. However, those results rely on the strongly convex assumption of F_i, which is unrealistic for deep learning models. Instead, our convergence upper bound can be used in non-convex scenarios.

With the adjusted aggregation rule by (19), we present the convergence result of federated learning with client-selection probability.

Theorem 1. *Let Assumptions 1 to 3 hold. When $\gamma \leq \frac{1}{LI}$, the federated learning with client-selection probability $\mathbf{p}$ satisfies*

$$\begin{aligned}\min_t \mathbb{E}\left\|\nabla f(x_t)\right\|^2 \leq & \frac{2\Gamma}{T\gamma I} + \frac{(I-1)(2I-1)L^2\gamma^2}{6}\sum_{i=1}^{N} d_i G_i^2 \\ & + \frac{LI\gamma}{M}\sum_{i=1}^{N}\frac{d_i^2 G_i^2}{p_i},\end{aligned} \tag{20}$$

where $\Gamma = f(x_0) - f(x^*)$.

We first use the L-Lipschitz gradient assumption to build the relationship between $f(x_{t_k})$ and $f(x_{t_{k+1}})$, denoting the function values of the k-th and $(k+1)$-th global models, respectively. We can find that $\left\|\nabla f(x_{t_k})\right\|^2$ is upper bounded by an affine function of $f(x_{t_k}) - f(x_{t_{k+1}})$. After summing up $\left\|\nabla f(x_{t_k})\right\|^2$ for $k = 0, 1, ..., T-1$, we can find the upper bound of $\min_t \mathbb{E}\left\|\nabla f(x_t)\right\|^2$. The full proof is in Appendix A.

From Theorem 1, it is noteworthy that our results reflect a convergence rate of $\mathcal{O}(\frac{1}{T})$. This rate aligns with the established convergence results for scenarios assuming strong convexity of local functions $\{F_i\}$ [37].

Corollary 1. *Choosing $\gamma = \frac{1}{LI\sqrt{T}}$, where $T \geq 1$, we have*

$$\begin{aligned}\min_t E\left\|\nabla f(x_t)\right\|^2 \leq & \frac{1}{\sqrt{T}}\left(2L\Gamma + \frac{1}{M}\sum_{i=1}^{N}\frac{d_i^2 G_i^2}{p_i}\right) \\ & + \frac{1}{T}\frac{(I-1)(2I-1)}{6I^2}\sum_{i=1}^{N} d_i G_i^2,\end{aligned} \tag{21}$$

where $\Gamma = f(x_0) - f(x^*)$.

Directly plugging $\gamma = \frac{1}{LI\sqrt{T}}$ into (20) produces (21).

Remark 1. *G_i measures the degree of none Independent and Identically Distributed (non-I.I.D.) distribution across clients. When clients' data are I.I.D., we have $G_1 = G_2 = \cdots = G_n = 0$, which makes upper bound be $O(\frac{1}{\sqrt{T}})$.*

Remark 2. *More global rounds T and more participants M reduce the upper bound.*

Corollary 2. *Let $\rho = \frac{\max_i G_i^2}{\max_i d_i G_i^2}$ be the heterogeneity ratio. When $T \geq T_d$, where $T_d = \left\lceil \frac{M^2\rho^2}{9} \right\rceil$, the first term in (20) dominates. Therefore, we have*

$$\min_t E\left\|\nabla f(x_t)\right\|^2 \leq \frac{1}{\sqrt{T}}\left(2L\Gamma + \frac{1}{M}\sum_{i=1}^{N}\frac{d_i^2 G_i^2}{p_i}\right), \tag{22}$$

where $\Gamma = f(x_0) - f(x^*)$.

Proof of Corollary 2 can be found in Appendix B.

D. Alternative Optimization Problem

To re-formulate Problem (P1), we use (22) to replace the convergence constraint and use the new expected overall latency $\mathbb{E}(\Delta)$ in (14) as the objective function. We have

$$\min_{\mathbf{p},T,M} \quad \left[\delta_N - \sum_{i=1}^{N-1}\left(\sum_{j=1}^{i} p_j\right)^M (\delta_{i+1} - \delta_i)\right] T \tag{P2}$$

$$\text{s.t.} \quad \frac{1}{\sqrt{T}}\left(\alpha+\frac{1}{M}\sum_{i=1}^{N}\frac{d_i^2G_i^2}{p_i}\right)\leq\epsilon, \tag{P2a}$$

$$\sum_{i=1}^{N}p_i=1, \tag{P2b}$$

$$T\in\mathbb{Z}^+, M\in\mathbb{Z}^+, \tag{P2c}$$

where $\alpha=2L\Gamma$.

Problem (P2) is a mixed integer nonlinear programming problem due to the inclusion of T and M, and its objective function is non-convex concerning $\mathbf{p}$ and M.

Before we present our solution to Problem (P2), we first discuss a special case where the response times are uniform across all clients. Under this uniform response time condition, Problem (P2) can be simplified into a convex optimization problem where the objective function becomes $\delta_N T$. This special case is noteworthy because it allows us to apply a different analytical approach to find the optimal solution. Solving the convex problem by the method of Lagrange multipliers produces Corollary 3 and we refer to its solution as *norm selection*.

Corollary 3. ***(Norm Selection)****. When $\delta_1=\delta_2=\cdots=\delta_N$, Problem (P2) is a convex problem and its optimal client selection probability is $p_i^*=\frac{d_iG_i}{\sum_{i=1}^{N}d_iG_i}$ for the i-th client.*

E. Solution to P2

With continuous ($\mathbf{p}$) and integer (T, M) variables, (P2) is a mixed integer nonlinear programming problem, which is difficult to solve directly in general.

As M denotes an integer representing the number of participants, we have $M\in\{1,2,...,N\}$, yielding a maximum of N candidates for (P2)'s optimal solution. Therefore, we apply grid search to find the best M. Given fixed M, we utilize the polyhedral active set algorithm to obtain the optimal $(\mathbf{p}^*,T^*)$. Subsequently, we calculate the objective function of (P2) using the candidate solution $(\mathbf{p}^*,T^*,M)$. Finally, we obtain the optimal solution $(\mathbf{p}^*,T^*,M^*)$ that yields the minimized value of the objective function.

It is worth noting that the grid search method results in a computational complexity that is linear with respect to the number of clients N. As N increases, the complexity of the solution also increases. However, since the computation is performed offline before the actual training process, it does not impact the runtime performance. For extremely large N, alternative strategies such as adaptive sampling or hierarchical search methods can be employed to further reduce the computational complexity.

Specifically, we first relax T to be a continuous variable, which converts constraint (P2c) to be $T\in\mathbb{R}^+$. According to (P2) and (P2a), we can see that T_i^* satisfies

$$\frac{1}{\sqrt{T^*}}\left(\alpha+\frac{1}{M}\sum_{i=1}^{N}\frac{d_i^2G_i^2}{p_i}\right)=\epsilon, \tag{23}$$

i.e.,

$$T^*=\frac{1}{\epsilon^2}\left(\alpha+\frac{1}{M}\sum_{i=1}^{N}\frac{d_i^2G_i^2}{p_i}\right)^2. \tag{24}$$

With (24) and fixed M, we can re-formulate Problem (P2) as follows:

$$\begin{aligned}\min_{\mathbf{p}} \quad & \left[\delta_N-\sum_{i=1}^{N-1}\left(\sum_{j=1}^{i}p_j\right)^M(\delta_{i+1}-\delta_i)\right] \\ & \times\left(\alpha+\frac{1}{M}\sum_{i=1}^{N}\frac{d_i^2G_i^2}{p_i}\right)^2\end{aligned} \tag{P3}$$

$$\text{s.t.} \quad \sum_{i=1}^{N}p_i=1. \tag{P3a}$$

Problem (P3) represents a nonlinear optimization challenge with polyhedral constraints. To address this issue, we employ the polyhedral active set algorithm [38], which is comprised of two distinct phases: the initial phase implements the gradient projection technique, whereas the subsequent phase incorporates an appropriate algorithm tailored for linearly constrained optimization problems. By alternating between these two phases according to well-defined branching criteria, the Polyhedral active set algorithm guarantees the convergence to a stationary point.

