FedAR: Addressing Client Unavailability in Federated Learning with Local Update Approximation and Rectification

Chutian Jiang, Hansong Zhou, Xiaonan Zhang ✉, and Shayok Chakraborty

Department of Computer Science, Florida State University, Tallahassee, FL, USA
cj20cn@fsu.edu, hz21e@fsu.edu, xzhang@cs.fsu.edu, shayok@cs.fsu.edu

Abstract. Federated learning (FL) enables clients to collaboratively train machine learning models under the coordination of a server in a privacy-preserving manner. One of the main challenges in FL is that the server may not receive local updates from each client in each round due to client resource limitations and intermittent network connectivity. The existence of unavailable clients severely deteriorates the overall FL performance. In this paper, we propose *FedAR*, a novel client update *A*pproximation and *R*ectification algorithm for FL to address the client unavailability issue. FedAR can get all clients involved in the global model update to achieve a high-quality global model on the server, which also furnishes accurate predictions for each client. To this end, the server uses the latest update from each client as a surrogate for its current update. It then assigns a different weight to each client's surrogate update to derive the global model, in order to guarantee contributions from both available and unavailable clients. Our theoretical analysis proves that FedAR achieves optimal convergence rates on non-IID datasets for both convex and non-convex smooth loss functions. Extensive empirical studies show that FedAR comprehensively outperforms state-of-the-art FL baselines including FedAvg, MIFA, FedVARP and Scaffold in terms of the training loss, test accuracy, and bias mitigation. Moreover, FedAR also depicts impressive performance in the presence of a large number of clients with severe client unavailability.

Keywords: Federated learning · Client selection · Bias mitigation

1 Introduction

Federated learning (FL) allows multiple clients to collaboratively learn a powerful global machine learning model without sharing the training data with the server. As a privacy-preserving and communication-efficient distributed learning framework, FL has garnered substantial research attention and has surged as a key enabler of distributed intelligence in many real-world applications, such as next-word prediction on mobile keyboards [11] and medical record analysis in digital health [4]. In the vanilla FL algorithm, known as FedAvg [21], the server distributes the current global model to all the clients in each round, which serves as the basis for running several steps of stochastic gradient descent (SGD) on the local data for each client. The local updates are then sent

back to the server to update the global model. This process is iterated until the global model converges.

In FL, clients can be diverse, ranging from medical wearables and IoT devices to smartphones. Many of these clients operate as low-power devices and communicate over wireless networks. This presents a challenge to FedAvg, as clients may abort training midway due to issues like low battery levels or incoming calls [2, 10, 15, 21]. As a result, clients may fail to return their trained local updates to the server, especially when the communication from the clients to the server is hampered by poor channel quality and intermittent connectivity (also referred to as *unavailable / non-participating clients* or the *partial client participation problem*). In FedAvg, the inability to receive local updates from unavailable clients can cause a serious delay and it can even discard these updates when deriving the global model to maintain learning efficiency [26, 31, 32, 34]. Missing the expected local updates introduces an undesired bias against unavailable clients [1, 31]. This will result in the global model overfitting the characteristics of consistently available clients, thereby diminishing its performance for clients that participate less frequently and reducing its overall generalization capability [5, 12, 13, 22, 33].

The primary goal of this paper is to develop and validate an efficient FL algorithm termed *F*ederated Learning with local update *A*pproximation and *R*ectification (FedAR), which addresses the partial client participation problem. We first study the contributions of the latest observed local updates from unavailable clients to the global update. Our observation reveals that unavailable clients with varying inactive rounds exert diverse positive influences on the global update. Motivated by this insight, we propose a novel server-side aggregation strategy that incorporates local updates from unavailable clients in the global update. More importantly, our framework does not require any additional computation at the clients or introduce any extra communication between the clients and the server. FedAR utilizes the latest update from each client observed by the server as a surrogate of its current update, which is then used in updating the global model. Moreover, we devise an innovative weighting scheme to accommodate the variable influence on the global model from local updates of clients with differing inactive rounds. We slightly magnify the contributions from unavailable clients (based on the number of inactive rounds) in addition to the contributions from the available clients, to update the global model. To achieve this, we design the weight as a mildly increasing function of the number of inactive rounds of each client. This strategy enables the server to include the local data distribution information from unavailable clients in updating the global model, thereby circumventing the bias against these clients. Lastly, unlike traditional FL, FedAR does not assume that the server is aware of the total number of clients in advance. Instead, it dynamically counts the number of clients who get involved in the global model update, which better reflects real-world application scenarios. In light of the above discussion, we summarize our key contributions in this paper as follows:

- We propose FedAR, a novel FL algorithm that addresses the client unavailability issue. FedAR unevenly weighs the contributions from both available and unavailable clients in the global model update based on the number of their inactive rounds. Moreover, FedAR does not necessitate any additional computation at the clients,

nor does it demand any extra communication between the clients and the server. It does not require all clients to participate in FL in the first round either.
- We theoretically provide a convergence guarantee for FedAR for both convex and non-convex smooth loss functions on non-IID datasets across clients.
- We evaluate the performance of FedAR on three real-world datasets MNIST, CIFAR-10, and SVHN. Compared to the vanilla and the state-of-the-art FL baselines, FedAvg, MIFA, FedVARP, and Scaffold, FedAR can achieve a 75% improvement in test accuracy and a 50% reduction in training loss in the best case. Moreover, we empirically show that FedAR can better mitigate the bias against unavailable clients, as evidenced by the observation that the derived global model generates more accurate predictions for clients who have been intermittently inactive during the training process. FedAR also demonstrates impressive performance in the presence of a large number of clients with severe client unavailability.

