Tackling System Induced Bias in Federated Learning: Stratification and Convergence Analysis

Ming Tang*, Vincent W.S. Wong†
*Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China
†Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, Canada
e-mail: *tangm3@sustech.edu.cn, †vincentw@ece.ubc.ca

***Abstract*—In federated learning, clients cooperatively train a global model by training local models over their datasets under the coordination of a central server. However, clients may sometimes be unavailable for training due to their network connections and energy levels. Considering the highly non-independent and identically distributed (non-IID) degree of the clients' datasets, the local models of the available clients being sampled for training may not represent those of all other clients. This is referred as system induced bias. In this work, we quantify the system induced bias due to time-varying client availability. The theoretical result shows that this bias occurs independently of the number of available clients and the number of clients being sampled in each training round. To address system induced bias, we propose a FedSS algorithm by incorporating stratified sampling and prove that the proposed algorithm is unbiased. We quantify the impact of system parameters on the algorithm performance and derive the performance guarantee of our proposed FedSS algorithm. Theoretical and experimental results on CIFAR-10 and MNIST datasets show that our proposed FedSS algorithm outperforms several benchmark algorithms by up to 5.1 times in terms of the algorithm convergence rate.**

***Index Terms*—Federated learning, system induced bias, stratified sampling**

I. Introduction

Federated learning (FL) [1], [2] is a decentralized machine learning approach, which enables a large number of clients to cooperatively train a global model using their local datasets. In FL, a central server maintains the global model and coordinates the training process for multiple training rounds. In each training round, some clients are sampled to perform training over the global model using their local datasets. The local model updates are then sent to the central server for global model update. During this process, clients do not need to send their local datasets to any central entity. Thus, the cost for data transmission (e.g., energy, bandwidth resources) can be reduced, and data privacy can be preserved. Federated averaging (FedAvg) algorithm [1] is one of the state-of-the-art FL algorithms. Other FL algorithms have also been proposed to address various issues such as the communication cost [3], [4], system and statistical heterogeneity [5]–[7], privacy [8], fairness [9], model compression [10], model retraining [11], and incentive mechanism design [12]. Some recent works, e.g., [2], [13], [14], have provided comprehensive survey on FL.

This work was supported by the National Natural Science Foundation of China under Grant 62202214, Natural Sciences and Engineering Research Council of Canada, and the Digital Research Alliance of Canada (alliance-can.ca).

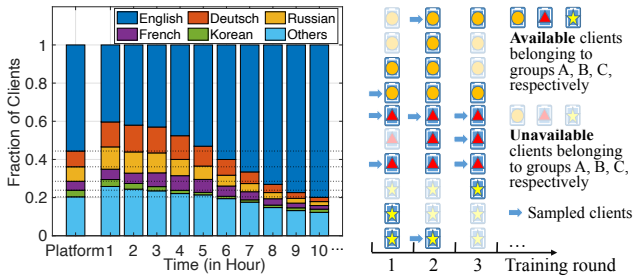

Fig. 1. (a) A real-world example of time-varying client availability on a live streaming platform; (b) An example to illustrate system induced bias.

There are two important facts related to the clients in FL. First, the clients' datasets are typically non-independent and identically distributed (non-IID). This is due to various factors including the geographical locations and personal preferences of the clients. We define a *group* as a set of clients whose datasets are approximately IID, e.g., clients from a geographical location with similar personal preference. Second, only a small subset of clients (e.g., 0.1%–10%) may be available for training in each training round due to various constraints (e.g., network conditions, energy levels) [2]. Meanwhile, the set of available clients may change over time. We refer to this as *time-varying client availability*. For example, Fig. 1(a) shows real-world streamers' data on a live streaming platform [15]. Streamers can act as clients for FL tasks (e.g., gesture prediction) when they are available. In this example, we define a group as a set of clients streaming with a particular language (e.g., English). Thus, the clients in the same group may have similar background and behaviors than those from other groups. The first bar entitled "Platform" shows the fraction of clients belonging to each group. The remaining bars (i.e., 1, 2, ...) show the fraction of available clients in each group at a particular time. Fig. 1(a) shows that the availability of clients in different groups can change significantly over time.

Although some of the existing works have considered the non-IID datasets (e.g., [3]–[11]), they do not take into account the time-varying client availability and the resulting *system induced bias*. The system induced bias occurs when the datasets of the sampled clients do not represent those of the entire population (i.e., both available and unavailable clients). Consider Fig. 1(b) as an example. Suppose the central

server samples three clients from the set of available clients in each training round. In the training rounds when those sampled clients belong to either one or two groups out of those three groups (e.g., training rounds 1 and 3), the global model obtained based on the local models of those sampled clients will be biased and may not fit the datasets of clients from the other groups in the system.

System induced bias is an important open problem [2] and some existing works have proposed approaches to tackle this issue. Perazzone *et al.* in [16] considered intermittent connectivity of clients and proposed an algorithm for determining the selection probability of each client. Buyukates *et al.* in [17] considered time-varying client availability and analyzed the average age of information of clients' local models. Xia *et al.* in [18] considered unknown future client availability and proposed an online algorithm called CS-UCB-Q to improve fairness among clients, i.e., to ensure that each client participates in a certain proportion of training rounds. Huang *et al.* in [19] proposed an online Lyapunov optimization algorithm to guarantee long-term fairness among clients. Ribero *et al.* in [20] proposed a client sampling strategy to minimize the variance of the participation rates of the clients. Avdiukhin *et al.* in [21] proposed an asynchronous FL algorithm to address temporal client unavailability. Chen *et al.* in [22] proposed an asynchronous online FL algorithm, where the local models of those clients that are available less frequently are assigned with higher weights. The aforementioned works address the system induced bias by making clients contribute approximately equally to the system (e.g., being sampled with similar frequency) in the long run, while the scheduling is independent of the datasets of the clients. Using these approaches, the trained global model in one training round may still be biased. Thus, the global model may vary significantly across training rounds, which may slow down the convergence rate.

In this work, we focus on time-varying client availability and aim to address the system induced bias by guaranteeing the client sampling be unbiased. We regard FedAvg algorithm [1] as a benchmark[1] and aim to answer the following questions:

- How do we quantify the system induced bias of FedAvg algorithm under time-varying client availability?
- How do we address the system induced bias?
- How much is the performance improvement?

Answering the first question is not straightforward. We first introduce an equivalent scheme which determines the same global model as FedAvg algorithm. To quantify system induced bias, we use the bias of the global model with respect to the local models of clients and time-varying client availability. To answer the second question, we incorporate stratified sampling and propose an algorithm, called FedSS. The main idea is to sample clients based on their data statistics and to ensure that the datasets of the sampled clients can represent those of the entire population. Although such an idea has been suggested by Kairouz *et al.* in [2, Section 7.2.3], they did not provide theoretical analysis. Shen *et al.* in [25] analyzed how stratified sampling can address the non-IID datasets of clients. However, they did not consider time-varying client availability.

Answering the third question is challenging. This is because the study of this system requires us to quantify how the system factors (e.g., data statistics of different groups) affect the algorithm performance via the interaction between the central server and clients as well as the selection of sampling schemes. We overcome the challenges and provide rigorous proof to derive the theoretical performance guarantee. We derive the client allocation scheme (i.e., number of clients sampled in each group) that optimizes the algorithm performance.

We summarize our main contributions as follows:

- We quantify the system induced bias of FedAvg algorithm under time-varying client availability. Our analytical results show that as long as the number of available clients of an arbitrary group is not proportional to the number of clients of that group (see the example in Fig. 1(a)), such bias always exists no matter how much we increase the number of available clients and sampled clients.
- To address the system induced bias, we propose FedSS algorithm by incorporating stratified sampling. We prove that FedSS algorithm is unbiased. Moreover, we theoretically derive its performance guarantee and propose an optimal client allocation scheme. Theoretical results show that our algorithm performance does not depend on the time-varying client availability, which is ideal. When compared with FedAvg algorithm, the performance improvement of our proposed FedSS algorithm is linearly increasing with the non-IID degree of the clients' data.
- We conduct experiments using MNIST [26] and CIFAR-10 [27] datasets. To verify that our proposed approach can be extended to the variants of FedAvg algorithm, we incorporate stratified sampling into FedProx algorithm [23], where this extended algorithm is called FedProxSS. The results show that both of our proposed FedSS and FedProxSS algorithms have a faster convergence rate than FedAvg [1], FedProx [23], and CS-UCB-Q [18] algorithms. The improvement can be up to 5.1 times when the non-IID degree of the clients' data is high.

This paper is organized as follows. We present the system model in Section II. Then, we quantify the bias and propose FedSS algorithm in Section III. In Section IV, we analyze the performance guarantee. We present experimental results in Section V. Conclusions are drawn in Section VI.

II. System Model

We consider N clients, denoted by set $\mathcal{N} = \{1, 2, \ldots, N\}$. Each client $n \in \mathcal{N}$ has a local dataset $\mathcal{D}_n$. Let D_n denote the number of data samples in dataset $\mathcal{D}_n$, i.e., $D_n = |\mathcal{D}_n|$. We consider a supervised learning task. Each data sample contains a feature $\boldsymbol{x}$ and a label $\boldsymbol{y}$. Different clients may have different empirical distributions in terms of their data samples. For example, clients may live in different geographic regions and have different personal preferences [2, Section

[1]The analysis in this paper can be extended for analyzing most of the synchronous FL algorithms. Typical examples include the variants of FedAvg algorithm, e.g., FedProx [23] and Scaffold [24].

3.1]. For mathematical simplicity, we consider K groups of clients and assume that data samples of an arbitrary client from group $k \in \mathcal{K} \triangleq \{1, 2, \ldots, K\}$ are generated from distribution $P_k(\boldsymbol{x}, \boldsymbol{y})$. Let $\mathcal{N}_k \subset \mathcal{N}$ denote the set of clients of group $k \in \mathcal{K}$. Since each client belongs to exactly one group, we have $\cup_{k\in\mathcal{K}}\mathcal{N}_k = \mathcal{N}$ and $\cap_{k\in\mathcal{K}}\mathcal{N}_k = \emptyset$. We consider the setting that the central server knows which group each client belongs to. When the central server does not have such information, it can estimate the group of a client based on certain statistics of clients' datasets (see Section V-A).