Therefore, for any given M, $\mathbf{p}^*$ is solved by (P3) and T^* is given by

$$T^*=\left\lceil\frac{1}{\epsilon^2}(\alpha+\frac{1}{M}\sum_{i=1}^{N}\frac{d_i^2G_i^2}{p_i})^2\right\rceil, \tag{25}$$

where $\lceil\cdot\rceil$ denotes the ceiling function.

At last, by searching $M\in\{1,2,...,N\}$, we obtain the suboptimal solution $(\mathbf{p}^*,T^*,M^*)$ of Problem (P3), which is our training scheduling for federated learning.

F. Estimates of Unknown Parameters

As shown in Section III-E, p_i^* and T_i^* are decided by d_i, G_i and α, which are unknown to the server prior to the federated learning training. Therefore, we need to estimate those unknown parameters.

We run one trial experiment with $\mathbf{p}_a=[d_1,d_2,...,d_N]$. The trial runs total T rounds and in each round M clients are selected according to the probabilistic client selection. Suppose ϵ_a denotes the loss value after T-rounds training for $\mathbf{p}_a$ and $\tilde{\alpha}$ be the estimate of α, we have the following equality according to (23):

$$\epsilon_a=\frac{1}{\sqrt{T}}\left(\tilde{\alpha}+\frac{1}{M}\sum_{i=1}^{N}d_iG_i^2\right), \tag{26}$$

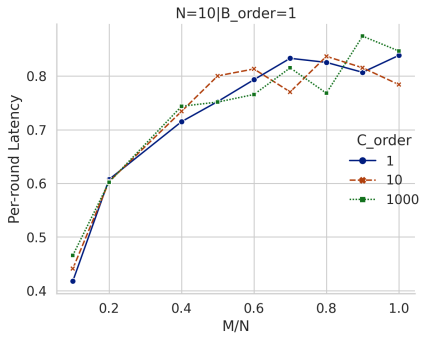
where d_i and G_i can be reported by clients along with their model updates.

By solving (26), we have $\tilde{\alpha}$ given by

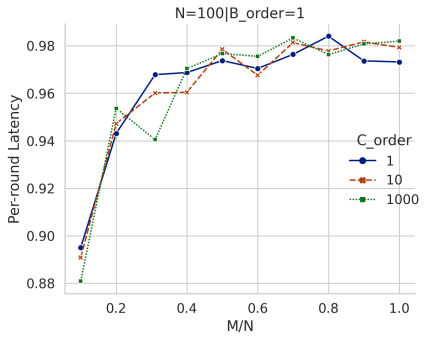
$$\tilde{\alpha}=\sqrt{T}\epsilon_a-\frac{1}{M}\sum_{i=1}^{N}d_iG_i^2. \tag{27}$$

To reduce the estimation error, we can record different sets of (T,ϵ) to obtain an averaged $\tilde{\alpha}$ in just one trial experiment. Regarding the cost, this estimation process does not add much communication overhead since only two numbers (d_i and G_i)

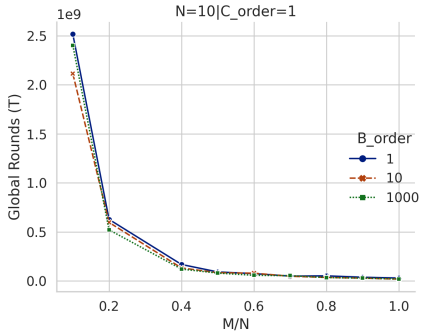
This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3425757



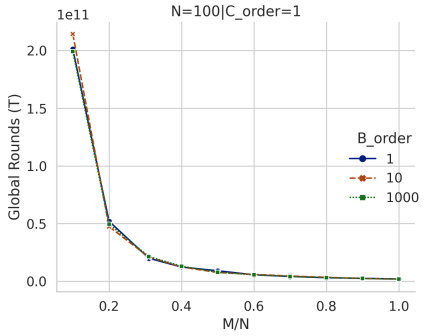
(a) Per-round latency when $N = 10$.



(b) Per-round latency when $N = 100$.

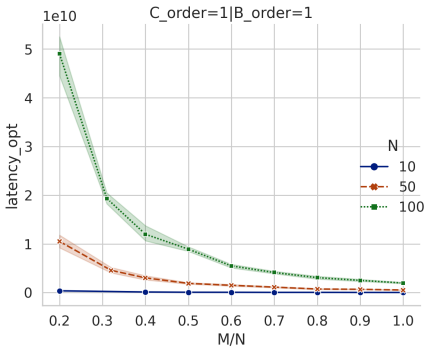


(c) Global rounds when $N = 10$.

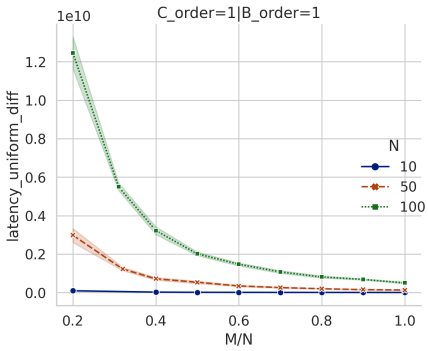


(d) Global rounds when $N = 100$.

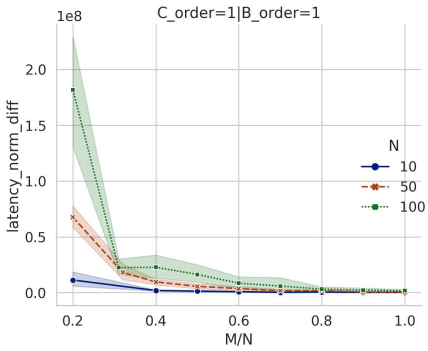
Fig. 2: Impact of C_{order} and B_{order} on optimal per-round latency and global rounds. The values of per-round latency for $C_{order} = 10$ and $C_{order} = 1000$ are normalized to allow for a direct comparison with $C_{order} = 1$. Similarly, the values of global rounds for $B_{order} = 10$ and $B_{order} = 1000$ are also normalized for the same reason.



(a) Optimal overall latency with various N and M/N.



(b) Latency difference between ours and uniform selection.



(c) Latency difference between ours and norm selection.

Fig. 3: Optimal overall latency and latency difference comparison.

are added to the uploaded data. The computation cost is also low because the trial experiment does not need to be a full training process. T can be set very small as long as all clients have participated at least once. Furthermore, the trained model in the estimation process can be reused as a good initial model for the experiment with $(\mathbf{p}^*, T^*, M^*)$, which avoids repeating experiments.

IV. Experiments

A. Numerical Experiments

We conduct a series of numerical experiments to study the effect of parameters such as M, δ_i, d_i, and G_i on the solution to Problem (P3). We also compare our optimal client-selection probability $\mathbf{p}^*$ with other probabilities in terms of the number of global rounds T and the overall latency $\mathbb{E}(\Delta)$.

1) Reformatting (P3): To simplify the parameter set, we define the data factor $B_i = d_i^2 G_i^2$ and the neighboring latency difference $C_i = \delta_{i+1} - \delta_i$. Therefore, Problem (P3) can be reformulated as follows:

$$\min_{\mathbf{p}} \quad \left[\delta_N - \sum_{i=1}^{N-1} C_i \left(\sum_{j=1}^{i} p_j\right)^M\right] \left(\alpha + \frac{1}{M}\sum_{i=1}^{N} \frac{B_i}{p_i}\right)^2 \tag{P4}$$

$$\text{s.t.} \quad \sum_{i=1}^{N} p_i = 1. \tag{P4a}$$

2) Experiment Settings: **Hyperparameters**. We utilize the SuiteOPT toolbox [38] to solve our optimization problem (P4). With $\alpha = 1$ and $\epsilon = 10^{-3}$, we perform an ablation study on the other parameters in Problem (P4). Specifically, we explore the impacts of the total number of clients $N \in \{10, 50, 100\}$ and the participation ratio $M/N \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$ on the selection probability. Furthermore, we study the effect of the discrepancy of local data by modifying the order of $B_{order} \in \{1, 10, 10^3\}$ and uniformly sampling $B_i \in (0, B_{order}]$. Similarly, to demonstrate the effect of the latency differences of clients, we change the order of $C_{order} \in \{1, 10, 10^3\}$ and uniformly sample $C_i \in (0, C_{order}]$. We repeat the experiment 10 times for each combination of parameters.