2 Related Work

One of the main challenges of the vanilla FL algorithm, FedAvg, is the intermittent unavailability of clients. Specifically, the server will not update the global model until receiving local updates from all clients, which results in considerable training delay in the presence of client unavailability. Client sampling can be used as a remedy to this issue, where some clients are selected to participate in the global model update. The common client sample strategies include random sampling, significant sampling, and cluster sampling. Random sampling [21] selects clients at random whereas importance sampling [6, 7, 20] selects the most valuable clients in terms of data quantity, communication time, and local training results. In cluster sampling [3, 8, 9], clients are first divided into groups based on sample size, model similarity etc.; the clients in each group are then selected for global update. All these sampling strategies engage only available clients but ignore unavailable clients in the global update. Consequently, the global model biases towards the available clients that are selected repetitively [23], which would undermine the FL performance.

A body of research addresses the client unavailability issue by incorporating stale updates from unavailable clients into the training process, such as the Memory-augmented Impatient Federated Averaging (MIFA) algorithm [10] and the Federated VAriance Reduction for Partial Client Participation (FedVARP) algorithm [14]. Their major differences with FedAR are listed in Table. 1. In particular, seeking to maximize non-IID data coverage, MIFA gives equal weightage to updates from both available and unavailable clients, making it a biased scheme. Even worse, MIFA requires all clients to participate in the first training round, which is an unrealistic assumption. FedVARP allocates higher weights to the updates from available clients than to the updates from unavailable clients. It also attempts to reduce the variance to available clients caused by the partial client partition, which, however, is not empirically demonstrated. Similar to both MIFA and FedVARP, the FedAR algorithm reuses the latest observed update for each client as an approximation of its current update. Different from MIFA, FedAR formulates a novel weighting scheme to efficiently involve unavailable clients with various inactive rounds in the global model update. Moreover, FedAR does not require all

clients to participate in FL in the first training round. Motivated by [30], FedAR assigns higher weights to the updates of the unavailable clients with a larger number of inactive rounds, i.e., we amplify the local updates from unavailable clients, which is contrary to FedVARP. Our experimental results show the efficacy of FedAR in terms of overall convergence, test accuracy and bias mitigation, compared to relevant baselines.

	MIFA	FedVARP	FedAR
Enhance the FL efficiency with uncertain availability of clients			
Issue addressed	maximize data coverage	reduce variance of available local updates	reduce bias against unavailable local updates
Rationale on local updates	all have the same contribution	available ones have higher contributions	unavailable ones can also have contributions
Solution	allocate the same weight to all local updates	allocate higher weights to available local updates	allocate higher weights to unavailable local updates with higher contributions
All clients assumption	must respond in the first round	**not necessarily** respond in the first round	

Table 1: Comparison of FedAR with MIFA and FedVARP

3 Problem Setup

We consider that a set of clients $\mathcal{N} = \{1, 2, \cdots, N\}$ with restricted power and computational resources collaborate with a server to execute FL over T rounds. The datasets for local training are subject to non-IID distributions. The clients and the server iteratively communicate over wireless networks to obtain a global model w aiming at minimizing the global loss function:

$$\min f(w) = \frac{1}{N} \sum\nolimits_{i=1}^{N} f_i(w), \tag{1}$$

where $f_i(w)$ is the loss function for client i.

3.1 Basic Algorithm of FL

We begin by recalling the vanilla FL setting in FedAvg. In round $t-1, t \in \{1, \cdots, T\}$, the server broadcasts the global model w_{t-1} to all the clients. Each client $i \in \mathcal{N}$ uses its own private dataset to execute K steps of Stochastic Gradient Descent (SGD) for the local update. For each step $k \in K$:

$$w^i_{t,k+1} = w^i_{t-1,k} - \eta_{t-1} \nabla f_i(w^i_{t-1,k}), \tag{2}$$

where η is the local learning rate and $\nabla f_i(\cdot)$ represents the gradient. Each client then sends back its local update to the server; the server aggregates all the client updates to derive the global model as:

$$w_t = \frac{1}{N} \sum\nolimits_{i=1}^{N} w^i_{t,K}. \tag{3}$$

Problem in FedAvg. Practically, due to the limited resources of each client and the intermittent network connectivity, the server may not receive the local updates $w^i_{t,K}$ from all the clients; these clients are called *unavailable* / *non-participating* clients. Due to this, FedAvg delays or even aborts the local updates from unavailable clients during the global update, causing an undesirable bias against these unavailable clients. However, the local updates from the unavailable clients also contain valuable information, which can be useful in global model updates. We conduct a toy experiment on a simple, restricted setup to demonstrate this idea and provide motivation for our approach.

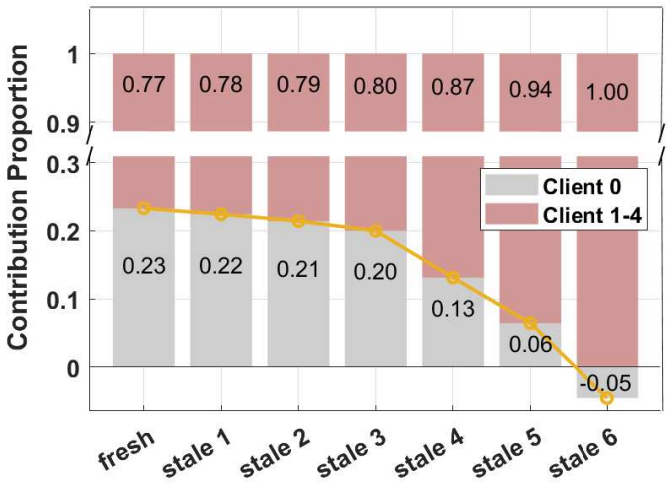


Fig. 1: Contribution of each client to the global model. "stale i" denotes that Client 0 has been inactive for the last i rounds. "fresh" denotes that all the clients are active for all the 9 rounds. A high staleness level indicates more inactive rounds

3.2 Motivation

Let us assume a standard FL setting where 5 clients (numbered 0 through 4) collaborate with a central server on a classification task using the CIFAR-10 dataset [17]. The server and clients execute a total of 9 rounds of communication. We conduct 7 different experiments, as shown by the vertical bars in Fig. 1. In all the experiments, client 1 to client 4 are always available across all the 9 rounds of communication. Client 0, conversely, becomes inactive after a certain number of rounds in each experiment. In Fig. 1, the term "stale i" refers to client 0 being active for the initial $9 - i$ rounds and then inactive for the subsequent i rounds. For instance, "stale 3" indicates that client 0 is active from rounds 1 to 6 but inactive during rounds 7 to 9. In this case, we aggregate the most recent local updates (from the 9^{th} round) for clients 1 to 4 and the local update from the 6^{th} round for client 0 (last active round) to update the global model.