In the following, we introduce the FL process among clients. Then, we present the performance metric for FL algorithm.

A. FL Process with Time-Varying Client Availability

In FL, clients cooperatively train a global model under the coordination of a central server. We consider synchronous FL algorithms, with FedAvg algorithm [1] and its variants [23], [24] as typical examples. Suppose there are T training rounds. At the beginning of the FL process, the central server first initializes the parameter vector (i.e., the weights of the neurons and the biases between neurons) of the global model as $\boldsymbol{\omega}_0$.

We consider *time-varying client availability*. For example, at any time, a client may be unavailable for training if it is disconnected from the network, busy in other computationally intensive tasks, or low in battery power. In training round $t \in \mathcal{T} \triangleq \{1, 2, \ldots, T\}$, let $\mathcal{A}_t^k \subseteq \mathcal{N}_k$ denote the set of available clients in group k. We assume that set $\mathcal{A}_t^k$ remains unchanged in training round t, and the central server is aware of the set of available clients $\mathcal{A}_t^k$ for $k \in \mathcal{K}$.[2] For analytical simplicity, we assume that each client $n \in \mathcal{N}_k$ belongs to set $\mathcal{A}_t^k$ with equal probability for $t \in \mathcal{T}$.[3]

In training round t, the central server first samples S clients from the available clients in set $\mathcal{A}_t \triangleq \cup_{k\in\mathcal{K}}\mathcal{A}_t^k$. Let $\mathcal{S}_t \subseteq \mathcal{A}_t$ denote the set of sampled clients in training round t. In FedAvg algorithm [1] and its variants [23], [24], clients are sampled using *simple random sampling*, i.e., clients in set $\mathcal{A}_t$ are randomly sampled with equal probability. Then, the central server sends the global model $\boldsymbol{\omega}_{t-1}$ determined in training round $t-1$ to the sampled clients. Upon receiving $\boldsymbol{\omega}_{t-1}$, client $n \in \mathcal{S}_t$ performs training for E epochs over its local dataset. Let $\boldsymbol{\omega}_t^n$ denote the local model of client $n \in \mathcal{S}_t$ in training round t, which is initialized to be $\boldsymbol{\omega}_{t-1}$. In each epoch, each sampled client randomly partitions its dataset into M mini-batches and performs stochastic gradient descent (SGD) steps:

$$\boldsymbol{\omega}_t^n := \boldsymbol{\omega}_t^n - \eta \nabla f_n(\boldsymbol{\omega}_t^n; \xi_n), \tag{1}$$

where η and ξ_n denote the step size and mini-batch of client n, respectively, and $f_n(\boldsymbol{\omega}_t^n; \xi_n)$ denotes the loss function of the model with parameter vector $\boldsymbol{\omega}_t^n$ given the mini-batch ξ_n. The operator $:=$ corresponds to assignment.

[2]This can be achieved through one of the following ways. First, clients may reserve their available time period *a priori* (e.g., midnight). Alternatively, when a client is available, it can send a message to the central server.

[3]For the scenario where clients $n \in \mathcal{N}_k$ belong to set $\mathcal{A}_t^k$ with different probabilities, we can partition each group into multiple sub-groups, where each sub-group contains the clients with similar probability of being available in each training round. Then, the theoretical analysis in this work still holds.

After E epochs, each sampled client $n \in \mathcal{S}_t$ uploads its local model $\boldsymbol{\omega}_t^n$ to the central server. Then, the central server updates the global model based on a predefined function, i.e.,

$$\boldsymbol{\omega}_t = G(\{\boldsymbol{\omega}_t^n, n \in \mathcal{S}_t\}), \tag{2}$$

where function $G(\cdot)$ has different expressions in different FL algorithms [1], [23], [24]. In FedAvg algorithm [1],

$$G(\{\boldsymbol{\omega}_t^n, n \in \mathcal{S}_t\}) = \sum_{n\in\mathcal{S}_t} \frac{D_n \boldsymbol{\omega}_t^n}{\sum_{n'\in\mathcal{S}_t} D_{n'}}. \tag{3}$$

B. Performance Metric

We use $f_n(\boldsymbol{\omega}_T) \triangleq \mathbb{E}_{\xi_n}[f_n(\boldsymbol{\omega}_T; \xi_n)]$ to characterize the degree that the global model $\boldsymbol{\omega}_T$ fits the data of client n [1], i.e., the expected loss of the global model with parameter vector $\boldsymbol{\omega}_T$ over the data of client n. We use $F(\boldsymbol{\omega}_T)$ to characterize the degree that $\boldsymbol{\omega}_T$ fits the data of all clients [1],

$$F(\boldsymbol{\omega}_T) \triangleq \sum_{n\in\mathcal{N}} \frac{D_n}{\sum_{n'\in\mathcal{N}} D_{n'}} f_n(\boldsymbol{\omega}_T). \tag{4}$$

We denote $\boldsymbol{\omega}^\star$ as the parameter vector of the global model that minimizes $F(\boldsymbol{\omega}_T)$, i.e., $\boldsymbol{\omega}^\star = \arg\min_{\boldsymbol{\omega}_T} F(\boldsymbol{\omega}_T)$.

The objective is to minimize the *precision* of the trained global model, i.e., the difference between the expected loss of the trained global model $\mathbb{E}_{\boldsymbol{\omega}_T}[F(\boldsymbol{\omega}_T)]$ and the minimum loss $F(\boldsymbol{\omega}^\star)$ [24], [28]. Note that a smaller precision implies that the trained global model better fits the data of all clients.

III. System Induced Bias and FedSS Algorithm

In most of the existing FL algorithms, the central server samples clients using simple random sampling, and the local models are aggregated with weights that are independent of the groups that the clients belong to, e.g., in (3). However, when we consider a scenario with time-varying client availability, such approaches for sampling and model aggregation can lead to system induced bias. In the following, we first introduce an equivalent scheme of the FL process and the notations for analysis. Then, we derive the system induced bias. Finally, we provide a solution to address the bias. For analytical simplicity, similar to [28], we assume that clients have the same number of data samples, i.e., $D_n = D_{n'}$ for $n, n' \in \mathcal{N}$. We will relax this assumption in Section V.

A. Equivalent Scheme and Notations

We introduce an equivalent scheme to the FL process presented in Section II, i.e., both of them lead to the same global model $\boldsymbol{\omega}_T$ given the realization of client and mini-batch sampling. This scheme is introduced only for the understanding of system induced bias and will *not* be considered in practical systems.

Equivalent Scheme. *The central server sends $\boldsymbol{\omega}_{t-1}$ to all clients in set $\mathcal{N}$ for local training. When local training is completed, only the sampled clients in set $\mathcal{S}_t$ send their local models to the central server for global model update.*

Recall that $\mathcal{S}_t \subseteq \mathcal{A}_t$, which contains only available clients. This equivalent scheme is introduced for theoretical analysis,

with which we can quantify the difference between the local models of the entire population and those of the sampled clients. Let $\mathcal{I} \triangleq \{1, \ldots, TEM\}$ denote the set of SGD steps. The SGD step over the m^{th} mini-batch in the e^{th} epoch of training round $t \in \mathcal{T}$ is denoted by the i^{th} SGD step, where $i = (t-1)EM + (e-1)M + m$. Global synchronization corresponds to the event when clients send their local models to the central server for model aggregation. We define set $\mathcal{I}_G \triangleq \{tEM \mid t = 1, \ldots, T\}$. After the i^{th} SGD step, where $i \in \mathcal{I}_G$, global synchronization is performed.

We use $\boldsymbol{\omega}_i^n$ to denote the parameter vector of the local model of client n after the i^{th} SGD step. Let $\boldsymbol{v}_i^n$ denote the parameter vector of the local model of client n after global synchronization (if required). Thus,

$$\boldsymbol{\omega}_i^n = \boldsymbol{v}_{i-1}^n - \eta_{i-1} \nabla f_n(\boldsymbol{v}_{i-1}^n; \xi_{i-1}^n), \ i \in \mathcal{I}, \tag{5}$$

where η_{i-1} and ξ_{i-1}^n are the step size and mini-batch in the i^{th} SGD step, respectively, and

$$\boldsymbol{v}_i^n = \begin{cases} \boldsymbol{\omega}_i^n, & i \in \mathcal{I} \setminus \mathcal{I}_G, \\ G(\{\boldsymbol{\omega}_i^n, n \in \mathcal{S}_{\tau(i)}\}), & i \in \mathcal{I}_G, \end{cases} \tag{6}$$

where $\tau(i) \triangleq \lfloor i/(EM) \rfloor \in \mathcal{T}$ denotes the index of training round that the i^{th} SGD step belongs to.

B. System Induced Bias under FedAvg Algorithm

Based on the equivalent scheme, we now derive the system induced bias of FedAvg algorithm. Let $A_t^k \triangleq |\mathcal{A}_t^k|$ and $A_t \triangleq \sum_{k \in \mathcal{K}} A_t^k$. For fairness of comparison, we assume that $A_t^k \geq S$ for all $k \in \mathcal{K}$, $t \in \mathcal{T}$. That is, the central server is able to sample any arbitrary number of clients from any group (ranging from zero to S) across training rounds. This assumption is reasonable because in practical systems, S and A_t are usually $50 - 5000$ and $10^5 - 10^7$, respectively [2].

Let $\bar{\boldsymbol{\omega}}_i \triangleq \sum_{n \in \mathcal{N}} \boldsymbol{\omega}_i^n / N$ and $\bar{\boldsymbol{v}}_i \triangleq \sum_{n \in \mathcal{N}} \boldsymbol{v}_i^n / N$ denote the average values of $\boldsymbol{\omega}_i^n$ and $\boldsymbol{v}_i^n$ for all $n \in \mathcal{N}$ in the i^{th} SGD step, respectively. We quantify the system induced bias using $\mathbb{E}_{\mathcal{S}_{\tau(i)}}[\bar{\boldsymbol{v}}_i] - \bar{\boldsymbol{\omega}}_i$ for $i \in \mathcal{I}_G$, i.e., the difference between the global model after global synchronization and the average parameter vector of local models of the entire population.