Evaluation Metrics. We adopt three metrics to evaluate the performance of different client-selection probabilities: expected per-round latency $\mathbb{E}(\Delta^t)$, global rounds T, and overall latency $\mathbb{E}(\Delta) = \mathbb{E}(\Delta^t)T$. We can compute $\mathbb{E}(\Delta^t)$ and T as

$$\mathbb{E}(\Delta^t) = \delta_N - \sum_{i=1}^{N-1} C_i \left(\sum_{j=1}^{i} p_j \right)^M \tag{28}$$

and

$$T = \left\lceil \frac{1}{\epsilon^2} (\alpha + \frac{1}{M} \sum_{i=1}^{N} \frac{B_i}{p_i})^2 \right\rceil, \tag{29}$$

respectively, where $B_i = d_i^2 G_i^2$ and $C_i = \delta_{i+1} - \delta_i$.

Benchmarks. We compare our solution to Problem (P4), denoted as $\mathbf{p}^*$, with two other selection schemes. The first one is the *uniform selection*, where $p_1 = p_2 = \cdots = p_N = 1/N$, which is the selection scheme used in FedAvg [31]. The second one is the *norm selection*, where participants are selected according to p_i calculated by the server based on the norms of participants' updates [15].

3) Experiment Results: The impact of N and M/N on the optimal per-round latency and global rounds is greater than that of C_{order} and B_{order}. Figures 2a and 2b illustrate the average optimal per-round latency with various values of C_{order} when $N = 10$ and $N = 100$, respectively. The per-round latency for $C_{order} = 10$ and $C_{order} = 1000$ is normalized by eliminating the effect of the magnitude of C, allowing for a direct comparison with $C_{order} = 1$. It is observed that different C_{order} values produce similar per-round latencies. However, per-round latency dramatically increases as the number of clients N or the participant ratio M/N increases. Similarly, as shown in Figures 2c and 2d, the number of global rounds increases as fewer clients participate in training per round or the number of clients increases. In contrast, different values of B_{order} play a small role in determining the number of global rounds. These observations suggest that the magnitude of C_i and B_i do not change the optimal selection probability $\mathbf{p}^*$.

Figure 3a shows the optimal overall latency with various N and M/N for $C_{order} = 1$ and $B_{order} = 1$. It shows that as the number of clients N decreases and the participation ratio M/N increases, the overall latency decreases. Additionally, the variance of the overall latency also decreases when N decreases and M/N increases. These results suggest that smaller client populations and higher participation ratios result in more efficient communication overall.

Figure 3b and 3c illustrate the difference in the overall latency between our selection scheme and uniform selection, and norm selection, respectively. It is observed that our selection scheme has the smallest overall latency. Interestingly, when N is sufficiently large, norm selection provides a good estimate of the optimal selection probabilities. This is further demonstrated in experiments conducted on real-world datasets.

B. Experiments on Real-World Datasets

In this section, we evaluate the effectiveness of our client selection probabilities in the generalized federated learning process (Algorithm 1) using real-world datasets.

1) Experimental Settings: **Platform**: We develop a customized federated learning platform using the Tensorflow Federated framework [39]. This platform allows for multi-machine simulation runtime experiments and can also be extended to multi-device implementation. Our experiments are conducted on a High-Performance Computing (HPC) cluster comprised of eight nodes, each of which is equipped with two 32-core Intel CPUs and four NVIDIA Ampere A100 GPUs with NVLink interconnect.

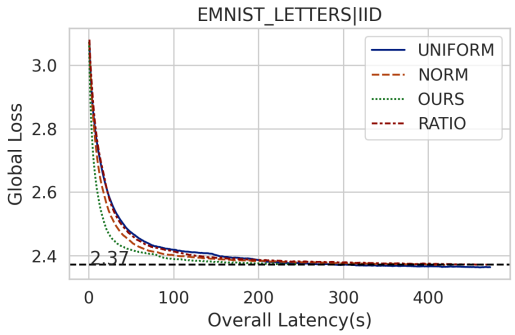
Datasets and Model: We use the EMNIST_LETTERS [40] and FASHION_MNIST [41] datasets. EMNIST_LETTERS contains images of 26 lowercase letters and FASHION_MNIST contains 10 different image classes. The training data of each dataset is partitioned into clients' datasets, while the testing data is used to evaluate the performance of our method. We use LeNet-5 [42] as the classification model.

Latency Heterogeneity: To emulate latency differences among clients, we independently sample $\delta \sim U(0, 1)$ for N clients and then arrange them in an ascending order such that $0 < \delta_1 \leq \delta_2 \leq \ldots \leq \delta_N \leq 1$. While δ_i is in seconds in the following results, this simulation is capable of representing the normalization of any latency heterogeneity scheme.

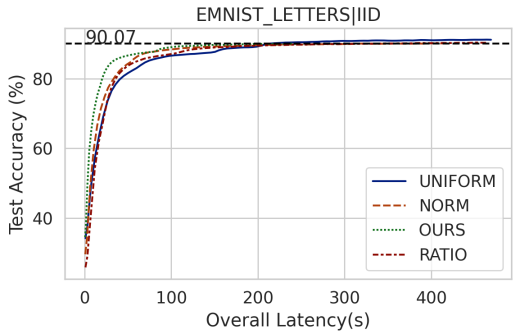
Data Heterogeneity: To simulate data heterogeneity in real-world federated learning applications, we employ three different data partition configurations, one I.I.D. and two Non-I.I.D. configurations.

- *I.I.D.* The training data is randomly partitioned among clients such that each client has an equal amount of data and an equal amount for each class.
- *Class*. The training data is partitioned among clients based on classes, with each client having data from only C randomly selected classes and with no overlapping data between clients, while the data volume in each client is the same. We let a client own 50% classes of the dataset, i.e., $C = 13$ for EMNIST_LETTERS and $C = 5$ for FASHION_MNIST.
- *Dir*. The training data is partitioned among clients following a Dirichlet process, with each client having a different amount of data and a non-uniform class distribution. We set the Dirichlet parameter $\alpha = 0.1$.

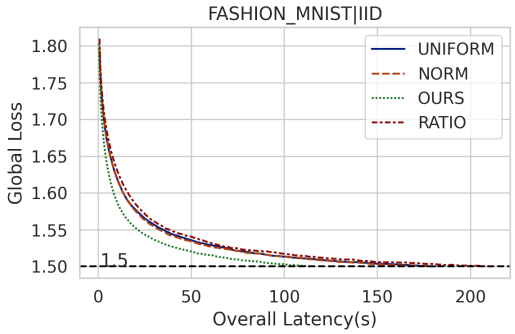
This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3425757



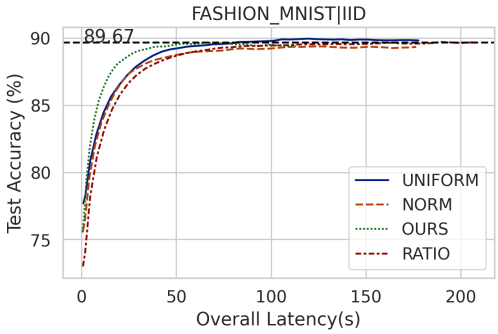
(a) Global loss on EMNIST_LETTERS dataset.



(b) Test accuracy on EMNIST_LETTERS dataset.

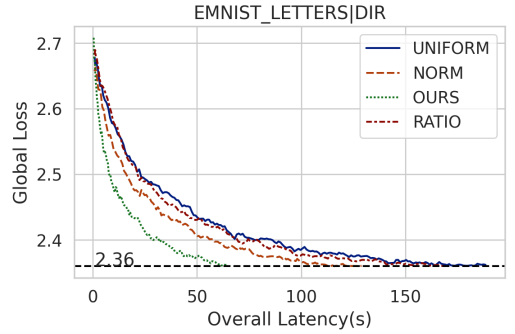


(c) Global loss on FASHION_MNIST dataset.