"fresh" denotes the case where all the 5 clients were available across all the 9 rounds of communication. After the 9^{th} round, we use the Shapley Value (SV) [25] to quantify the contribution of each local update to the global model. Shapley value is a classical concept in cooperative game theory, and it is extensively used to evaluate client contributions in FL [27–29]. We compute each client's SV based on the global model's test accuracy, which is obtained by different combinations of the local client updates for the different experiments. We sum up all the SV and represent the contribution of client 0 and clients 1 to 4 as a percentage; the larger the value, the greater the contribution. From Fig. 1, it can be observed that as the staleness level of client 0 increases (larger number of inactive rounds), its contribution to the global model (height of the gray bar) decreases. At stale 6, the contribution of client 0 is negative, meaning that its local update has an adverse effect on the global model. Based on the above toy experiment, we draw the following conclusions:

- The stale local updates from unavailable clients can still contribute to the global model (as evident from the gray bars in stale 1 through stale 5).
- The contribution of the stale local updates decreases with increasing staleness level, suggesting it may be necessary to assign higher weights (during the global update) to stale local updates with more inactive rounds.
- An excessively high staleness level is detrimental to the performance of the global model. In the global update, it may not be necessary to include these local updates.

Given these observations, we propose our FedAR algorithm, as detailed below.

4 FedAR Algorithm

FedAR is designed as a simple and effective algorithm by involving local updates of unavailable clients in the global model update on the server. In addition, given that a client's unavailability leads to decreased contributions, we assign weights to different local updates accordingly. Our goal is to enhance FL performance by efficiently involving updates from all clients in the global model update. Specifically, FedAR consists of two components: *local update approximation* and *local update rectification*. In each round, the server sets a maximum waiting time for the local updates from all clients. When the maximum waiting time is reached, the server estimates the local updates that would be obtained from the unavailable clients. The weighted average over all the local updates is then performed to derive the global model for the next round. We describe our system in detail next.

Local Update Approximation To approximate the local updates, the server maintains an update-matrix $\mathbf{G}[t] = [G_1[t]; \cdots; G_i[t]; \cdots; G_N[t]]$ saving its most recent observed local updates from all clients. Initially, $\mathbf{G}[0]$ is a zero matrix. In round t, $G_i[t]$ will only be replaced if the server obtains the client i's local update $w^i_{t,K}$. Otherwise, $G_i[t]$ will not change. Let $\mathcal{A}(t) \subset \mathcal{N}$ represent the set of available clients whose updates are successfully received by the server in round t. Mathematically, we have,

$$G_i[t] = \begin{cases} \frac{1}{\eta_t}(w_t - w^i_{t,K}) & \text{if client } i \in \mathcal{A}(t) \\ G_i[t-1] & \text{otherwise.} \end{cases} \tag{4}$$

FedAR uses $G_i[t] \in \mathbf{G}[t]$ as the estimates for local updates while deriving the global model. The global model is thus able to include the data distribution from the unavailable clients, which will help mitigate the bias against them.

Local Update Rectification Fig. 1 shows that stale local updates with different inactive rounds have various contributions to the global model, inspiring us to weigh local updates during the global update. We propose to assign weights to local updates based on the number of their inactive rounds, and by doing so, we expect to enhance the contributions from unavailable clients and further mitigate the bias.

Formally, the server maintains an update-array $\tau(t-1) = [\tau(1, t-1), \cdots, \tau(i, t-1), \cdots, \tau(N, t-1)]$ to record the number of inactive rounds for all clients. $\tau(0)$ is initialized as a zero array. In round t, if the update from client i is received, the server resets $\tau(i, t)$ to 0. Otherwise, the server increases $\tau(i, t-1)$ by 1 to get $\tau(i, t)$. We express $\tau(i, t)$ as:

$$\tau(i,t) = \begin{cases} 0 & \text{if client } i \in \mathcal{A}(t) \\ \tau(i,t-1)+1 & \text{otherwise.} \end{cases} \tag{5}$$

Based on $\tau(i, t)$, we design a weight function $\psi_{i,t}$ to quantify the contribution from client i to the global update. The general expression of $\psi_{i,t}$ is given as:

$$\psi_{i,t} = \begin{cases} 0 & \text{if } \tau(i,t) \geq g(t) \\ \min([\tau(i,t)+1]^{\rho}, 2) & \text{otherwise.} \end{cases} \tag{6}$$

If the client i is available at round t, i.e., $\tau(i, t) = 0$, we have $\psi_{i,t} = 1$, which aligns with FedAvg. We introduce $g(t)$ to prevent local updates with many inactive rounds from negatively impacting the global model and to remove such updates from the current global update. $g(t)$, as a function of round t, is different based on whether we are optimizing a convex loss function or a non-convex loss function. We will discuss it in more detail in Section 5 (Theoretical Analysis).

Since unavailable clients with more inactive rounds contribute less to the global update, we assign them higher weights to increase their contributions, as shown in Eq. (6). However, an extremely high weight $\psi_{i,t}$ will cause the unavailable clients to dominate the global model update, which would induce bias against available clients. We therefore introduce the hyperparameter $\rho \in [0, 1]$ in Eq. (6) to restrict the growth of $\psi_{i,t}$. We also set the maximum value of $\psi_{i,t}$ (ψ_{max}) to 2 to guarantee the convergence of FedAR. Please refer to the Appendix for more details on the convergence analysis.