Lemma 1 (Bias)**.** *For any SGD step $i \in \mathcal{I}_G$, FedAvg algorithm with simple random sampling leads to the following bias:*

$$\mathbb{E}_{\mathcal{S}_{\tau(i)}}[\bar{\boldsymbol{v}}_i] - \bar{\boldsymbol{\omega}}_i = \sum_{k \in \mathcal{K}} \left(\left(\frac{A_{\tau(i)}^k}{N_k A_{\tau(i)}} - \frac{1}{N} \right) \sum_{n \in \mathcal{N}_k} \boldsymbol{\omega}_i^n \right). \tag{7}$$

Proof. Under the assumption that $D_n = D_{n'}$ for $n, n' \in \mathcal{N}$, for any $i \in \mathcal{I}_G$, we have $\mathbb{E}_{\mathcal{S}_{\tau(i)}}[\bar{\boldsymbol{v}}_i] = \mathbb{E}_{\mathcal{S}_{\tau(i)}}[\sum_{n \in \mathcal{S}_{\tau(i)}} \boldsymbol{\omega}_i^n / S]$ based on the definition of $\bar{\boldsymbol{v}}_i$ and equations (3) and (6). Thus,

$$\mathbb{E}_{\mathcal{S}_{\tau(i)}}[\bar{\boldsymbol{v}}_i] = \frac{1}{S} \sum_{k \in \mathcal{K}} \sum_{n \in \mathcal{N}_k} \mathbb{P}(n \in \mathcal{S}_{\tau(i)} \mid n \in \mathcal{N}_k) \boldsymbol{\omega}_i^n, \tag{8}$$

where $\mathbb{P}(n \in \mathcal{S}_{\tau(i)} \mid n \in \mathcal{N}_k)$ denotes the probability that a client in set $\mathcal{N}_k$ is sampled in training round $\tau(i)$. We have assumed that any client $n \in \mathcal{N}_k$ belongs to set $\mathcal{A}_t^k$ with equal probability. Thus, for $n \in \mathcal{N}_k$, we have $\mathbb{P}(n \in \mathcal{S}_{\tau(i)} \mid n \in \mathcal{N}_k) = (A_{\tau(i)}^k / N_k)(S / A_{\tau(i)})$. That is, the probability that client n is available multiplied by the probability that this client is sampled. By substituting (8) and the expression of $\bar{\boldsymbol{\omega}}_i$, we obtain (7). □

According to Lemma 1, we have the following observations.

Remark 1 (Bias)**.** *For any $i \in \mathcal{I}_G$, if there exists $k \in \mathcal{K}$ such that $A_{\tau(i)}^k / A_{\tau(i)} \neq N_k / N$, then the equality $\mathbb{E}_{\mathcal{S}_{\tau(i)}}[\bar{\boldsymbol{v}}_i] - \bar{\boldsymbol{\omega}}_i = 0$ does not always hold, i.e., the system induced bias exists.*

Remark 2 (Impact of S)**.** *This bias is independent of the number of clients sampled in each training round, i.e., S.*

Remark 3 (Impact of $A_{\tau(i)}$)**.** *Suppose we fix $A_{\tau(i)}^k / A_{\tau(i)}$ to be a constant. The bias is independent of the total number of available clients, i.e., $A_{\tau(i)}$.*

In practical systems, the ratio $A_{\tau(i)}^k / A_{\tau(i)}$ may be very different from N_k / N and can change significantly over time (see the real-world example in Fig. 1(a)). In this case, system induced bias always exists no matter how much we increase either the number of available clients or sampled clients.

C. FedSS Algorithm

In this section, we propose our FedSS algorithm, which is an FL algorithm that can tackle the aforementioned bias. Our FedSS algorithm differs from FedAvg algorithm from the following two aspects: client sampling and function $G(\cdot)$.

1) Client sampling: In training round $t \in \mathcal{T}$, the central server samples S clients using stratified sampling [29], [30]. That is, the central server randomly samples clients from each group independently. An important question is how many clients should be sampled from each group. We will provide an answer to this question in Section IV-B when we analyze the performance guarantee. For now, let S^k denote the number of clients sampled from group k in each training round. We have $\sum_{k \in \mathcal{K}} S^k = S$. Let $\mathcal{S}_t^k$ denote the set of sampled clients of group k in training round t, and $\mathcal{S}_t \triangleq \cup_{k \in \mathcal{K}} \mathcal{S}_t^k$.

2) Function $G(\cdot)$: In FedSS algorithm, the central server computes $\boldsymbol{\omega}_t$ as follows:

$$G(\{\boldsymbol{\omega}_t^n, n \in \mathcal{S}_t\}) = \sum_{k \in \mathcal{K}} \frac{\sum_{n \in \mathcal{N}_k} D_n}{\sum_{n' \in \mathcal{N}} D_{n'}} \left(\sum_{n \in \mathcal{S}_t^k} \frac{D_n \boldsymbol{\omega}_t^n}{\sum_{n' \in \mathcal{S}_t^k} D_{n'}} \right). \tag{9}$$

Specifically, $\sum_{n \in \mathcal{S}_t^k} D_n \boldsymbol{\omega}_t^n / (\sum_{n' \in \mathcal{S}_t^k} D_{n'})$ is the weighted average of the parameter vectors of the sampled clients of group k. The term $\sum_{n \in \mathcal{N}_k} D_n / \sum_{n' \in \mathcal{N}} D_{n'}$ is the weight assigned to group k. Hence, $G(\{\boldsymbol{\omega}_t^n, n \in \mathcal{S}_t\})$ is the weighted average of the associated average parameter vectors of all groups $k \in \mathcal{K}$. For example, suppose the number of data samples of a particular group is higher than other groups. When the central server computes $G(\{\boldsymbol{\omega}_t^n, n \in \mathcal{S}_t\})$, the weight assigned to the local models of the clients of that group will be larger.

3) Unbiasedness: The proposed algorithm can address the system induced bias by making the global model an unbiased estimation of the local models of all clients. Such unbiasedness does not depend on the choice of S^k for $k \in \mathcal{K}$.

Lemma 2 (Unbiasedness)**.** *When S^k is positive for all $k \in \mathcal{K}$, FedSS algorithm is unbiased, i.e., $\mathbb{E}_{\mathcal{S}_{\tau(i)}}[\bar{\boldsymbol{v}}_i] = \bar{\boldsymbol{\omega}}_i$ for $i \in \mathcal{I}_G$.*

Proof. For any $i \in \mathcal{I}_G$, based on the definition of $\bar{\boldsymbol{v}}_i$ and $\boldsymbol{v}_i^n$ as well as function $G(\cdot)$ in (9), we have

$$\mathbb{E}_{\mathcal{S}_{\tau(i)}}[\bar{\boldsymbol{v}}_i] = \sum_{k\in\mathcal{K}} \frac{N_k}{N S^k_{\tau(i)}} \left(\sum_{n\in\mathcal{N}_k} \mathbb{P}(n \in \mathcal{S}_{\tau(i)} \mid n \in \mathcal{N}_k) \boldsymbol{\omega}_i^n \right), \tag{10}$$

where with stratified sampling, $\mathbb{P}(n \in \mathcal{S}_{\tau(i)} \mid n \in \mathcal{N}_k) = (A^k_{\tau(i)}/N_k)(S^k_{\tau(i)}/A^k_{\tau(i)})$ for any $n \in \mathcal{N}_k$. By substituting (10), we have $\mathbb{E}_{\mathcal{S}_{\tau(i)}}[\bar{\boldsymbol{v}}_i] = \sum_{k\in\mathcal{K}} \sum_{n\in\mathcal{N}_k} \boldsymbol{\omega}_i^n / N = \bar{\boldsymbol{\omega}}_i$. □

Lemma 2 confirms that the expected global model $\mathbb{E}_{\mathcal{S}_{\tau(i)}}[\bar{\boldsymbol{v}}_i]$ is always equal to the average local models of all clients $\bar{\boldsymbol{\omega}}_i$. That is, under our proposed algorithm, $\bar{\boldsymbol{v}}_i$ is an unbiased estimation of $\bar{\boldsymbol{\omega}}_i$, which is independent of S and $A_{\tau(i)}$. In the next section, we derive the mean squared error (MSE) of such an estimation (i.e., the expected value of $\|\bar{\boldsymbol{v}}_i - \bar{\boldsymbol{\omega}}_i\|^2$). A small MSE implies that the realization of the global model $\bar{\boldsymbol{v}}_i$ will not be far from the average local models $\bar{\boldsymbol{\omega}}_i$ of all clients.

IV. Convergence Analysis and Comparison

In this section, we first present the assumptions we used for analysis. Then, we derive the MSE of the global model. This is challenging because the analysis requires the computation of the variance of the clients' local models within each group with respect to system parameters. Based on the MSE result, we analyze the performance guarantee of our proposed FedSS algorithm in terms of the precision of the global model and propose an optimal client allocation scheme. Finally, we derive the performance improvement of our proposed algorithm.

A. Assumptions of Loss Function and Datasets

We make the following assumptions on the loss function and the datasets of clients. Assumptions 1−4 are commonly considered in the existing works (e.g., [24], [28]). Assumption 5 is related to the datasets of the K groups of clients. Let Ω denote the space of $\boldsymbol{\omega}$. Let $\mathcal{I}^+ \triangleq \mathcal{I} \cup \{0\}$.