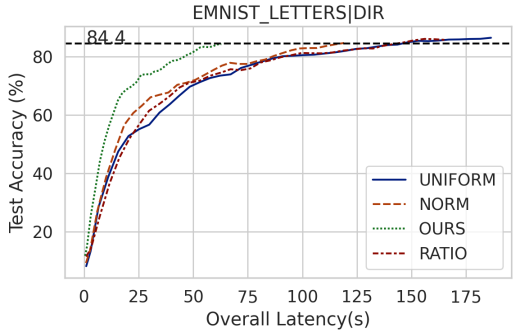


(d) Test accuracy on FASHION_MNIST dataset.

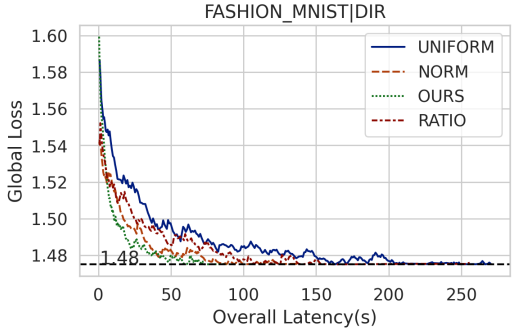
Fig. 4: Performance comparison under *I.I.D.* data setting.



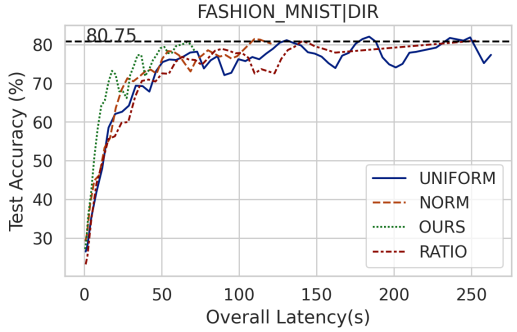
(a) Global loss on EMNIST_LETTERS dataset.



(b) Test accuracy on EMNIST_LETTERS dataset.



(c) Global loss on FASHION_MNIST dataset.



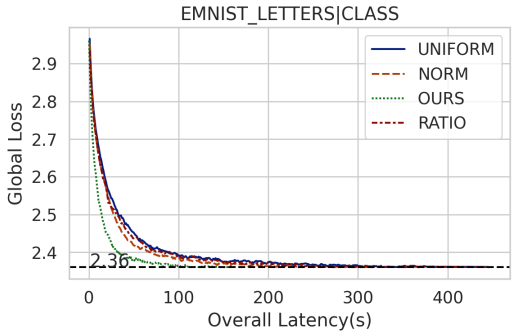
(d) Test accuracy on FASHION_MNIST dataset.

Fig. 5: Performance comparison under *Dir* data setting.

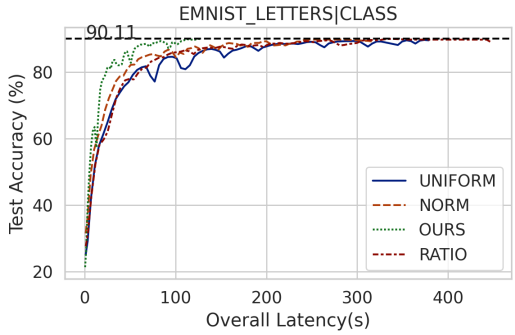
This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3425757

Dataset	Data Setting	Ours	Uniform [31]	Norm [15]	Ratio [10]
EMNIST_LETTERS	I.I.D.	**245.66**	472.87 (**1.92**×)	419.93 (**1.71**×)	469.51 (**1.91**×)
	Class	**164.82**	444.37 (**2.70**×)	317.70 (**1.93**×)	447.33 (**2.71**×)
	Dir.	**64.12**	188.33 (**2.94**×)	128.90 (**2.01**×)	169.61 (**2.65**×)
FASHION_MNIST	I.I.D.	**114.05**	181.46 (**1.59**×)	177.35 (**1.56**×)	210.93 (**1.85**×)
	Class	**220.92**	439.33 (**1.99**×)	392.50 (**1.78**×)	439.42 (**1.99**×)
	Dir.	**88.08**	270.14 (**3.07**×)	137.35 (**1.56**×)	255.69 (**2.90**×)

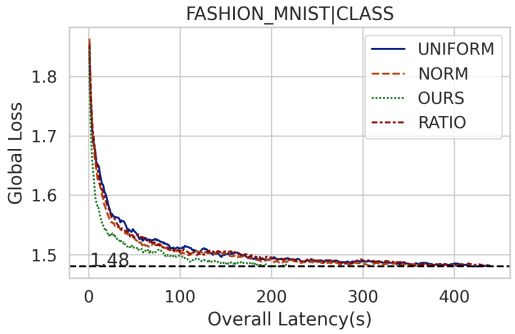
TABLE I: Overall latency in seconds of various methods under different datasets and data heterogeneity settings. $(\cdot\ \times)$ denotes the acceleration ratio of ours in comparison with benchmarks in terms of the overall latency.



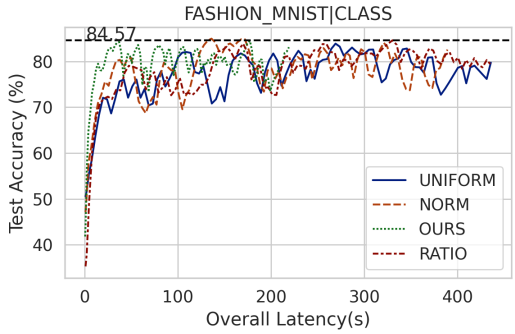
(a) Global loss on EMNIST_LETTERS dataset.



(b) Test accuracy on EMNIST_LETTERS dataset.



(c) Global loss on FASHION_MNIST dataset.



(d) Test accuracy on FASHION_MNIST dataset.

Fig. 6: Performance comparison under *Class* data setting.

Hyperparameters: For our experiments, we set the total number of clients to $N = 10$ for FASHION_MNIST and $N = 40$ for EMNIST_LETTERS. The number of participants is $M = 5$ for FASHION_MNIST and $M = 10$ for EMNIST_LETTERS. In each round, participants are selected according to given selection schemes, and each participant updates the local model in $I = 5$ epochs using SGD with the batch size of 256. The default parameters of Adam optimizer in Tensorflow are used, i.e., learning rate is 0.001 and exponential decay rates are 0.9 and 0.999, respectively [43].

Benchmarks: To evaluate the performance of our proposed selection scheme, we compare it with these existing schemes:

- *Uniform selection* [31]. The probability for each client to be selected is identical, i.e., $p_1 = p_2 = \cdots = p_N = 1/N$.
- *Norm selection* [15]. Participants are selected according to p_i, where p_i is calculated by the server based on the norms of participants' updates.
- *Ratio selection* [10]. Participants are selected according to $p_i = d_i$, proposed by FedProx.

2) Experiment Results: We evaluate the global loss and test accuracy of different selection schemes on two datasets under different data settings. Each experiment is independently run 10 times with different random seeds and the same seed is used across different selection schemes. Results averaged over 10 runs are reported.

In Table I, we summarize the overall latency required to achieve the targeted global loss with various client selection schemes on EMNIST_LETTERS and FASHION_MNIST. As is shown, ours outperforms other selection schemes in both I.I.D. and non-I.I.D. configurations with the lowest overall latency and achieves up to 3 times acceleration. More details are shown by the decreasing global loss in (a) and (c) of Figures 4, 5, and 6.

In comparison with the I.I.D. setting, our approach attains more acceleration ratios within two none-I.I.D. scenarios. Specifically, in contrast to uniform selection on EMNIST_LETTERS, our method yields $2.94\times$ and $2.7\times$ acceleration for *Dir* and *Class* categories, respectively, surpassing the $1.92\times$ acceleration observed in the I.I.D. scenario.