Global Model Update Clients arbitrarily participate in global model update in each round due to their limited resources and intermittent network connectivity. Hence, the server does not know the exact number of clients in advance; instead, it dynamically counts the clients that contribute to the global model update, i.e. those clients that are either available or unavailable but not too stale. Suppose there are N_t contributing clients in round t. With $G_i[t]$ and $\psi_{i,t}$, FedAR updates the global model as follows:

Algorithm 1: FedAR

Input: initial w_0, learning rate η_t, local step K, total round number T, total client number N
Output: The derived global model w_T
Server executes:

1: Initialize $\psi_{i,1} = 1$, $\tau(i,1) = 0$, and $G_i[0] = 0$, $\forall i$, temporary client set $\mathcal{E}$
2: **for** t=1,2, $\cdots$, T **do**
3: $\mathcal{E} \leftarrow \mathcal{E} \cup \{\text{new active client}\}$, $N_t = |\mathcal{E}|$.
4: **for** i=1,2 $\cdots$, N **in parallel do**
5: **if** client i is available **then**
6: $G_i[t] \leftarrow$ DeviceUpdate(i,w_t)
7: $\tau(i,t) = 0$
8: **else**
9: $\tau(i,t) = \tau(i,t) + 1$
10: **end if**Calculate the $\psi_{i,t}$ by Eq. (6)
11: **if** $\psi_{i,t} = 0$ **then**
12: $N_t = N_t - 1$
13: **end if**
14: **end for**
15: $w_{t+1} \leftarrow w_t - \frac{\eta_t}{N_t} \sum_{i=1}^{N} G_i[t]\psi_{i,t}$
16: **end for**

DeviceUpdate(i,w_t):

1: $w_{t,0}^i \leftarrow w_t$
2: **for** k = 0,1,$\cdots$, $K-1$ **do**
3: $w_{t,k+1}^i \leftarrow w_{t,k}^i - \eta_t \nabla f_i(w_{t,k}^i)$
4: **end for**
5: Return $\frac{1}{\eta_t}(w_t - w_{t,K}^i)$

$$w_{t+1} = w_t - \frac{\eta_t}{N_t} \sum\nolimits_{i=1}^{N} G_i[t]\psi_{i,t}. \tag{7}$$

Combined with Eq. (4), Eq. (7) ensures that the update matrix $G_i[t]$ always reflects the most recent client updates, while being able to reasonably consider the contributions of all clients when the global model is updated. In addition, although Eq. (7) seems to have all clients in the global model update, some clients do not get involved. They are either the clients that have never participated in FL, i.e., $G_i[t] = 0$, or the clients that have been inactive for many rounds, i.e., $\psi_{i,t} = 0$. Hence, Eq. (7) aligns with our idea of engaging only the contributing clients in the global model update.

Algorithm 1 shows the details of FedAR. We use the "temporary client set $\mathcal{E}$" to include the clients that have ever participated in the global model update in Line 3. Initially, N_t is the number of clients in $\mathcal{E}$. When the client i has been inactive for many rounds, i.e., $\psi_{i,t} = 0$, it will be excluded from the global model update, i.e., $N_t = N_t - 1$ in Line 11. Ultimately, N_t counts the contributing clients as in Eq. (7).

Regarding privacy enhancements in FL, FedAvg suggests that Differential Privacy (DP) can improve data privacy performance. However, our work is not primarily focused on privacy protection, and as such, an in-depth examination of this topic will not be included in our current research.

5 Theoretical Analysis of FedAR

In this section, we analyze the convergence of the proposed FedAR for convex and non-convex smooth loss functions.

5.1 Convex Loss Function

To analyze the convergence of FedAR for a convex loss function, we make the following assumptions regarding $f_i(w), i = 1, 2, \cdots, N$.
Assumption 1: L-smoothness. The loss function $f_i(w)$ is L-smooth. That is: for all $x, y \in \mathbb{R}, f(x) - f(y) \leq \langle \nabla f(y), x - y \rangle + \frac{L}{2}\|y - x\|^2$ with $L > 0$.
Assumption 2: μ-strong convex. The loss function $f_i(w)$ is μ-strong convex. That is: for all $x, y \in \mathbb{R}, f(x) - f(y) \geq \langle \nabla f(y), x - y \rangle + \frac{\mu}{2}\|y - x\|^2$ with $\mu > 0$.
Assumption 3: Variance bound. The variance of the unbiased estimator of $\nabla f_i(w)$ in round t is upper bounded, where $\mathbb{E}\{\|\tilde{\nabla} f_i(w) - \nabla f_i(w)\psi_{i,t}\|^2\} \leq \sigma^2$.

Theorem 1: Suppose the objective loss function $f_i(w)$ satisfies Assumptions 1 to 3, $\tau_{max} \leq g(t)$. By setting the learning rate $\eta_t = \frac{4}{\mu(t+a)}$ and constant $a = 100(\frac{L}{\mu})^{1.5}$, after T rounds, FedAR satisfies:

$$\begin{aligned}\mathbb{E}[f(\overline{w}_T)] - f(w_*) = & \mathcal{O}(\frac{\sigma^2(1+\overline{\tau}_T)}{\mu KNT}) \\ & + \mathcal{O}(\frac{F + \|w_1 - w_*\|^2 + \tau_{max}^2 L\sigma^2 N\psi_{max}}{K\mu^3 T^2}),\end{aligned}$$