Assumption 1. *Loss function $f_n(\cdot)$ is β-smooth for $n \in \mathcal{N}$, i.e., $|f_n(\boldsymbol{y}) - f_n(\boldsymbol{x}) - \nabla f_n(\boldsymbol{x})^T(\boldsymbol{y}-\boldsymbol{x})| \le \frac{\beta}{2}\|\boldsymbol{y}-\boldsymbol{x}\|^2$, $\boldsymbol{x}, \boldsymbol{y} \in \Omega$.*

Assumption 2. *Loss function $f_n(\cdot)$ is μ-strongly convex for $n \in \mathcal{N}$, i.e., $f_n(\boldsymbol{y}) \ge f_n(\boldsymbol{x}) + \nabla f_n(\boldsymbol{x})^T(\boldsymbol{y}-\boldsymbol{x}) + \frac{\mu}{2}\|\boldsymbol{y}-\boldsymbol{x}\|^2$, $\boldsymbol{x}, \boldsymbol{y} \in \Omega$.*

Assumption 3. *The variance of the stochastic gradients is bounded for $n \in \mathcal{N}$, i.e., $\mathbb{E}_{\xi_i^n}[\|\nabla f_n(\boldsymbol{\omega}_i^n; \xi_i^n) - \nabla f_n(\boldsymbol{\omega}_i^n)\|^2] \le \sigma_n^2$, $i \in \mathcal{I}$.*

Assumption 4. *The expected value of the squared norm of the stochastic gradients is bounded for $n \in \mathcal{N}$, i.e., $\mathbb{E}_{\xi_i^n}[\|\nabla f_n(\boldsymbol{\omega}_i^n; \xi_i^n)\|^2] \le R^2$, $i \in \mathcal{I}^+$.*

Assumption 5. *The expected value of the variance of the stochastic gradients from the clients of any group $k \in \mathcal{K}$ is bounded. That is, for group $k \in \mathcal{K}$,*

$$\mathbb{E}_{\boldsymbol{\xi}_i^k}\left[\sum_{n\in\mathcal{N}_k} \frac{\left\| \boldsymbol{h}_i^n(\xi_i^n) - \bar{\boldsymbol{h}}_i^k(\boldsymbol{\xi}_i^k) \right\|^2}{N_k - 1} \right] \le H_k^2, \ i \in \mathcal{I}^+, \tag{11}$$

where $\boldsymbol{h}_i^n(\xi_i^n) \triangleq \nabla f_n(\boldsymbol{\omega}_i^n; \xi_i^n)$, $\boldsymbol{\xi}_i^k \triangleq (\xi_i^n, \ n \in \mathcal{N}_k)$, and $\bar{\boldsymbol{h}}_i^k(\boldsymbol{\xi}_i^k) \triangleq \sum_{n\in\mathcal{N}_k} \boldsymbol{h}_i^n(\xi_i^n)/N_k$.

Although the data samples of the clients from group k are generated using the same distribution $P_k(\boldsymbol{x}, \boldsymbol{y})$, in Assumption 5, the value of H_k^2 is non-zero for two reasons. First, the empirical distributions of the data samples of different clients may be different. Second, various data samples can be generated by distribution $P_k(\boldsymbol{x}, \boldsymbol{y})$, under which the randomly sampled mini-batches ξ_i^n of different clients may be different. Thus, H_k^2 can reflect the dissimilarity of the empirical distributions of the clients from group $k \in \mathcal{K}$ and that of the data samples generated by distribution $P_k(\boldsymbol{x}, \boldsymbol{y})$. We refer to H_k^2 as the degree of dissimilarity of the clients' datasets in group k.

B. MSE of the Unbiased Estimation

Let $\boldsymbol{\xi}_i^{\mathrm{H}} = (\xi_{i'}^n, \ i' \le i, n \in \mathcal{N})$ denote the realization of the mini-batch sampling until the i^{th} SGD step. Let $\boldsymbol{\mathcal{S}}_i^{\mathrm{H}} = (\mathcal{S}_{t'}, t' \le \tau(i))$ denote the collection of sampled client sets until training round $\tau(i)$. We define $\boldsymbol{\mathcal{H}}_i \triangleq (\boldsymbol{\xi}_i^{\mathrm{H}}, \boldsymbol{\mathcal{S}}_i^{\mathrm{H}})$. The value of $\bar{\boldsymbol{\omega}}_i$ depends on $\boldsymbol{\mathcal{H}}_{i-1}$, and the value of $\bar{\boldsymbol{v}}_i$ depends on $\boldsymbol{\mathcal{H}}_{i-1}$ and the sampled client set $\mathcal{S}_{\tau(i)}$ for $i \in \mathcal{I}_G$.

We now characterize the MSE of the global model after global synchronization with respect to client and mini-batch sampling. Such an MSE reveals the expected quadratic gap between the realization of the global model with client sampling $\bar{\boldsymbol{v}}_i$ and the average local models of all clients $\bar{\boldsymbol{\omega}}_i$.

Lemma 3 (MSE)**.** *Let η_i be non-increasing with respect to $i \in \mathcal{I}^+$ and satisfies $\eta_i \le 2\eta_{i+EM}$.*[4] *Under Assumption 5,*

$$\begin{aligned} &\mathbb{E}_{\boldsymbol{\mathcal{H}}_{i-1}, \mathcal{S}_{\tau(i)}}\left[\left\| \bar{\boldsymbol{v}}_i - \bar{\boldsymbol{\omega}}_i \right\|^2 \right] \\ &\le 4E^2M^2\eta_i^2 \sum_{k\in\mathcal{K}} \left(\frac{N_k}{N} \right)^2 \left(\frac{1}{S^k} - \frac{1}{N_k} \right) H_k^2, \ i \in \mathcal{I}. \end{aligned} \tag{12}$$

Proof. For $i \in \mathcal{I} \setminus \mathcal{I}_G$, we have $\bar{\boldsymbol{v}}_i = \bar{\boldsymbol{\omega}}_i$ based on (6). Thus, $\mathbb{E}_{\boldsymbol{\mathcal{H}}_{i-1}, \mathcal{S}_{\tau(i)}}[\|\bar{\boldsymbol{v}}_i - \bar{\boldsymbol{\omega}}_i\|^2]$ is equal to zero and hence is not larger than any nonnegative constant, i.e., inequality (12) holds. We now focus on the case for $i \in \mathcal{I}_G$. Based on (6) and Lemma 2, given any $\boldsymbol{\mathcal{H}}_{i-1}$, $\bar{\boldsymbol{v}}_i$ is the estimator of the population mean $\bar{\boldsymbol{\omega}}_i$ under stratified sampling. Based on [29, eqn. (5.2)],

$$\begin{aligned} &\mathbb{E}_{\boldsymbol{\mathcal{H}}_{i-1}, \mathcal{S}_{\tau(i)}}\left[\left\| \bar{\boldsymbol{v}}_i - \bar{\boldsymbol{\omega}}_i \right\|^2 \right] \\ &= \sum_{k\in\mathcal{K}} \left(\frac{N_k}{N} \right)^2 \left(\frac{1}{S^k} - \frac{1}{N_k} \right) \mathbb{E}_{\boldsymbol{\mathcal{H}}_{i-1}} \left[\frac{\sum_{n\in\mathcal{N}_k} \|\boldsymbol{\omega}_i^n - \bar{\boldsymbol{\omega}}_i^k\|^2}{N_k - 1} \right], \end{aligned} \tag{13}$$

[4] For example, $\eta_i = 2/(a(i+b))$ for $i \in \mathcal{I}^+$, where $a > 0$ and $b > EM$.

where $\bar{\boldsymbol{\omega}}_i^k \triangleq \sum_{n\in\mathcal{N}_k} \boldsymbol{\omega}_i^n/N_k$ for $k \in \mathcal{K}$. We first bound $\|\boldsymbol{\omega}_i^n - \bar{\boldsymbol{\omega}}_i^k\|$ for $i \in \mathcal{I}_G$. Let i_0 be the largest index that corresponds to a global synchronization process before index i, so $i_0 = i - EM$. We always have $\boldsymbol{v}_{i_0}^n = \boldsymbol{v}_{i_0}^{n'}$ for all $n, n' \in \mathcal{N}$. According to the definition of $\boldsymbol{\omega}_i^n$ and $\bar{\boldsymbol{\omega}}_i^k$,

$$\begin{aligned}
&\|\boldsymbol{\omega}_i^n - \bar{\boldsymbol{\omega}}_i^k\| \\
=\ & \Big\|\boldsymbol{v}_{i_0}^n - \textstyle\sum_{i'=i_0}^{i-1} \eta_{i'} \nabla f_n(\boldsymbol{v}_{i'}^n; \xi_{i'}^n) \\
& - \tfrac{1}{N_k} \textstyle\sum_{n'\in\mathcal{N}_k} \left(\boldsymbol{v}_{i_0}^{n'} - \sum_{i'=i_0}^{i-1} \eta_{i'} \nabla f_{n'}(\boldsymbol{v}_{i'}^{n'}; \xi_{i'}^{n'})\right) \Big\| \\
=\ & \Big\| \textstyle\sum_{i'=i_0}^{i-1} \eta_{i'} \big(\sum_{n'\in\mathcal{N}_k} \nabla f_{n'}(\boldsymbol{v}_{i'}^{n'}; \xi_{i'}^{n'})/N_k - \nabla f_n(\boldsymbol{v}_{i'}^n; \xi_{i'}^n)\big) \Big\| \\
\le\ & \textstyle\sum_{i'=i_0}^{i-1} \eta_{i'} \big\| \sum_{n'\in\mathcal{N}_k} \nabla f_{n'}(\boldsymbol{v}_{i'}^{n'}; \xi_{i'}^{n'})/N_k - \nabla f_n(\boldsymbol{v}_{i'}^n; \xi_{i'}^n)\big\|.
\end{aligned} \tag{14}$$