As shown in (b) and (d) of Figures 4, 5, and 6, our approach, in some cases, achieves superior test accuracy with much less overall latency required for the convergence of the training model. In the non-I.I.D. scenarios, our methodology exhibits enhanced robustness in comparison with uniform selection, as evidenced by the mitigation of test accuracy fluctuations.

It is worth noting that norm-based selection frequently facilitates the reduction of the overall latency relative to uniform selection, as shown in Table I. It indicates that the prior knowledge of d_i and G_i is helpful for a better client selection scheme. On the other hand, norm selection is not optimal because it ignores the latency differences among clients.

V. Conclusion

We have introduced a novel client selection scheme designed to minimize the overall training time of federated learning by considering both data and latency heterogeneity among clients. Through the derivation of a convergence upper bound with probabilistic client selection, we have established the theoretical convergence guarantee for our proposed scheme. Our numerical analyses and experimental evaluations on real-world datasets have demonstrated the superiority of our selection scheme in achieving faster convergence rates and competitive test accuracy, even in scenarios with highly non-I.I.D. data. Notably, our scheme obtains up to 3 times acceleration in comparison with random client selection.

In the future, conducting extensive experiments on large-scale datasets, such as ImageNet, and using complex models, such as ResNet-50 and Vision Transformers (ViTs), will further explore the applicability of our optimization framework across diverse learning scenarios. Additionally, developing a unified framework to evaluate both probabilistic selection and dynamic selection methods is an intriguing research direction. Moreover, implementing a federated learning testbed with real federated clients will provide additional insights into the practical network variability and real-world deployment challenges.

Appendix A
Proof of Theorem 1

Throughout the proof, we denote $t_k = kI, k = 0, 1, ..., T-1$ as the time instants when global aggregation happens. We define $\bar{y}_{t_k,I} = \sum_{i=1}^{N} d_i y^i_{t_k,I}$, which denotes the virtual aggregated model with all the N participants. To aid the proof of Theorem 1, we first prove some important lemmas.

A. Lemma 1 & Proof

Lemma 1. *Following Algorithm 1, when x_{t_k} is given, the expectation of $x_{t_{k+1}}$ is unbiased to $\bar{y}_{t_k,I}$, i.e.,*

$$\mathbb{E}(x_{t_{k+1}}|x_{t_k}) = \mathbb{E}_{t_k}(x_{t_{k+1}}) = \bar{y}_{t_k,I}, \tag{30}$$

where $\mathbb{E}_{t_k}$ denotes conditional expectation $\mathbb{E}(\cdot|t_k)$.

Proof. Let $X_i^{t_k}$ be a random variable denoting the number of times the i-th client is selected in the t_k-th round. Then, $X_i^{t_k}$ follows a binomial distribution with parameters M and p_i, i.e., $X_i^{t_k} \sim \text{Binomial}(M, p_i)$. The expectation of $X_i^{t_k}$ is given by $\mathbb{E}[X_i^{t_k}] = Mp_i$.

Since the participants of each round are selected according to probability $\mathbf{p}$ with replacement, the expectation of the aggregated model $\mathbb{E}_{t_k}(x_{t_{k+1}})$ is given by

$$\begin{aligned}\mathbb{E}_{t_k}(x_{t_{k+1}}) &= \mathbb{E}\left(\sum_{i\in\mathcal{M}^{t_k}} \frac{d_i}{Mp_i} y^i_{t_k,I} | x_{t_k}\right) \\ &= \sum_{i=1}^{N} \mathbb{E}[X_i^{t_k}] \frac{d_i}{Mp_i} y^i_{t_k,I} = \sum_{i=1}^{N} d_i y^i_{t_k,I} = \bar{y}_{t_k,I}. \end{aligned}$$ □

B. Lemma 2 & Proof

Lemma 2. *For given x_{t_k}, let $\mathcal{M}^{t_k} = \{i_1, i_2, ..., i_M\} \subset [N]$ in Algorithm 1, where i_l denotes the index of the l-th participant. Then for any i_l, the expectation of $\frac{d_{i_l}}{p_{i_l}} y^{i_l}_{t_k,I}$ is unbiased to $\bar{y}_{t_k,I}$, i.e.,*

$$\mathbb{E}_{t_k}\left(\frac{d_{i_l}}{p_{i_l}} y^{i_l}_{t_k,I}\right) = \bar{y}_{t_k,I}. \tag{31}$$

Proof. For certain i_l in $\mathcal{M}^{t_k}$, it could be any index $a \in [1, N]$ with probability p_a. Therefore,

$$\begin{aligned}\mathbb{E}_{t_k}\left(\frac{d_{i_l}}{p_{i_l}} y^{i_l}_{t_k,I}\right) &= \mathbb{E}_{t_k} \sum_{a=1}^{N} P(i_l = a) \frac{d_a}{p_a} y^a_{t_k,I} \\ &= \mathbb{E}_{t_k} \sum_{a=1}^{N} d_a y^a_{t_k,I} = \bar{y}_{t_k,I}. \end{aligned}$$ □

C. Lemma 3 & Proof

Lemma 3. *Given x_{t_k}, the variance of $x_{t_{k+1}}$ satisfies*

$$\mathbb{E}_{t_k}\left\|x_{t_{k+1}} - \bar{y}_{t_k,I}\right\|^2 \le \frac{I^2\gamma^2}{M} \sum_{i=1}^{N} \frac{d_i^2 G_i^2}{p_i}. \tag{32}$$

Proof.

$$\begin{aligned}&\mathbb{E}_{t_k}\left\|x_{t_{k+1}} - \bar{y}_{t_k,I}\right\|^2 = \mathbb{E}_{t_k}\left\|\sum_{l=1}^{M} \frac{d_{i_l}}{p_{i_l}} y^{i_l}_{t_k,I} - \bar{y}_{t_k,I}\right\|^2 \\ &= \mathbb{E}_{t_k}\left\|\frac{1}{M}\sum_{l=1}^{M}\left(\frac{d_{i_l}}{p_{i_l}} y^{i_l}_{t_k,I} - \bar{y}_{t_k,I}\right)\right\|^2 \\ &\overset{(a)}{=} \frac{1}{M^2}\mathbb{E}_{t_k}\sum_{l=1}^{M}\left\|\frac{d_{i_l}}{p_{i_l}} y^{i_l}_{t_k,I} - \bar{y}_{t_k,I}\right\|^2 \\ &\overset{(b)}{=} \frac{1}{M^2}\sum_{i=1}^{N} Mp_i \mathbb{E}_{t_k}\left\|\frac{d_i}{p_i} y^i_{t_k,I} - \bar{y}_{t_k,I}\right\|^2 \\ &= \frac{1}{M}\sum_{i=1}^{N} p_i \mathbb{E}_{t_k}\left\|\frac{d_i}{p_i} y^i_{t_k,I} - \frac{d_i}{p_i} y^i_{t_k,0} - (\bar{y}_{t_k,I} - \frac{d_i}{p_i} y^i_{t_k,0})\right\|^2 \\ &\overset{(c)}{\le} \frac{1}{M}\sum_{i=1}^{N} p_i \mathbb{E}_{t_k}\left\|\frac{d_i}{p_i} y^i_{t_k,I} - \frac{d_i}{p_i} y^i_{t_k,0}\right\|^2 \\ &= \frac{1}{M}\sum_{i=1}^{N} \frac{d_i^2}{p_i} \mathbb{E}_{t_k}\left\|-\sum_{j=0}^{I-1} \gamma \nabla F_i(y^i_{t_k,j}, \xi_j)\right\|^2 \end{aligned}$$

$$\overset{(d)}{\leq} \frac{1}{M}\sum_{i=1}^{N}\frac{d_i^2}{p_i}\gamma^2 I\sum_{j=0}^{I-1}\mathbb{E}_{t_k}\left\|\nabla F_i(y_{t_k,j}^i,\xi_j)\right\|^2$$
$$\overset{(e)}{\leq} \frac{1}{M}\sum_{i=1}^{N}\frac{d_i^2}{p_i}\gamma^2 I^2 G_i^2 = \frac{I^2\gamma^2}{M}\sum_{i=1}^{N}\frac{d_i^2 G_i^2}{p_i},$$

where (a) uses Lemma 2, (b) takes expectation over client selection, (c) uses $\mathbb{E}\left\|x-\mathbb{E}x\right\|^2 \leq \mathbb{E}\left\|x\right\|^2$, (d) uses Jensen's Inequality, and (e) uses Assumption 3. □