where τ_{max} is the maximum number of $\tau(i,t)$ over all clients and rounds. $g(t) = t_0 + \frac{1}{b}t$ for a constant $t_0 > 0$ and $b > 2$, $F = LKND + L(K-1)^2 \cdot (DN^2 + \frac{\sigma^2}{K})$, $\overline{w}_T = \frac{\sum_{t=1}^{T}(t+a-1)(t+a-2)w_t}{W_T}$, $W_T = \sum_{t=1}^{T}(t+a-1)(t+a-2)$, $\overline{\tau}_T = \frac{1}{N(T-1)}\sum_{t=1}^{T-1}\sum_{i=1}^{N}\tau(i,t)$, and $D = \frac{1}{N}\sum_{i=1}^{N}\|\nabla f_i(w_*)\|^2$.
Remark 1. In Theorem 1, both the first and the second terms tend to zero as T increases, indicating that FedAR converges at the rate of $\mathcal{O}(1/T)$. The first term's convergence is related to the average inactive round number $\overline{\tau}_T$. We can find that too high a value of $\overline{\tau}_T$ will negatively impact convergence, which is consistent with our observation in Section 3.2 (Motivation). Also, convergence is adversely affected when most clients remain unavailable for a long time, i.e., a large $\overline{\tau}_T$. Besides, we observe that weight function ψ has a relatively negligible effect on the convergence rate. This can be attributed to our restriction on ψ in Eq. (6) to prevent it from becoming excessively large with an increase of τ. This is because a larger ψ could lead to the dominance of clients with more inactive rounds during the global model update.

5.2 Non-Convex Loss Function

To analyze the convergence of FedAR for a non-convex smooth loss function, we make the following assumptions regarding $f_i(w), i = 1, 2, \cdots, N$.

Assumption 4: Hessian Lipschitz. The Hessian of a twice differentiable function f: $\mathbb{R}^d \to \mathbb{R}$ is λ-Lipschitz continuous if $\left\|\nabla^2 f(x) - \nabla^2 f(x)\right\| \leq \lambda \|x - y\|$ for all x, y.
Assumption 5: Gradient noise. The noise of the local stochastic gradients in round t is upper bounded by a constant δ: $\left\|\tilde{\nabla} f_i(w) - \nabla f_i(w)\right\| \leq \delta$.
Assumption 6: Gradient dissimilarity. $\exists\, \alpha > 0$ and $\beta_i > 0$: $\|\nabla f_i(w)\|^2 \leq \alpha \|\nabla f_i(w)\|^2 + \beta_i > 0$ and we define $\beta = \frac{1}{N}\sum_{i=1}^{N} \beta_i$.
Assumption 7: There exists a constant v_i such that $\tau(i,t) \leq v_i$ for $\forall i \in \mathcal{N}$, and define $\overline{v} = \frac{1}{N}\sum_{i=1}^{N} v_i$, $v_{max} = max_{i \in \mathcal{N}} v_i$.
Theorem 2: Suppose Assumptions 1 to 7 hold, set learning rate $\eta = \sqrt{\frac{N}{KTL(1+\overline{v})}}$, $T \geq max\{32\alpha LNK, 16LN^5K, \frac{8KNv_{max}^2(L^2+\lambda\delta N^2)}{L}\}$, and $\tau_{max} \leq g(t)$. After T rounds, FedAR satisfies:

$$\mathbb{E}[\|\nabla f(w_T)\|^2] \leq \mathcal{O}(R\sqrt{\frac{L(1+\overline{v})}{TKN}}(f(w_1) - f^* + \sigma^2) + \frac{\alpha\sigma^2\overline{v}LKN^2\psi_{max}}{T} + \frac{\sigma^2\lambda\delta N\psi_{max}}{LT} + \frac{F_1}{T}),$$

where $g(t) = \frac{1}{4}\sqrt{\frac{L}{(L^2+\beta\lambda N)KN}} \times max\{\sqrt{t}, \sqrt{t_0}\}$ for a constant $t_0 > 0$, $F_1 = (\alpha + 1)(LKN\sigma^2\overline{v} + LKN\sigma v_{max}\sqrt{\beta + \frac{\sigma^2}{KN}}) + \frac{(L^2+\lambda\delta N^2)\sigma v_{max}}{L} + (K-1)(2\beta + \frac{\sigma^2}{K})$, and $R = \frac{8\psi_{max}^2}{4\psi_{max}^2 - 1}$.
Remark 2. In Theorem 2, the convergence of FedAR for a non-convex smooth loss function is dominated by the first term, which converges at the rate of $\mathcal{O}(\sqrt{1/T})$. This dominant term is mainly influenced by the initial error$f(w_1) - f^*$, the variance bound σ, and the average upper bound of inactive round number across clients $\overline{v}$. In addition, we observe that weight function ψ appears in the dominant term through the parameter R. Regardless of how ψ changes, the value of R tends towards a constant, and thus the impact produced by ψ is not significant. Compared to ψ, τ and N have a greater influence on the convergence via impacting the dominant term. We can draw similar conclusions as Theorem 1: as more clients continue to join the FL, more rounds are required to achieve convergence. Meanwhile, the fact that τ_{max} is a major variable affecting convergence aligns with our initial observations in Section 3.2 (Motivation); that is, the local updates with more inactive rounds negatively impact the global model's performance and further prevent the global model from converging. Thus, there must exist a critical value $g(t)$ as we express in Eq. (6) to exclude those clients from the global update to ensure the model convergence, i.e.,. the clients whose inactive round number exceeds $g(t)$ will not be considered.

Please refer to our Appendix for the proof of Theorem 1 and Theorem 2, as well as Remark 3 on Theorem 2.

6 Experiments and Evaluations

In this section, we evaluate the performance of FedAR by conducting extensive experiments on a desktop with the GeForce RTX 3060 graphic card.

6.1 Experimental Setup

System Settings. We conduct the FL experiments with one server and 100 clients. Let p_i denote the probability that client i is available during any given round. The availability of all clients is independent, with a minimum probability of p_{min}, indicating that the client availability probability varies from p_{min} to 1. This is a practical setting given that clients have their unique resource constraints and face distinct wireless environments. We examine both the challenging and mild client unavailability, where $p_{min} = 0.1$ and 0.5, respectively. In summary, most clients are inactive for 5 - 20 rounds, with a few clients being inactive for more than 40 rounds.