We now bound $\mathbb{E}_{\boldsymbol{\mathcal{H}}_{i-1}}[\sum_{n\in\mathcal{N}_k} \|\boldsymbol{\omega}_i^n - \bar{\boldsymbol{\omega}}_i^k\|^2/(N_k-1)]$. Recall that $\boldsymbol{h}_i^n(\xi_i^n) \triangleq \nabla f_n(\boldsymbol{\omega}_i^n; \xi_i^n)$. For any $i \notin \mathcal{I}_G$, $\boldsymbol{v}_i^n = \boldsymbol{\omega}_i^n$, we have $\boldsymbol{h}_i^n(\xi_i^n) = \nabla f_n(\boldsymbol{v}_i^n; \xi_i^n)$. Based on Assumption 5,

$$\begin{aligned}
& \mathbb{E}_{\boldsymbol{\mathcal{H}}_{i-1}} \left[\textstyle\sum_{n\in\mathcal{N}_k} \|\boldsymbol{\omega}_i^n - \bar{\boldsymbol{\omega}}_i^k\|^2/(N_k-1)\right] \\
\le\ & \mathbb{E}_{\boldsymbol{\mathcal{H}}_{i-1}} \left[\frac{\sum_{n\in\mathcal{N}_k} \left(\sum_{i'=i_0}^{i-1} \eta_{i'} \|\boldsymbol{h}_{i'}^n(\xi_{i'}^n) - \bar{\boldsymbol{h}}_{i'}^k(\boldsymbol{\xi}_{i'}^k)\|\right)^2}{N_k - 1}\right] \\
\overset{(a)}{\le}\ & \mathbb{E}_{\boldsymbol{\mathcal{H}}_{i-1}} \left[(EM\eta_{i_0})^2 \frac{\sum_{n\in\mathcal{N}_k} \|\boldsymbol{h}_i^n(\xi_i^n) - \bar{\boldsymbol{h}}_i^k(\boldsymbol{\xi}_i^k)\|^2}{N_k - 1}\right] \\
\le\ & 4E^2M^2\eta_i^2 H_k^2,
\end{aligned} \tag{15}$$

where (a) is due to the convexity of $(\sum_{i'=i_0}^{i-1} \eta_{i'} \|\boldsymbol{h}_{i'}^n(\xi_{i'}^n) - \bar{\boldsymbol{h}}_{i'}^k(\boldsymbol{\xi}_{i'}^k)\|)^2$ and the non-increasing η_i that satisfies $\eta_i \le 2\eta_{i+EM}$ for $i \in \mathcal{I}^+$. According to (13) and (15), we obtain (12) for $i \in \mathcal{I}_G$. □

Recall that a smaller MSE implies that the realization of the global model $\bar{\boldsymbol{v}}_i$ is closer to the average local models $\bar{\boldsymbol{\omega}}_i$ of all clients, which intuitively leads to a better algorithm performance. Lemma 3 implies that under our proposed FedSS algorithm, the MSE does not depend on the variation of the number of available clients. This is ideal because this implies that the algorithm performance will not be affected by the time-varying client availability. In addition, when H_k^2 is small (e.g., the empirical distributions of the clients from group k is more similar), the MSE of the global model becomes small as well. When N_k increases, the value of H_k^2 has a stronger impact on the MSE.

C. Precision and Client Allocation

Let $\boldsymbol{\omega}_T^{\text{SS}}$ denote the global model obtained using our proposed FedSS algorithm. We now present the bound of the precision, i.e., $\mathbb{E}_{\boldsymbol{\omega}_T}[F(\boldsymbol{\omega}_T^{\text{SS}})] - F(\boldsymbol{\omega}^\star)$, under FedSS algorithm.

Theorem 1 (Precision)**.** *Let* $\alpha_1 = \max\{8\beta/\mu - 1, EM\}$ *and* $\eta_i = 2/(\mu(i+\alpha_1))$*. Under Assumptions 1–5,*

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{\omega}_T}[F(\boldsymbol{\omega}_T^{\text{SS}})] - F(\boldsymbol{\omega}^\star) \\
&\le B^{\text{SS}} \triangleq \frac{2\beta}{\alpha_1 + TEM} \left(\frac{C_1 + C_2^{\text{SS}}}{\mu^2} + \frac{2\beta\|\boldsymbol{\omega}_0 - \boldsymbol{\omega}^\star\|^2}{\mu}\right),
\end{aligned} \tag{16}$$

where

$$C_1 \triangleq \textstyle\sum_{n\in\mathcal{N}} \sigma_n^2/N^2 + 6\beta\Gamma + 8\beta(EM-1)^2R^2, \tag{17}$$

$$C_2^{\text{SS}} \triangleq 4E^2M^2 \textstyle\sum_{k\in\mathcal{K}} (N_k/N)^2 \left(1/S^k - 1/N_k\right) H_k^2, \tag{18}$$

and σ_n^2 *is the bound of the variance given in Assumption 3. We define* $\Gamma \triangleq F(\boldsymbol{\omega}^\star) - \sum_{n\in\mathcal{N}} \mathbb{E}_{\xi_n}[f_n(\boldsymbol{\omega}_n^\star; \xi_n)]/N$*, where* $\boldsymbol{\omega}_n^\star = \arg\min_{\boldsymbol{\omega}} \mathbb{E}_{\xi_n}[f_n(\boldsymbol{\omega}; \xi_n)]$.

The proof can be found in Appendix. Based on Theorem 1, we propose an *optimal client allocation* scheme, which determines the number of clients sampled from each group such that the bound of the precision, i.e., B^{SS}, is minimized.

Proposition 1 (Optimal Client Allocation)**.** *The client allocation that minimizes the bound of the precision* B^{SS} *is*

$$S^k = H_k N_k S/(\textstyle\sum_{k'\in\mathcal{K}} H_{k'} N_{k'}),\ k \in \mathcal{K}, \tag{19}$$

where H_k *is the squared root of* H_k^2 *in* (11)*.*

Proposition 1 is proven by showing that minimizing the bound of the precision is equivalent to minimizing the bound of the MSE $\mathbb{E}_{\boldsymbol{\mathcal{H}}_{i-1}, \mathcal{S}_{\tau(i)}}[\|\bar{\boldsymbol{v}}_i - \bar{\boldsymbol{\omega}}_i\|^2]$ in (3). Then, the optimal S^k for $k \in \mathcal{K}$ is achieved by solving the following problem:

$$\begin{aligned}
\underset{S^k, k\in\mathcal{K}}{\text{minimize}} \quad & \textstyle\sum_{k\in\mathcal{K}} (N_k/N)^2 \left(1/S^k - 1/N_k\right) H_k^2 \\
\text{subject to} \quad & \textstyle\sum_{k\in\mathcal{K}} S^k = S, \\
& S^k > 0,\ k \in \mathcal{K}.
\end{aligned} \tag{20}$$

Problem (20) is a convex programming problem, as its objective function is the sum of multiple convex functions and its constraints are linear. Moreover, constraint $S^k > 0$ can be written as $S^k \ge 0$ for $k \in \mathcal{K}$. This is because for any S_k approaches zero, the objective function approaches infinity, which cannot be the optimal value to problem (20). By checking the Karush–Kuhn–Tucker (KKT) conditions of problem (20), we obtain the optimal S^k in (19). Proposition 1 implies that if the degree of dissimilarity of the data H_k^2 or the number of clients N_k in group k is higher than that of other groups, then more clients should be sampled from group k. The central server can estimate H_k^2 across training rounds. However, the exact value of H_k^2 can be obtained only after the entire FL process is completed. Alternatively, the central server can use *proportional client allocation* for implementation simplicity. That is, $S^k = SN_k/N$ for $k \in \mathcal{K}$. When H_k^2 is identical for all $k \in \mathcal{K}$, proportional client allocation minimizes the bound of the precision B^{SS}.

D. Comparison between FedSS and FedAvg Algorithms

We first introduce some notations. Based on Assumption 5, the expected value of the variance of the stochastic gradients from all clients is bounded. Thus, there exists an H^2 that satisfies the following inequality:

$$\mathbb{E}_{\boldsymbol{\xi}_i}\left[\sum_{n\in\mathcal{N}} \frac{\left\|\boldsymbol{h}_i^n(\xi_i^n) - \bar{\boldsymbol{h}}_i(\boldsymbol{\xi}_i)\right\|^2}{N-1}\right] \le H^2,\ i \in \mathcal{I}^+, \tag{21}$$

where $\boldsymbol{\xi}_i \triangleq (\boldsymbol{\xi}_i^k, k \in \mathcal{K})$, and $\bar{\boldsymbol{h}}_i(\boldsymbol{\xi}_i) \triangleq \sum_{n\in\mathcal{N}} \boldsymbol{h}_i^n(\xi_i^n)/N$. Different from H_k^2 defined in (11), the reason for having a non-zero H^2 is that the data samples of the clients from different groups are generated using different distributions, i.e., $P_k(\boldsymbol{x}, \boldsymbol{y})$ for $k \in \mathcal{K}$. Thus, the value of H^2 can be much larger than those of H_k^2 for $k \in \mathcal{K}$, and it reflects the non-IID degree of the datasets of all clients.

It is challenging to determine the performance of FedAvg algorithm under arbitrary client availability due to its biased nature. Thus, we focus on the case where $A_t^k/A_t = N_k/N$ holds for $k \in \mathcal{K}$ and $t \in \mathcal{T}$. In this case, the FedAvg algorithm is unbiased as well, while it may achieve a higher MSE than our proposed algorithm.[5] Let B^{AVG} denote the upper bound of the precision of the global model under FedAvg algorithm, i.e., $\mathbb{E}_{\boldsymbol{\omega}_T}[F(\boldsymbol{\omega}_T^{\text{AVG}})] - F(\boldsymbol{\omega}^\star) \leq B^{\text{AVG}}$, where $\boldsymbol{\omega}_T^{\text{AVG}}$ denotes the global model obtained using FedAvg algorithm. Let $B^{\text{SS-O}}$ and $B^{\text{SS-P}}$ denote the corresponding bound of FedSS algorithm under optimal and proportional client allocation, respectively.

Corollary 1 (Precision Difference)**.** *The difference between B^{AVG} and $B^{\text{SS-P}}$, i.e., $B^{\text{AVG}} - B^{\text{SS-P}}$, is given by*

$$\frac{8\beta E^2 M^2}{\mu^2(\alpha_1 + TEM)} \left(\frac{1}{S} - \frac{1}{N}\right) \left(H^2 - \sum_{k\in\mathcal{K}} \frac{N_k H_k^2}{N}\right). \quad (22)$$

In addition, $B^{\text{AVG}} - B^{\text{SS-O}}$ is not smaller than $B^{\text{AVG}} - B^{\text{SS-P}}$.

This is proven by first deriving the bound B^{AVG}, where the proof is simiar as that of Theorem 1, and then computing the difference between B^{AVG} and $B^{\text{SS-P}}$ or between B^{AVG} and $B^{\text{SS-O}}$. Note that a larger gap $B^{\text{AVG}} - B^{\text{SS-P}}$ or $B^{\text{AVG}} - B^{\text{SS-O}}$ implies a more significant improvement of our proposed algorithm. Corollary 1 shows that when compared with FedAvg algorithm, the improvement of our proposed algorithm linearly increases with the non-IID degree of the clients' datasets (i.e., H^2) and linearly decreases with the degree of dissimilarity of the clients' datasets in any arbitrary group (i.e., H_k^2).