D. Proof of Theorem 1

Proof. According to Assumption 2, for any given x_{t_k}, we have

$$\mathbb{E}_{t_k} f(x_{t_{k+1}}) \leq \mathbb{E}_{t_k}(f(x_{t_k}) + \left\langle x_{t_{k+1}} - x_{t_k}, \nabla f(x_{t_k})\right\rangle + \frac{L}{2}\left\|x_{t_{k+1}} - x_{t_k}\right\|^2). \tag{33}$$

Consider the third term in (33):

$$\frac{L}{2}\mathbb{E}_{t_k}\left\|x_{t_{k+1}} - x_{t_k}\right\|^2$$
$$\overset{(a)}{=} \frac{L}{2}(\mathbb{E}_{t_k}\left\|x_{t_{k+1}} - \bar{y}_{t_k,I}\right\|^2 + \mathbb{E}_{t_k}\left\|\bar{y}_{t_k,I} - x_{t_k}\right\|^2)$$
$$\overset{(b)}{\leq} \frac{LI^2\gamma^2}{2M}\sum_{i=1}^{N}\frac{d_i^2 G_i^2}{p_i} + \frac{L}{2}\mathbb{E}_{t_k}\left\|\bar{y}_{t_k,I} - x_{t_k}\right\|^2, \tag{34}$$

where (a) uses Lemma 1 where $\mathbb{E}_{t_k}(x_{t_{k+1}} - \bar{y}_{t_k,I}) = 0$ and (b) uses Lemma 3.

Consider the second term in (33):

$$\mathbb{E}_{t_k}\left\langle x_{t_{k+1}} - x_{t_k}, \nabla f(x_{t_k})\right\rangle = \mathbb{E}_{t_k}\left\langle \bar{y}_{t_k,I} - x_{t_k}, \nabla f(x_{t_k})\right\rangle$$
$$= -\gamma\mathbb{E}_{t_k}\left\langle \sum_{i=1}^{N} d_i \sum_{j=0}^{I-1}\nabla F_i(y_{t_k,j}^i), \nabla f(x_{t_k})\right\rangle$$
$$\overset{(a)}{=} \frac{\gamma}{2I}\mathbb{E}_{t_k}\left(\left\|\sum_{i=1}^{N} d_i\sum_{j=0}^{I-1}(\nabla F_i(y_{t_k,j}^i) - \nabla F_i(x_{t_k}))\right\|^2 - \left\|\sum_{i=1}^{N} d_i\sum_{j=0}^{I-1}\nabla F_i(y_{t_k,j}^i)\right\|^2 - I^2\left\|\nabla f(x_{t_k})\right\|^2\right)$$
$$\overset{(b)}{=} \frac{\gamma}{2I}\mathbb{E}_{t_k}\left(\left\|\sum_{i=1}^{N} d_i\sum_{j=0}^{I-1}(\nabla F_i(y_{t_k,j}^i) - \nabla F_i(x_{t_k}))\right\|^2 - \frac{1}{\gamma^2}\left\|\bar{y}_{t_k,I} - x_{t_k}\right\|^2 - I^2\left\|\nabla f(x_{t_k})\right\|^2\right), \tag{35}$$

where (a) uses $-\left\langle a,b\right\rangle = \frac{1}{2}(\left\|a-b\right\|^2 - \left\|a\right\|^2 - \left\|b\right\|^2)$ and (b) uses $\mathbb{E}_{t_k}(\bar{y}_{t_k,I} - x_{t_k}) = -\gamma\mathbb{E}_{t_k}\left(\sum_{i=1}^{N} d_i\sum_{j=0}^{I-1}\nabla F_i(y_{t_k,j}^i)\right)$.

The first term in (35) is

$$\mathbb{E}_{t_k}\left\|\sum_{i=1}^{N} d_i\sum_{j=0}^{I-1}(\nabla F_i(y_{t_k,j}^i) - \nabla F_i(x_{t_k}))\right\|^2$$
$$\overset{(a)}{\leq} \mathbb{E}_{t_k}\sum_{i=1}^{N} d_i\sum_{j=0}^{I-1} I\left\|\nabla F_i(y_{t_k,j}^i) - \nabla F_i(x_{t_k})\right\|^2$$
$$\overset{(b)}{=} \mathbb{E}_{t_k}\sum_{i=1}^{N} d_i\sum_{j=1}^{I-1} I\left\|\nabla F_i(y_{t_k,j}^i) - \nabla F_i(x_{t_k})\right\|^2$$
$$\overset{(c)}{\leq} I\sum_{i=1}^{N} d_i\sum_{j=1}^{I-1} L^2\mathbb{E}_{t_k}\left\|y_{t_k,j}^i - x_{t_k}\right\|^2$$
$$\overset{(d)}{=} I\sum_{i=1}^{N} d_i\sum_{j=1}^{I-1} L^2\mathbb{E}_{t_k}\left\|-\sum_{g=0}^{j-1}\gamma\nabla F_i(y_{t_k,g}^i,\xi_g)\right\|^2$$
$$\overset{(e)}{\leq} IL^2\gamma^2\sum_{i=1}^{N} d_i\sum_{j=1}^{I-1}\sum_{g=0}^{j-1}\mathbb{E}_{t_k}\left\|\nabla F_i(y_{t_k,g}^i,\xi_g)\right\|^2 j$$
$$\overset{(f)}{\leq} IL^2\gamma^2\sum_{i=1}^{N} d_i\sum_{j=1}^{I-1}\sum_{g=0}^{j-1} G_i^2 j$$
$$\overset{(g)}{=} \frac{I^2(I-1)(2I-1)L^2\gamma^2}{6}\sum_{i=1}^{N} d_i G_i^2, \tag{36}$$

where (a) and (e) use Jensen's inequality, (b) holds because $\nabla F_i(y_{t_k,0}^i) = \nabla F_i(x_{t_k})$, (c) uses Assumption 2, (d) uses the update rule of SGD, (f) uses Assumption 3, and (g) holds because of $\sum_{j=1}^{I-1} j^2 = \frac{I(I-1)(2I-1)}{6}$.

Plugging (36) back, (35) becomes

$$\mathbb{E}_{t_k}\left\langle x_{t_{k+1}} - x_{t_k}, \nabla f(x_{t_k})\right\rangle \leq \frac{I(I-1)(2I-1)L^2\gamma^3}{12}\sum_{i=1}^{N} d_i G_i^2 - \frac{1}{2I\gamma}\mathbb{E}_{t_k}\left\|\bar{y}_{t_k,I} - x_{t_k}\right\|^2 - \frac{\gamma I}{2}\mathbb{E}_{t_k}\left\|\nabla f(x_{t_k})\right\|^2. \tag{37}$$

Plugging (34) and (37) back in (33), we have

$$\mathbb{E}_{t_k} f(x_{t_{k+1}}) \leq \mathbb{E}_{t_k} f(x_{t_k}) + \frac{I(I-1)(2I-1)L^2\gamma^3}{12}\sum_{i=1}^{N} d_i G_i^2 + (\frac{L}{2} - \frac{1}{2I\gamma})\mathbb{E}_{t_k}\left\|\bar{y}_{t_k,I} - x_{t_k}\right\|^2 - \frac{\gamma I}{2}\mathbb{E}_{t_k}\left\|\nabla f(x_{t_k})\right\|^2 + \frac{LI^2\gamma^2}{2M}\sum_{i=1}^{N}\frac{d_i^2 G_i^2}{p_i}$$
$$\overset{(a)}{\leq} \mathbb{E}_{t_k} f(x_{t_k}) + \frac{I(I-1)(2I-1)L^2\gamma^3}{12}\sum_{i=1}^{N} d_i G_i^2 - \frac{\gamma I}{2}\mathbb{E}_{t_k}\left\|\nabla f(x_{t_k})\right\|^2 + \frac{LI^2\gamma^2}{2M}\sum_{i=1}^{N}\frac{d_i^2 G_i^2}{p_i}, \tag{38}$$

where (a) holds if $\gamma \leq \frac{1}{IL}$. Rearranging (38), we have

$$\mathbb{E}_{t_k}\left\|\nabla f(x_{t_k})\right\|^2 \leq \frac{2}{\gamma I}(\mathbb{E}_{t_k} f(x_{t_k}) - \mathbb{E}_{t_k} f(x_{t_{k+1}})) + \frac{(I-1)(2I-1)L^2\gamma^2}{6}\sum_{i=1}^{N} d_i G_i^2 + \frac{LI\gamma}{M}\sum_{i=1}^{N}\frac{d_i^2 G_i^2}{p_i}. \tag{39}$$