Data and Model. We evaluate FedAR on three real-world datasets: MNIST [18], CIFAR-10 [17], and SVHN [24]. To ensure non-IID data distribution among all clients, we assume all datasets to be evenly distributed on all clients, and each client to contain only two classes of data. We use the logistic regression for MNIST, Lenet-5 for CIFAR-10, and Resnet-18 for SVHN as the local models. We set all experiments' initial local learning rate as $\eta_0 = 0.1$, local training step as $K = 5$, local batch size as 64, and hyperparameter as $\rho = 0.1$. We set weight decay as 0.001 during the local SGD.

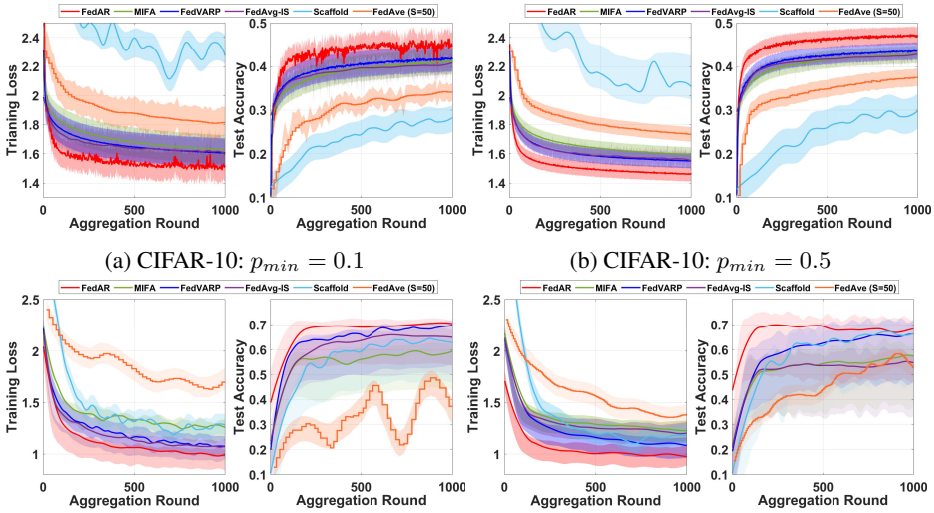
Baselines. We compare FedAR with recent FL baselines: (1) *MIFA* [10]. It assigns the same weight to both available and unavailable clients; (2) *FedVARP* [14]. It assigns higher weights to available clients' updates, while the weights for unavailable clients remain unchanged; (3) *FedAvg-IS*. It engages only available clients in global update using the FedAvg algorithm. The local updates are weighted by clients' availability probabilities; (4) *FedAvg (S=50)*. It involves at most half of available clients in the global update with the FedAvg algorithm. Given 100 clients, at most 50 clients join the global update; and (5) *Scaffold* [16]. It is a FL algorithm designed to improve the quality of global model updates by applying personalized control variate adjustments to each client; it does not consider client unavailability.

6.2 Experimental Results[1]

Overall Convergence Performance. We evaluate the convergence performance of FedAR on different datasets in both the challenging and mild settings in Fig. 2. We find that FedAR has a similar convergence speed as FedAvg-IS, MIFA, and FedVARP. Notably, on CIFAR and SVHN datasets, the convergence speed of FedAR is markedly superior to that of Scaffold. This observation is consistent with our theoretical analysis that our designed weight function ψ has negligible negative impacts on convergence.

When $p_{min} = 0.1$, Fig. 2a shows that FedAR on CIFAR-10 reduces the training loss to 1.5 and attains the highest test accuracy of 44%, an enhancement of over 3% compared to baseline algorithms. Fig. 2c shows that FedAR is the only algorithm achieving a training loss below 1 and a test accuracy over 70% on SVHN. When more clients are available, i.e., $p_{min} = 0.5$, FedAR in Fig. 2b greatly boosts the test accuracy to 46% on CIFAR-10. Additionally, we find that FedAR consistently reaches a test accuracy of around 70% in most training rounds on SVHN, and outperforms all the baselines.

[1] For clear observation, we recommend viewing all figures about experimental results in color



(a) CIFAR-10: $p_{min} = 0.1$

(b) CIFAR-10: $p_{min} = 0.5$

(c) SVHN: $p_{min} = 0.1$

(d) SVHN: $p_{min} = 0.5$

Fig. 2: Convergence, training loss and test accuracy performance

Dataset	Baselines				
	FedVARP	MIFA	Scaffold	FedAve(s=50)	FedAve-IS
Cifar10;p=0.1	$1.15*10^{-163}$	$1.15*10^{-194}$	$1.87*10^{-194}$	$1.53*10^{-110}$	$4.36*10^{-180}$
Cifar10;p=0.5	0.0	0.0	$1.38*10^{-246}$	$3.46*10^{-316}$	$2.02*10^{-285}$
SVHN;p=0.1	0.00029	$1.49*10^{-54}$	$5.72*10^{-35}$	$1.63*10^{-60}$	$4.77*10^{-45}$
SVHN;p=0.5	$1.34*10^{-52}$	$5.91*10^{-111}$	$1.71*10^{-81}$	$2.25*10^{-77}$	$1.96*10^{-116}$

Table 2: P-Value analysis of FedAR performance

We also conduct statistical tests of significance using paired t-test to assess whether the improvement in performance achieved by FedAR is statistically significant. We compare the test accuracy of FedAR against each of the baselines individually for both CIFAR-10 and SVHN, and for $p_{min} = 0.1$ and $p_{min} = 0.5$. The results are illustrated in Table 2; each entry in the table denotes the p-value of the paired t-test between FedAR and the corresponding baseline (denoted in the columns) for the corresponding dataset (denoted in the rows). From the table, we find that the improvement in performance achieved by FedAR is statistically significant ($p < 0.001$) compared to all the baselines, consistently for both the datasets and both values of p_{min}. These results further corroborate the promise and potential of FedAR. FedAR also shows superior performance on the MNIST dataset with lower training losses and higher test accuracy upon convergence, as elaborated in the Appendix.