V. Performance Evaluation

In the experiments, clients collect data from MNIST [26] or CIFAR-10 [27] datasets. Both datasets have been used in many existing works on FL, e.g., [1], [7]. To evaluate how the non-IID degree affects the performance, we consider the following setting [24]. The dataset of a client consists of two parts, i.e., an IID part and a non-IID part. A client first selects data samples for the IID part by randomly selecting data from the dataset. Then, we sort the remaining data based on their labels. The sorted dataset is divided into equal size blocks. Finally, each client selects data samples for the non-IID part by randomly selecting blocks from the sorted dataset. We define the *non-IID ratio* as the ratio of the number of data samples in the non-IID part to the total number of data samples. A higher ratio implies that the distributions for generating the data samples of clients from different groups (i.e., $P_k(\boldsymbol{x}, \boldsymbol{y})$, $k \in \mathcal{K}$) are more different, and hence it implies a higher non-IID degree of clients' data. We follow [1] and set $N = 100$ and $S = 10$. For the time-varying client availability, we use the dataset in [15], which contains the availability information of streamers on a live streaming platform. We have relaxed the assumptions and approximation considered in Section IV.

A. Client Classification

In the experiments, we relax the setting where the central server knows the group that each client belongs to. We focus on an important scenario with *label distribution skew* [2, Section 3.1]. That is, the clients of different groups $k \in \mathcal{K}$ have different label distributions $P_k(\boldsymbol{y})$, while the conditional distribution $P(\boldsymbol{x} \mid \boldsymbol{y})$ is identical for all groups.[6] Our proposed approach is also applicable to feature distribution skew [2, Section 3.1] by classifying clients based on feature distributions.

Let $\mathcal{Y} \triangleq \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_L\}$ denote the set of discrete labels, where L is the total number of labels. Let $\boldsymbol{p}_n \triangleq (D_n^l/D_n, l \in \mathcal{L} \triangleq \{1, 2, \ldots, L\})$ denote the empirical label distribution of client $n \in \mathcal{N}$.[7] Here, D_n^l is the number of client n's data samples with label $\boldsymbol{y}_l$. The main idea for the central server is to classify clients into groups by clustering their vectors $\boldsymbol{p}_n$ for $n \in \mathcal{N}$ into multiple groups using clustering algorithms (e.g., the expectation-maximization (EM) algorithm for Gaussian mixture model [31]). Since most clustering algorithms require a pre-determined number of groups, we let the central server exhaustively try different number of groups K°. For each K°, the central server determines the group classification result $\mathrm{C}(K^\circ)$ and computes $\mathrm{score}(K^\circ)$ using Silhouette score [32]:

$$\mathrm{score}(K^\circ) = \sum_{n\in\mathcal{N}} \frac{s_n^{\text{out}}(\mathrm{C}(K^\circ)) - s_n^{\text{in}}(\mathrm{C}(K^\circ))}{N\max\{s_n^{\text{in}}(\mathrm{C}(K^\circ)), s_n^{\text{out}}(\mathrm{C}(K^\circ))\}}. \quad (23)$$

Here, $s_n^{\text{out}}(\mathrm{C}(K^\circ)) = \min_{k\neq k_n} \sum_{n'\in\mathcal{N}_k} \|\boldsymbol{p}_n - \boldsymbol{p}_{n'}\|/N_k$, where k_n denotes the group that client n belongs to. The value of $s_n^{\text{in}}(\mathrm{C}(K^\circ)) = \sum_{n'\in\mathcal{N}_{k_n}\setminus\{n\}} \|\boldsymbol{p}_n - \boldsymbol{p}_{n'}\|/(N_{k_n} - 1)$ for $\mathcal{N}_{k_n} \setminus \{n\} \neq \emptyset$, and $s_n^{\text{in}}(\mathrm{C}(K^\circ)) = s_n^{\text{out}}(\mathrm{C}(K^\circ))$ otherwise. A larger $\mathrm{score}(K^\circ)$ implies that the label distributions of the clients within a group is more similar, and those from different groups are more diverse. Finally, the central server obtains the best group classification result by finding the maximum $\mathrm{score}(K^\circ)$.

It is challenging to analyze the performance of such classification approach due to the complicated relationship between the clients' datasets. We now focus on a special case.

Assumption 6 (Identical Label Case)**.** *We assume that vector $\boldsymbol{p}_n \in \{\boldsymbol{e}_1, \boldsymbol{e}_2 \ldots, \boldsymbol{e}_L\}$ for any $n \in \mathcal{N}$, where $\boldsymbol{e}_l$ is of size L*

[5] Intuitively, under such a scenario, the derived precision reduction of our proposed algorithm (when compared with that of FedAvg algorithm) provides a lower bound of the actual precision reduction. That is, when we relax such a scenario setting, our proposed FedSS algorithm can improve the performance more significantly than the derived result. We will show the results under the relaxed setting in Section V.

[6] Consider image recognition as an example. The label distribution skew corresponds to the case where clients of different groups have images of different types of objects. If some clients have images containing the same type of object, then these images may look similar regardless of the clients.

[7] Although submitting the label distribution may reveal the statistics of clients' data, its impact on the clients' privacy could be minimal. Take image recognition for bus and plane classification as an example. Clients need to submit only the number of bus and plane images that they have, and they do not need to submit the corresponding images.

TABLE I
IMPACT OF NON-IID RATIO.

Non-IID ratio	80%		90%		95%	
	Rounds	Speedup	Rounds	Speedup	Rounds	Speedup
(CIFAR-10 dataset)						
FedAvg	35	(1×)	41	(1×)	199	(1×)
FedSS-O	14	(2.50×)	23	(1.78×)	39	(5.10×)
FedSS-P	14	(2.50×)	22	(1.86×)	47	(4.23×)
FedProx	34	(1.03×)	67	(0.61×)	114	(1.75×)
FedProxSS-O	15	(2.33×)	23	(1.78×)	43	(4.63×)
FedProxSS-P	15	(2.33×)	20	(2.05×)	46	(4.33×)
CS-UCB-Q	14	(2.50×)	28	(1.46×)	159	(1.25×)
(MNIST dataset)						
FedAvg	39	(1×)	51	(1×)	89	(1×)
FedSS-O	32	(1.22×)	40	(1.28×)	44	(2.02×)
FedSS-P	30	(1.30×)	34	(1.50×)	44	(2.02×)
FedProx	37	(1.05×)	65	(0.78×)	58	(1.53×)
FedProxSS-O	34	(1.15×)	40	(1.28×)	51	(1.75×)
FedProxSS-P	36	(1.08×)	36	(1.42×)	52	(1.71×)
CS-UCB-Q	36	(1.08×)	42	(1.21×)	61	(1.50×)

TABLE II
IMPACT OF S (WITH A NON-IID RATIO OF 95%).

S	10		20		30	
	Rounds	Speedup	Rounds	Speedup	Rounds	Speedup
FedAvg	199	(1×)	34	(1×)	32	(1×)
FedSS-O	39	(5.10×)	25	(1.36×)	24	(1.33×)
FedSS-P	47	(4.23×)	24	(1.42×)	23	(1.39×)
FedProx	114	(1.75×)	28	(1.21×)	37	(0.86×)
FedProxSS-O	43	(4.63×)	25	(1.36×)	22	(1.45×)
FedProxSS-P	46	(4.33×)	25	(1.36×)	24	(1.33×)

and has 1 *as the* l^{th} *element and zeros elsewhere. That is, all data of a client corresponds to an identical label. Meanwhile, the set of clients whose data corresponds to label* $l \in \mathcal{L}$*, i.e.,* $\mathcal{N}_l^{label} \triangleq \{n \mid \boldsymbol{p}_n = \boldsymbol{e}_l, n \in \mathcal{N}\}$*, is non-empty.*

Proposition 2 (Performance Guarantee)**.** *Given* $i \in \mathcal{I}$ *and* $k \in \mathcal{K}$*, let* conv_i^k *denote the convex hull of* $\boldsymbol{h}_i^n(\xi_i^n)$ *for* $n \in \mathcal{N}_k$*. There exists a threshold* χ *such that given any* $i \in \mathcal{I}$*, if* $\|\boldsymbol{z}_i^k - \boldsymbol{z}_i^{k'}\| > \chi$ *for any* $\boldsymbol{z}_i^k \in \mathrm{conv}_i^k$ *and* $\boldsymbol{z}_i^{k'} \in \mathrm{conv}_i^{k'}$*, then the proposed client classification approach ensures* $H^2 - \sum_{k\in\mathcal{K}} N_k H_k^2/N > 0$.

This proposition is proven with two steps. First, under Assumption 6, we can prove that $K^\circ = L$ leads to $\mathrm{C}(L) = (\mathcal{N}_l^{\text{label}}, l \in \mathcal{L})$ based on the clustering algorithm [31]. Thus, $\mathrm{score}(L) = 1$. By showing that there does not exist any $K^\circ \neq L$ that ensures $\mathrm{score}(K^\circ) = 1$, the proposed approach sets $K = L$ as the number of groups and classifies clients using $\mathrm{C}(L)$.

Second, based on the separating hyperplane theorem [33, Section 2.5] and the definition of H_k^2 for $k \in \mathcal{K}$ and H^2, if the convex hull of the stochastic gradients over the datasets corresponding to different labels are separated (i.e., $\|\boldsymbol{z}_i^k - \boldsymbol{z}_i^{k'}\| > \chi$), then $H^2 - \sum_{k\in\mathcal{K}} N_k H_k^2/N > 0$.

Proposition 2 shows that under Assumption 6, the proposed client classification approach ensures $B^{\text{SS-O}} \leq B^{\text{SS-P}} < B^{\text{AVG}}$.