We now have

$$\min_t \mathbb{E}\left\|\nabla f(x_t)\right\|^2 \leq \min_{t_k\in\{0,I,\ldots,(T-1)I\}}\mathbb{E}_{t_k}\left\|\nabla f(x_{t_k})\right\|^2$$

$$
\begin{aligned}
&\leq \frac{1}{T}\sum_{t_k=0}^{(T-1)I}\mathbb{E}_{t_k}\left\|\nabla f(x_{t_k})\right\|^2 \\
&\overset{(a)}{=} \frac{2}{\gamma IT}(f(x_0)-f(x_{TI})) \\
&\quad+\frac{(I-1)(2I-1)L^2\gamma^2}{6}\sum_{i=1}^{N}d_iG_i^2+\frac{LI\gamma}{M}\sum_{i=1}^{N}\frac{d_i^2G_i^2}{p_i} \\
&\leq \frac{2}{\gamma IT}(f(x_0)-f(x^*)) \\
&\quad+\frac{(I-1)(2I-1)L^2\gamma^2}{6}\sum_{i=1}^{N}d_iG_i^2+\frac{LI\gamma}{M}\sum_{i=1}^{N}\frac{d_i^2G_i^2}{p_i},
\end{aligned}
\tag{40}
$$

where (a) uses the telescoping sum of (39). □

APPENDIX B
PROOF OF COROLLARY 2

Proof. According to Corollary 1, we have

$$
\begin{aligned}
\min_t E\left\|\nabla f(x_t)\right\|^2 \leq &\frac{1}{\sqrt{T}}\left(2L\Gamma+\frac{1}{M}\sum_{i=1}^{N}\frac{d_i^2G_i^2}{p_i}\right) \\
&+\frac{1}{T}\frac{(I-1)(2I-1)}{6I^2}\sum_{i=1}^{N}d_iG_i^2.
\end{aligned}
$$

To make the first term dominate, we simply let

$$
\frac{1}{\sqrt{T}}\left(2L\Gamma+\frac{1}{M}\sum_{i=1}^{N}\frac{d_i^2G_i^2}{p_i}\right)\geq\frac{1}{T}\frac{(I-1)(2I-1)}{6I^2}\sum_{i=1}^{N}d_iG_i^2,
$$

i.e.,

$$
\sqrt{T}\geq\left(\frac{(I-1)(2I-1)}{6I^2}\sum_{i=1}^{N}d_iG_i^2\right)/\left(2L\Gamma+\frac{1}{M}\sum_{i=1}^{N}\frac{d_i^2G_i^2}{p_i}\right).
$$

Let B denote the RHS of the above inequality, we have

$$
\begin{aligned}
B &\overset{(a)}{\leq} \left(\frac{(I-1)(2I-1)}{6I^2}\sum_{i=1}^{N}d_iG_i^2\right)/\left(\frac{1}{M}\sum_{i=1}^{N}\frac{d_i^2G_i^2}{p_i}\right) \\
&\overset{(b)}{\leq} \left(\frac{(I-1)(2I-1)}{6I^2}\max_i G_i^2\right)/\left(\frac{1}{M}\min_i d_iG_i^2\right) \\
&< \frac{M}{3}\frac{\max_i G_i^2}{\min_i d_iG_i^2}=\frac{M}{3}\rho,
\end{aligned}
$$

where (a) uses $2L\Gamma\geq 0$; (b) uses the following inequalities: $\sum_{i=1}^{N}d_iG_i^2\leq\max_i G_i^2$, $\sum_{i=1}^{N}\frac{d_i^2G_i^2}{p_i}\geq\min_i d_iG_i^2$; $\rho=\frac{\max_i G_i^2}{\min_i d_iG_i^2}$.

As a result, $\sqrt{T}\geq B$ when $T\geq\left\lceil\frac{M^2\rho^2}{9}\right\rceil$. □

REFERENCES

[1] X. Cao, T. Başar, S. Diggavi, Y. C. Eldar, K. B. Letaief, H. V. Poor, and J. Zhang, "Communication-efficient distributed learning: An overview," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 4, pp. 851–873, 2023.

[2] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "A novel framework for the analysis and design of heterogeneous federated learning," *IEEE Trans. Signal Process.*, vol. 69, pp. 5234–5249, 2021.

[3] A. Reisizadeh, A. Mokhtari, H. Hassani, A. Jadbabaie, and R. Pedarsani, "FedPAQ: A communication-efficient federated learning method with periodic averaging and quantization," in *Proc. Int. Conf. Artificial Intelligence and Statistics (AISTATS)*, 2020, pp. 2021–2031.

[4] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "QSGD: Communication-efficient SGD via gradient quantization and encoding," in *Proc. Conf. Neural Inf. Process. Syst. (NeurIPS)*, vol. 30, 2017, pp. 1707–1718.

[5] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-IID data," *IEEE Trans. Neural Networks and Learning Syst.*, vol. 31, no. 9, pp. 3400–3413, 2019.

[6] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, and A. Anandkumar, "signSGD: Compressed optimisation for non-convex problems," in *Proc. Int. Conf. Machine Learning (ICML)*, 2018, pp. 560–569.

[7] D. Alistarh, T. Hoefler, M. Johansson, N. Konstantinov, S. Khirirat, and C. Renggli, "The convergence of sparsified gradient methods," in *Proc. Conf. Neural Inf. Process. Syst. (NeurIPS)*, vol. 31, 2018, pp. 5977–5987.

[8] S. U. Stich, J.-B. Cordonnier, and M. Jaggi, "Sparsified SGD with memory," in *Proc. Conf. Neural Inf. Process. Syst. (NeurIPS)*, vol. 31, 2018, pp. 4452–4463.

[9] D. Basu, D. Data, C. Karakus, and S. Diggavi, "Qsparse-local-SGD: Distributed SGD with quantization, sparsification and local computations," in *Proc. Conf. Neural Inf. Process. Syst. (NeurIPS)*, vol. 32, 2019.

[10] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. Conf. Machine Learning and Syst. (MLSys)*, vol. 2, 2020, pp. 429–450.

[11] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "SCAFFOLD: Stochastic controlled averaging for federated learning," in *Proc. Int. Conf. Machine Learning (ICML)*, 2020, pp. 5132–5143.

[12] S. Wang and M. Ji, "A unified analysis of federated learning with arbitrary client participation," *arXiv:2205.13648*, 2022.

[13] L. Yu, R. Albelaihi, X. Sun, N. Ansari, and M. Devetsikiotis, "Jointly optimizing client selection and resource management in wireless federated learning for internet of things," *IEEE Internet of Things J.*, vol. 9, no. 6, pp. 4385–4395, 2022.

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

[15] W. Chen, S. Horvath, and P. Richtarik, "Optimal client sampling for federated learning," *Trans. Machine Learning Research*, 2022.

[16] X. Zeng, M. Yan, and M. Zhang, "Mercury: Efficient on-device distributed dnn training via stochastic importance sampling," in *Proc. ACM Conf. Embedded Networked Sensor Systems*, 2021, p. 29–41.

[17] H. Cao, Q. Pan, Y. Zhu, and J. Liu, "Birds of a feather

help: Context-aware client selection for federated learning," in *Proc. Int. Workshop Trustable, Verifiable and Auditable Federated Learning with AAAI*, 2022.