Bias Mitigation. We study the bias mitigation performance of FedAR on CIFAR-10 in the challenging setting, where $p_{min} = 0.1$. Specifically, the global model is used to

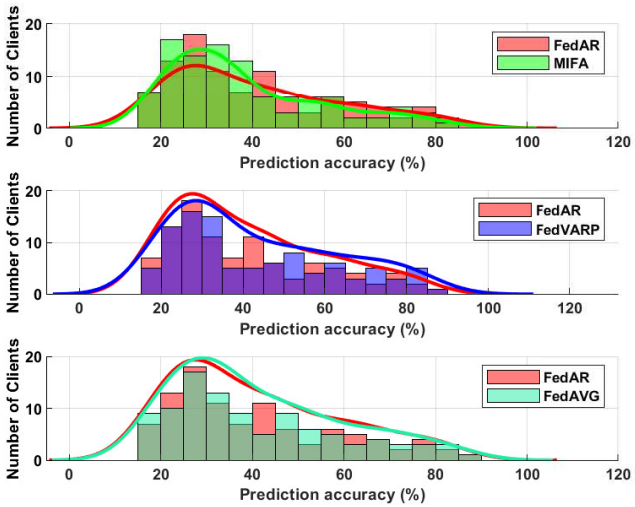


Fig. 3: Accuracy distributions

make predictions for each client after convergence, and we study the consistency of the prediction accuracies across all clients. In addition to MIFA and FedVARP, we compare FedAR with the ideal situation of FedAvg, where all the clients are continuous available throughout the entire training process.

ALGO	Mean (%)	Var	Worst 10% (%)	Best 10% (%)
FedAR	40.9±18.1	325	20.8±4.5	67.7±8.7
MIFA	34.0±13.6	182.5	19.7±4.2	53.8±8.8
FedVARP	41.3±20.7	432.5	19.4±2.6	73.9±11.6
FedAvg	41.0±18.0	321.6	21.2±4.3	69.5±12.6

Table 3: Accuracy statistics

Table. 3 depicts the statistics (mean $\pm$ std and variance) of the prediction accuracy across clients. In addition, we record the prediction accuracy of the worst 10% clients and the best 10% clients, denoted by "Worst 10%" and "Best 10%" respectively [19]. From Table. 3, we observe that the "Mean", "Worst 10%", and "Best 10%" prediction accuracy of FedAR closely align with FedAvg. This suggests that the performance of FedAR is comparable to the ideal situation of full client availability. Furthermore, FedAR achieves an average accuracy approximately 6% higher than MIFA, which requires all the clients to participate in the first training round. Although the average prediction accuracy of FedVARP is marginally higher than FedAR, it exhibits a con-

siderably higher variance of 432.5, over 100 more than FedAR. Such a high variance indicates a significant variation in prediction accuracy across different clients in FedVARP.

To more intuitively evaluate the bias mitigation performance, we visually depict the distribution of the number of clients and its Probability Density Function (PDF) of prediction accuracy in Fig. 3. Compared to MIFA, FedAR enables a larger number of clients to achieve a prediction accuracy of 40% or higher. Additionally, within the accuracy interval between 25% and 65%, the PDF curve of FedAR surpasses that of FedVARP. Outside this interval, PDF curve of FedAR falls below that of FedVARP. This pattern indicates that the prediction accuracies in FedAR are more centralized around the mean value (40%). This explains the high variance values of FedVARP in Table. 3. Furthermore, the PDF curve of FedAR almost coincides with that of FedAvg. This indicates that even under the challenging client unavailability ($p_{min} = 0.1$), FedAR maintains prediction accuracy distribution similar to the ideal full client availability situation. Both Table. 3 and Fig. 3 confirm that FedAR can effectively mitigate the bias despite severe client unavailability.

Hyperparameter Evaluation. We study the effect of hyperparameters under the challenging setting of $p_{min} = 0.1$ on CIFAR-10. Please refer to our Appendix for the performance analysis on SVHN and the evaluation for ρ value in Eq. (6).

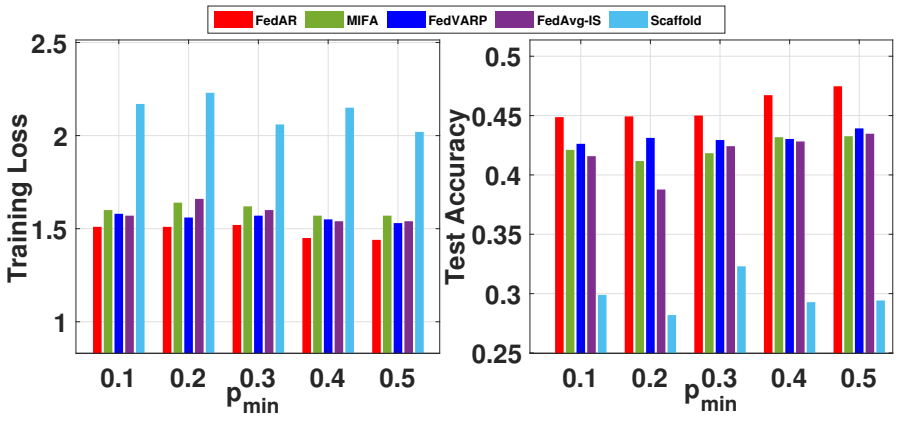


Fig. 4: Effect of p_{min}

Minimum Client Participation Probability p_{min}. We evaluate FedAR under various client participation probability, i.e., p_{min} spanning from 0.1 to 0.5. We exclude FedAvg (S=50) due to its notably inferior performance compared to other baselines. As shown in Fig. 4, FedAR consistently outperforms all the baselines for every p_{min}. Additionally, we note a marginal enhancement in FedAR's performance as p_{min} increases. When $p_{min} = 0.5$, FedAR achieves an accuracy of 47% whereas the accuracy of all the baselines is below 45%. This is because a higher participation probability reduces the average number of inactive rounds, thus positively impacting the FL performance.