B. Experimental Results

Table I shows the number of training rounds required to achieve a loss of 1.5 and 0.1 with CIFAR-10 and MNIST datasets, respectively. We choose a small threshold for MNIST dataset, as a small number of training rounds is sufficient for achieving a small loss. The blue horizontal bars graphically show the corresponding number of training rounds, where the FedAvg algorithm [1] serves as the baseline. Each column with "Speedup" shows the number of training rounds of the baseline divided by that of the corresponding algorithm. To verify that our proposed approach can be extended to improve the performance of other synchronous FL algorithms, we incorporate stratified sampling into FedProx [23], a variant of FedAvg algorithm. We call this extended algorithm as FedProxSS and set $\mu = 0.01$ (see [23]). For both FedSS and FedProxSS, we use suffix "-O" and "-P" to refer the optimal and proportional client allocation schemes, respectively. In practical systems, we need to estimate the values of H_k^2 for $k \in \mathcal{K}$ across training rounds and cannot know their ground-truth values before the FL process is completed. Thus, the algorithms with optimal allocation scheme may not necessarily achieve a better empirical performance than those with proportional scheme.

In Table I, we compare our proposed algorithms with FedAvg [1], FedProx [23], and CS-UCB-Q [18] algorithms. A smaller number of training rounds implies a higher convergence rate. We have the following observations. First, for both CIFAR-10 and MNIST datasets, our FedSS and FedProxSS algorithms always achieve a higher convergence rate than the FedAvg algorithm. This improvement is significant when the non-IID ratio is high. Under a ratio of 95%, the improvement can be up to 5.1 times with CIFAR-10 dataset. Second, the convergence rate improvement of our FedSS and FedProxSS algorithms under CIFAR-10 is more significant than that under MNIST. This implies that our proposed algorithms are more beneficial when the training requires a larger number of rounds to reach a certain level of performance. Third, when compared with CS-UCB-Q, our proposed FedSS and FedProxSS algorithms can improve the convergence rate by up to 4.08 times when the non-IID ratio is high.

We now present the test accuracy results (i.e., the ratio of correct image recognition over test dataset) with CIFAR-10 dataset. Results in Figs. 2(a) and (b) show that our FedSS algorithm improves the test accuracy when compared with the FedAvg algorithm. When the non-IID ratio is high (i.e., 95%), at training round $t = 30$, the accuracy improvement can be up to 53.4%. Similarly, in Figs. 2(c) and (d), our proposed FedProxSS achieves a higher test accuracy than FedProx. Moreover, the fluctuation of the test accuracy under our proposed FedSS and FedProxSS algorithms are less significant than that under FedAvg and FedProx algorithms. This validates that our proposed algorithms can mitigate the system induced bias and hence the unexpected variation of the global model updates across training rounds.

From Table II, we can observe that when the number of sampled clients is small, our proposed algorithms improve the convergence rate more significantly. This is consistent with Corollary 1. When compared with FedAvg and FedProx

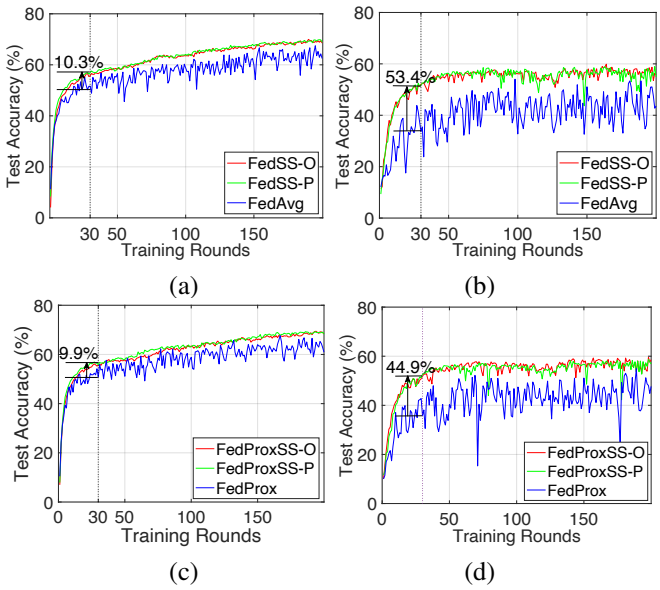

Fig. 2. Test accuracy: FedAvg and FedSS with a non-IID ratio of (a) 80% and (b) 95%; FedProx and FedProxSS with (c) 80% and (d) 95%.

algorithms under $S = 30$, our proposed algorithms under $S = 10$ can achieve a comparable convergence rate. Thus, our proposed algorithms can reduce the number of sampled clients while maintaining a satisfactory performance.

VI. Conclusion

In this work, we characterized the system induced bias of FedAvg algorithm under time-varying client availability. We found that such system induced bias always exists even if the number of available clients and sampled clients is increased. To address the system induced bias, we proposed a FedSS algorithm by incorporating stratified sampling. We proved that the proposed FedSS algorithm is unbiased and derived the theoretical performance guarantee. Experimental results show that when compared with FedAvg, FedProx, and CS-UCB-Q algorithms, our proposed FedSS and FedProxSS algorithms can improve the convergence rate by up to 5.1 times.

To extend this work, it may be interesting to consider clients of the same group having diverse probabilities of being available and quantify the system induced bias. Moreover, it may be worthwhile to design an algorithm for detecting the groups of clients in an online fashion without requesting the statistics of clients' datasets.

Appendix

We first present lemmas that are related to the SGD steps at the clients. Then, we prove Theorem 1 using these lemmas.

1) Lemmas: Let $\boldsymbol{g}_i(\boldsymbol{\xi}_i) = \sum_{n\in\mathcal{N}} \nabla f_n(\boldsymbol{v}_i^n;\xi_i^n)/N$, where $\boldsymbol{\xi}_i \triangleq (\xi_i^n, n\in\mathcal{N})$. Recall that $f_n(\boldsymbol{v}_i^n) \triangleq \mathbb{E}_{\xi_i^n}[f_n(\boldsymbol{v}_i^n;\xi_i^n)]$. We denote $\bar{\boldsymbol{g}}_i = \sum_{n\in\mathcal{N}} \nabla f_n(\boldsymbol{v}_i^n)/N$. Note that $\mathbb{E}_{\boldsymbol{\xi}_i}[\boldsymbol{g}_i(\boldsymbol{\xi}_i)] = \bar{\boldsymbol{g}}_i$ for $i\in\mathcal{I}$ and $\bar{\boldsymbol{\omega}}_{i+1} = \bar{\boldsymbol{v}}_i - \eta_i \boldsymbol{g}_i(\boldsymbol{\xi}_i)$ based on (5) and (6). Under Assumptions 1–4, we have the following lemmas.

Lemma 4 (Bound of SGD Step)**.** *Under Assumptions 1 and 2, if $\eta_i \le 1/4\beta$, then for any $i\in\mathcal{I}$, $\mathbb{E}_{\boldsymbol{\mathcal{H}}_i}\left[\|\bar{\boldsymbol{\omega}}_{i+1}-\boldsymbol{\omega}^\star\|^2\right] \le (1-\eta_i\mu)\mathbb{E}_{\boldsymbol{\mathcal{H}}_{i-1},\mathcal{S}_{\tau(i)}}\left[\|\bar{\boldsymbol{v}}_i-\boldsymbol{\omega}^\star\|^2\right] + \eta_i^2\mathbb{E}_{\boldsymbol{\mathcal{H}}_i}\left[\|\boldsymbol{g}_i(\boldsymbol{\xi}_i)-\bar{\boldsymbol{g}}_i\|\right] + 6\beta\eta_i^2\Gamma + 2\mathbb{E}_{\boldsymbol{\mathcal{H}}_{i-1},\mathcal{S}_{\tau(i)}}\left[\sum_{n\in\mathcal{N}}\|\bar{\boldsymbol{v}}_i-\boldsymbol{v}_i^n\|^2/N\right]$.*

Lemma 5 (Variance of SGD Step)**.** *Under Assumption 3, for any $i\in\mathcal{I}^+$, $\mathbb{E}_{\boldsymbol{\mathcal{H}}_i}\left[\|\boldsymbol{g}_i(\boldsymbol{\xi}_i)-\bar{\boldsymbol{g}}_i\|^2\right] \le \sum_{n\in\mathcal{N}}\sigma_n^2/N^2$.*

Lemma 6 (Divergence)**.** *Under Assumption 4, if η_i is non-increasing in $i\in\mathcal{I}^+$ and satisfies $\eta_i \le 2\eta_{i+EM}$, then $\mathbb{E}_{\boldsymbol{\mathcal{H}}_{i-1},\mathcal{S}_{\tau(i)}}\left[\sum_{n\in\mathcal{N}}\|\bar{\boldsymbol{v}}_i-\boldsymbol{v}_i^n\|^2/N\right] \le 4\eta_i^2(EM-1)^2R^2, i\in\mathcal{I}$.*

The proofs are similar as those for Lemmas 1−3 in [28, Section A.2] and hence are omitted here. The major difference is that in our work, sampled clients perform SGD steps over multiples mini-batches in each local epoch.

2) Proof for Theorem 1: Now, we prove Theorem 1. We define $\Delta_{i+1} \triangleq \mathbb{E}_{\boldsymbol{\mathcal{H}}_i,\mathcal{S}_{\tau(i+1)}}[\|\bar{\boldsymbol{v}}_{i+1}-\boldsymbol{\omega}^\star\|^2]$ for $i\in\mathcal{I}^+$. Note that $\bar{\boldsymbol{v}}_0 = \boldsymbol{\omega}_0$, we set $\Delta_0 = \|\boldsymbol{\omega}_0-\boldsymbol{\omega}^\star\|^2$. In the following, we first bound the value of Δ_{i+1} for $i\in\mathcal{I}^+$. Then, we prove Theorem 1 by bounding $\mathbb{E}_{\boldsymbol{\mathcal{H}}_{TEM-1},\mathcal{S}_T}[F(\boldsymbol{\omega}_T^{\text{SS}})]-F(\boldsymbol{\omega}^\star)$, which is equivalent to $\mathbb{E}_{\boldsymbol{\omega}_T}[F(\boldsymbol{\omega}_T^{\text{SS}})]-F(\boldsymbol{\omega}^\star)$.