[18] F. Lai, X. Zhu, H. V. Madhyastha, and M. Chowdhury, "Oort: Efficient federated learning via guided participant selection." in *Proc. USENIX Symp. Operating Syst. Design Implement. (OSDI)*, 2021, pp. 19–35.

[19] H. T. Nguyen, V. Sehwag, S. Hosseinalipour, C. G. Brinton, M. Chiang, and H. Vincent Poor, "Fast-convergent federated learning," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 201–218, 2021.

[20] L. WANG, W. WANG, and B. LI, "Cmfl: Mitigating communication overhead for federated learning," in *Proc. IEEE Int. Conf. Distrib. Comput. Sys. (ICDCS)*, 2019, pp. 954–964.

[21] 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," *IEEE Trans. Wireless Commun.*, vol. 20, no. 6, pp. 3643–3658, 2021.

[22] Y. Jee Cho, J. Wang, and G. Joshi, "Towards understanding biased client selection in federated learning," in *Proc. Int. Conf. Artificial Intelligence and Statistics (AISTATS)*, ser. Proceedings of Machine Learning Research, G. Camps-Valls, F. J. R. Ruiz, and I. Valera, Eds., vol. 151. PMLR, 28–30 Mar 2022, pp. 10 351–10 375.

[23] L. Fu, H. Zhang, G. Gao, M. Zhang, and X. Liu, "Client selection in federated learning: Principles, challenges, and opportunities," *IEEE Internet of Things J.*, pp. 1–1, 2023.

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

[25] J. Shin, Y. Li, Y. Liu, and S.-J. Lee, "Fedbalancer: Data and pace control for efficient federated learning on heterogeneous clients," in *Proc. Annu. Int. Conf. Mobile Sys., Appl. Services*, 2022, p. 436–449.

[26] 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, 2020.

[27] C. Keçeci, M. Shaqfeh, F. Al-Qahtani, M. Ismail, and E. Serpedin, "Clustered scheduling and communication pipelining for efficient resource management of wireless federated learning," *IEEE Internet of Things J.*, vol. 10, no. 15, pp. 13 303–13 316, 2023.

[28] 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, 2020.

[29] B. Luo, W. Xiao, S. Wang, J. Huang, and L. Tassiulas, "Tackling system and statistical heterogeneity for federated learning with adaptive client sampling," in *Proc. IEEE Conf. Computer Commun. (INFOCOM)*, 2022, pp. 1739–1748.

[30] J. Konečnỳ, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," *arXiv:1610.02527*, 2016.

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

[32] Y. Fraboni, R. Vidal, L. Kameni, and M. Lorenzi, "Clustered sampling: Low-variance and improved representativity for clients selection in federated learning," in *Proc. Int. Conf. Machine Learning (ICML)*, 2021, pp. 3407–3416.

[33] C. Lu, W. Ma, R. Wang, S. Deng, and Y. Wu, "Federated learning based on stratified sampling and regularization," *Complex Intell. Syst.*, vol. 9, no. 2, pp. 2081–2099, 2023.

[34] B. Luo, X. Li, S. Wang, J. Huang, and L. Tassiulas, "Cost-effective federated learning in mobile edge networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3606–3621, 2021.

[35] M. L. Pinedo, *Scheduling: Theory, Algorithms, and Systems*, 3rd ed. Springer Publishing Company, Incorporated, 2008.

[36] F. Haddadpour and M. Mahdavi, "On the convergence of local descent methods in federated learning," *CoRR*, vol. abs/1910.14425, 2019.

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

[38] W. W. Hager and H. Zhang, "An active set algorithm for nonlinear optimization with polyhedral constraints," *Science China Mathematics*, vol. 59, no. 8, pp. 1525–1542, 2016.

[39] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis *et al.*, "Tensorflow: a system for large-scale machine learning," in *Proc. USENIX Symp. Operating Syst. Design Implement. (OSDI)*, vol. 16, no. 19, 2016, pp. 265–283.

[40] G. Cohen, S. Afshar, J. Tapson, and A. Van Schaik, "EMNIST: Extending MNIST to handwritten letters," in *Proc. Int. Joint Conf. Neural Networks (IJCNN)*, 2017, pp. 2921–2926.

[41] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms," *arXiv:1708.07747*, 2017.

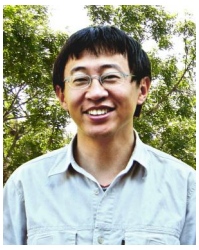
[42] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[43] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

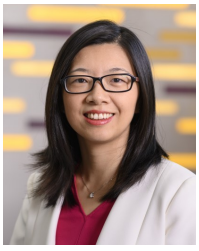
This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3425757

Xiaobing Chen received the B.E. degree in electrical engineering and M.E. degree in control science and engineering from University of Electronic Science and Technology of China, Chengdu, China, in 2017 and 2020, respectively. He is currently pursuing the Ph.D. degree in the Division of Electrical and Computer Engineering, Louisiana State University, where he joined as a Graduate Research Assistant in 2021. His research interests include federated learning, privacy in machine learning, and optimization theory.

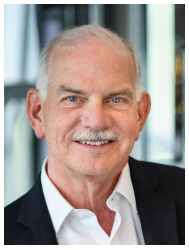
Xiangwei Zhou received the B.S. degree in communication engineering from Nanjing University of Science and Technology, Nanjing, China, the M.S. degree in information and communication engineering from Zhejiang University, Hangzhou, China, and the Ph.D. degree in electrical and computer engineering from Georgia Institute of Technology, Atlanta, GA, USA, in 2005, 2007, and 2011, respectively. He is currently an Associate Professor with the Division of Electrical and Computer Engineering, Louisiana State University, Baton Rouge, LA, USA. His research interests include wireless communications and statistical signal processing, with current emphasis on coexistence of wireless systems, Internet of Things, and machine learning for intelligent communications. He was the recipient of the Best Paper Award at the 2014 International Conference on Wireless Communications and Signal Processing and served as an Editor for the IEEE Transactions on Wireless Communications from 2013 to 2018.



Hongchao Zhang received the B.S. degree in Mathematics from Shandong University, China, in 1998, the M.S. degree from the Computing Center, Chinese Academy of Sciences, in 2001, and the Ph.D. degree in Mathematics from the University of Florida, Gainesville, FL, USA, in 2006. He is currently a Professor in the Department of Mathematics and the Center for Computation and Technology (CCT) at Louisiana State University, Baton Rouge, LA, USA. His research interests include nonlinear optimization theory and its applications, sparse matrix computing, graph partitioning, stochastic optimization algorithms and applications, and numerical linear algebra.



Mingxuan Sun received the B.S. degree in computer science and engineering from Zhejiang University, Hangzhou, China in 2004, the M.S. degree in computer science from University of Kentucky, Lexington, KY, USA in 2006, and the Ph.D. degree in computer science from Georgia Institute of Technology, Atlanta, GA, USA in 2012. She is currently an Associate Professor with the Division of Computer Science and Engineering, Louisiana State University, Baton Rouge, LA, USA. Her research interests include machine learning, information retrieval, and data mining. She is also interested in machine learning and AI applications in social informatics, security, and wireless communications. She has published research papers in leading journals and conferences including PAMI, JMLR, NeurIPS, AAAI, AISTATS, KDD, ICDM, WWW, and WSDM.



H. Vincent Poor received the Ph.D. degree in EECS from Princeton University in 1977. From 1977 until 1990, he was on the faculty of the University of Illinois at Urbana Champaign. Since 1990 he has been on the faculty at Princeton, where he is currently the Michael Henry Strater University Professor. During 2006 to 2016, he served as the dean of Princeton's School of Engineering and Applied Science. He has also held visiting appointments at several other universities, including most recently at Berkeley and Cambridge. His research interests are in the areas of information theory, machine learning and network science, and their applications in wireless networks, energy systems and related fields. Among his publications in these areas is the recent book Machine Learning and Wireless Communications (Cambridge University Press, 2022). Dr. Poor is a member of the National Academy of Engineering and the National Academy of Sciences and is a foreign member of the Chinese Academy of Sciences, the Royal Society, and other national and international academies. He received the IEEE Alexander Graham Bell Medal in 2017.