Bound of Δ_{i+1}: Based on the definition of Δ_{i+1}, we have

$$\Delta_{i+1} = \mathbb{E}_{\boldsymbol{\mathcal{H}}_i,\mathcal{S}_{\tau(i+1)}}\left[\|\bar{\boldsymbol{v}}_{i+1}-\bar{\boldsymbol{\omega}}_{i+1}\|^2\right] + \mathbb{E}_{\boldsymbol{\mathcal{H}}_i}\left[\|\bar{\boldsymbol{\omega}}_{i+1}-\boldsymbol{\omega}^\star\|^2\right] + \mathbb{E}_{\boldsymbol{\mathcal{H}}_i,\mathcal{S}_{\tau(i+1)}}\left[2\langle\bar{\boldsymbol{v}}_{i+1}-\bar{\boldsymbol{\omega}}_{i+1},\bar{\boldsymbol{\omega}}_{i+1}-\boldsymbol{\omega}^\star\rangle\right],\ i\in\mathcal{I}^+, \quad (24)$$

where the operator $\langle\bar{\boldsymbol{v}}_{i+1}-\bar{\boldsymbol{\omega}}_{i+1},\bar{\boldsymbol{\omega}}_{i+1}-\boldsymbol{\omega}^\star\rangle$ denotes the dot product of vectors $\bar{\boldsymbol{v}}_{i+1}-\bar{\boldsymbol{\omega}}_{i+1}$ and $\bar{\boldsymbol{\omega}}_{i+1}-\boldsymbol{\omega}^\star$.

For the right-hand side of (24), according to Lemmas 4–6, $\mathbb{E}_{\boldsymbol{\mathcal{H}}_i}\left[\|\bar{\boldsymbol{\omega}}_{i+1}-\boldsymbol{\omega}^\star\|^2\right] \le (1-\eta_i\mu)\mathbb{E}_{\boldsymbol{\mathcal{H}}_{i-1},\mathcal{S}_{\tau(i)}}\left[\|\bar{\boldsymbol{v}}_i-\boldsymbol{\omega}^\star\|^2\right] + \eta_i^2C_1$, $i\in\mathcal{I}$, where C_1 is defined in (17). Meanwhile, if $i+1\in\mathcal{I}\setminus\mathcal{I}_G$, then $\bar{\boldsymbol{v}}_{i+1} = \bar{\boldsymbol{\omega}}_{i+1}$ according to (6). If $i+1\in\mathcal{I}_G$, then $\mathbb{E}_{\boldsymbol{\mathcal{H}}_i,\mathcal{S}_{\tau(i+1)}}[2\langle\bar{\boldsymbol{v}}_{i+1}-\bar{\boldsymbol{\omega}}_{i+1},\bar{\boldsymbol{\omega}}_{i+1}-\boldsymbol{\omega}^\star\rangle]$ is equal to zero by Lemma 2. As a result, based on Lemma 3,

$$\Delta_{i+1} \le (1-\eta_i\mu)\mathbb{E}_{\boldsymbol{\mathcal{H}}_{i-1},\mathcal{S}_{\tau(i)}}\left[\|\bar{\boldsymbol{v}}_i-\boldsymbol{\omega}^\star\|^2\right] + \eta_i^2(C_1+C_2^{\text{SS}}),\ i+1\in\mathcal{I}_G, \quad (25)$$

where C_2^{SS} is defined in (18).

We now bound the value of Δ_{i+1} for $i\in\mathcal{I}^+$. Consider a diminishing step size $\eta_i = \alpha_0/(i+\alpha_1)$, where $\alpha_0 > 1/\mu$ and $\alpha_1 > 0$ ensures that $\eta_1 \le \min\{1/\mu, 1/(4\beta)\}$ and $\eta_i \le 2\eta_{i+EM}$. We can prove that for $i\in\{-1\}\cup\mathcal{I}^+$,[8]

$$\Delta_{i+1} \le \frac{V}{\alpha_1+i+1}, \quad (26)$$

where $V \triangleq \max\left\{(\alpha_0^2(C_1+C_2^{\text{SS}}))/(\alpha_0\mu-1), (\alpha_1+1)\Delta_0\right\}$. This is proven using mathematical induction.

Bound $\mathbb{E}_{\boldsymbol{\mathcal{H}}_{TEM-1},\mathcal{S}_T}[F(\boldsymbol{\omega}_T^{\text{SS}})] - F(\boldsymbol{\omega}^\star)$: Based on Assumptions 1 and 2, we have $\mathbb{E}_{\boldsymbol{\mathcal{H}}_{TEM-1},\mathcal{S}_T}[F(\boldsymbol{\omega}_T^{\text{SS}})] - F(\boldsymbol{\omega}^\star) \le \beta\Delta_{TEM}/2$. Based on (26), by setting $\alpha_0 = 2/\mu$ and $\alpha_1 = \max\{8\beta/\mu-1, EM\}$, we obtain (16).

[8] We also consider $i=-1$ in order to include the initial value Δ_0.

REFERENCES

[1] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int'l Conf. on Artificial Intelligence and Statistics*, Fort Lauderdale, FL, Apr. 2017.

[2] P. Kairouz *et al.*, "Advances and open problems in federated learning," *Foundations and Trends in Machine Learning*, vol. 14, no. 1, Mar. 2021.

[3] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "Adaptive federated learning in resource constrained edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1205–1221, Jun. 2019.

[4] G. Malinovsky, D. Kovalev, E. Gasanov, L. Condat, and P. Richtarik, "From local SGD to local fixed point methods for federated learning," in *Proc. Int'l Conf. Machine Learning (ICML)*, Jul. 2020.

[5] 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 Int'l Conf. Comp. Commun. (INFOCOM)*, May 2022.

[6] Y. Zhao and X. Gong, "Quality-aware distributed computation and user selection for cost-effective federated learning," in *Proc. IEEE Int'l Conf. Comp. Commun. Workshops (INFOCOM WKSHPS)*, May 2021.

[7] A. Li, L. Zhang, J. Tan, Y. Qin, J. Wang, and X.-Y. Li, "Sample-level data selection for federated learning," in *Proc. IEEE Int'l Conf. Comp. Commun. (INFOCOM)*, May 2021.

[8] J. Wang, S. Guo, X. Xie, and H. Qi, "Protect privacy from gradient leakage attack in federated learning," in *Proc. IEEE Int'l Conf. Comp. Commun. (INFOCOM)*, May 2022.

[9] T. Li, M. Sanjabi, and V. Smith, "Fair resource allocation in federated learning," in *Proc. Int'l Conf. Learning Representations (ICLR)*, Apr. 2020.

[10] L. Cui, X. Su, Y. Zhou, and J. Liu, "Optimal rate adaption in federated learning with compressed communications," in *Proc. IEEE Int'l Conf. Comp. Commun. (INFOCOM)*, May 2022.

[11] Y. Liu, L. Xu, X. Yuan, C. Wang, and B. Li, "The right to be forgotten in federated learning: An efficient realization with rapid retraining," in *Proc. IEEE Int'l Conf. Comp. Commun. (INFOCOM)*, May 2022.

[12] M. Tang and V.W.S. Wong, "An incentive mechanism for cross-silo federated learning: A public goods perspective," in *Proc. IEEE Int'l Conf. Comp. Commun. (INFOCOM)*, May 2021.

[13] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Commun. Surveys & Tuts.*, vol. 22, no. 3, pp. 2031–2063, Third Quarter 2020.

[14] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020.

[15] M. Tang and J. Huang, "How do you earn money on live streaming platforms?—A study of donation-based markets," *IEEE/ACM Trans. Netw.*, vol. 29, no. 4, pp. 1813–1826, Aug. 2021.

[16] J. Perazzone, S. Wang, M. Ji, and K. S. Chan, "Communication-efficient device scheduling for federated learning using stochastic optimization," in *Proc. IEEE Int'l Conf. Comp. Commun. (INFOCOM)*, May 2022.

[17] B. Buyukates and S. Ulukus, "Timely communication in federated learning," in *Proc. IEEE INFOCOM Age of Information Workshop*, May 2021.

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

[19] T. Huang, W. Lin, W. Wu, L. He, K. Li, and A. Y. Zomaya, "An efficiency-boosting client selection scheme for federated learning with fairness guarantee," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 7, pp. 1552–1564, Jul. 2021.

[20] M. Ribero, H. Vikalo, and G. De Veciana, "Federated learning under intermittent client availability and time-varying communication constraints," *IEEE Journal of Selected Topics in Signal Processing*, 2022 (Early Access).

[21] D. Avdiukhin and S. Kasiviswanathan, "Federated learning under arbitrary communication patterns," in *Proc. Int'l Conf. Machine Learning (ICML)*, Vienna, Austria, Jul. 2021.

[22] Y. Chen, Y. Ning, M. Slawski, and H. Rangwala, "Asynchronous online federated learning for edge devices with non-IID data," in *Proc. IEEE Int'l Conf. on Big Data*, Atlanta, GA, Dec. 2020.

[23] 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 Systems (MLSys)*, Austin, TX, Mar. 2020.

[24] S. P. Karimireddy, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, and A. T. Suresh, "SCAFFOLD: Stochastic controlled averaging for federated learning," *Proc. Int'l Conf. Machine Learning (ICML)*, Jul. 2020.

[25] G. Shen, D. Gao, L. Yang, F. Zhou, D. Song, W. Lou, and S. Pan, "Variance-reduced heterogeneous federated learning via stratified client selection," *arXiv preprint arXiv:2201.05762*, Apr. 2022.

[26] Y. LeCun, C. Cortes, and C. J.C. Burges, "The MNIST database of handwritten digits," http://yann.lecun.com/exdb/mnist/, accessed Jan. 5, 2023.

[27] A. Krizhevsky, V. Nair, and G. Hinton, "The CIFAR-10 dataset," https://www.cs.toronto.edu/~kriz/cifar.html, accessed Jan. 5, 2023.

[28] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of FedAvg on non-IID data," in *Proc. Int'l Conf. Learning Representations (ICLR)*, Apr. 2020.

[29] R. Singh and N. S. Mangat, *Elements of Survey Sampling*. Netherlands: Springer, Dordrecht, 1996.

[30] E. Liberty, K. Lang, and K. Shmakov, "Stratified sampling meets machine learning," in *Proc. Int'l Conf. Machine Learning (ICML)*, New York City, NY, Jun. 2016.

[31] B. Everitt, *Finite Mixture Distributions*. Springer Science & Business Media, 2013.

[32] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, Nov. 1987.

[33] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.