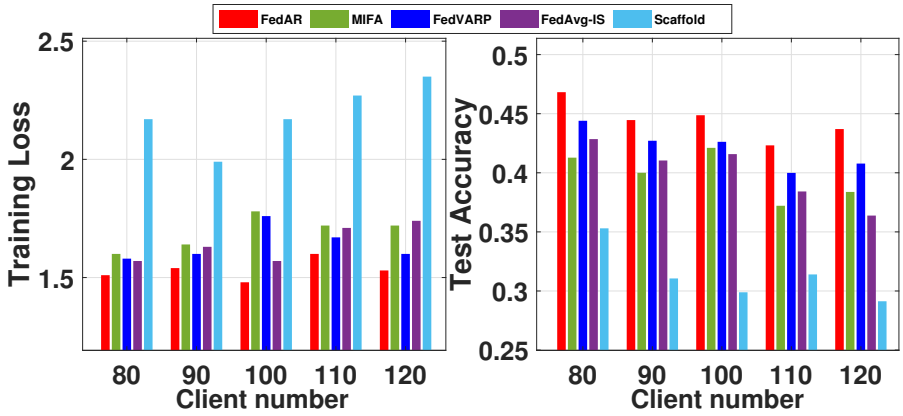


Fig. 5: Effect of N

Number of Clients N. We evaluate FedAR with varying numbers of clients from 80 to 120. As shown in Fig. 5, as N increases, all algorithms show an increasing trend in training loss and a decreasing trend in test accuracy, among which FedAR achieves the best performance. Specifically, FedAR marginally increases the training loss only from 1.5 to 1.6 when N is increased from 80 to 120. The lowest accuracy of FedAR is 43% in the case of $N = 110$. In contrast, the performance of other baselines degrades significantly with the increase in the number of clients. Except for FedVARP, the test accuracy of the rest of the baselines has fallen below 40%. The surge in the number of clients inherently leads to a rise in unavailable clients, posing challenges across all algorithms. This suggests that FedAR is more adept at handling a large number of clients, making it ideal for large-scale FL, especially in the presence of significant client unavailability.

7 Conclusion

In this paper, we propose a novel FL algorithm, FedAR, to address the client unavailability. We found that clients with different numbers of inactive rounds have diverse contributions to the current global update. Based on this observation, we design a novel weighting strategy that not only engages the unavailable clients in the global model update, but also quantifies their contributions based on the number of their inactive rounds. We theoretically prove the convergence of FedAR for both convex and non-convex smooth loss functions with non-IID data across clients. Our experimental results demonstrate that FedAR significantly outperforms competing FL baselines FedAvg, MIFA, FedVARP and Scaffold with respect to the training loss, the test accuracy, and the bias mitigation. FedAR further demonstrates remarkable performance and surpasses those baselines in large-scale FL with severe client unavailability. As part of future work, we will study the performance of FedAR under other practical challenges such as missing data and class imbalance across clients.

8 Acknowledgment

The work of X. Zhang is partially supported by the National Science Foundation under Grant Number: CCF-2312617. The work of S. Chakraborty is partially supported by the National Science Foundation under Grant Number: IIS-2143424 (NSF CAREER Award).

References

1. Abdelmoniem, A.M., Sahu, A.N., Canini, M., Fahmy, S.A.: Refl: Resource-efficient federated learning. In: Proceedings of the Eighteenth European Conference on Computer Systems. pp. 215–232 (2023)
2. Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., Kiddon, C., Konečnỳ, J., Mazzocchi, S., McMahan, B., et al.: Towards federated learning at scale: System design. Proceedings of Machine Learning and Systems **1**, 374–388 (2019)
3. Briggs, C., Fan, Z., Andras, P.: Federated learning with hierarchical clustering of local updates to improve training on non-iid data. In: 2020 International Joint Conference on Neural Networks (IJCNN). pp. 1–9. IEEE (2020)
4. Brisimi, T.S., Chen, R., Mela, T., Olshevsky, A., Paschalidis, I.C., Shi, W.: Federated learning of predictive models from federated electronic health records. International journal of medical informatics **112**, 59–67 (2018)
5. Chen, S., Li, B.: Towards optimal multi-modal federated learning on non-iid data with hierarchical gradient blending. In: IEEE INFOCOM 2022-IEEE conference on computer communications. pp. 1469–1478. IEEE (2022)
6. Cho, Y.J., Gupta, S., Joshi, G., Yağan, O.: Bandit-based communication-efficient client selection strategies for federated learning. In: 2020 54th Asilomar Conference on Signals, Systems, and Computers. pp. 1066–1069. IEEE (2020)
7. Cho, Y.J., Wang, J., Joshi, G.: Towards understanding biased client selection in federated learning. In: International Conference on Artificial Intelligence and Statistics. pp. 10351–10375. PMLR (2022)
8. Fraboni, Y., Vidal, R., Kameni, L., Lorenzi, M.: Clustered sampling: Low-variance and improved representativity for clients selection in federated learning. In: International Conference on Machine Learning. pp. 3407–3416. PMLR (2021)
9. Ghosh, A., Chung, J., Yin, D., Ramchandran, K.: An efficient framework for clustered federated learning. Advances in Neural Information Processing Systems **33**, 19586–19597 (2020)
10. Gu, X., Huang, K., Zhang, J., Huang, L.: Fast federated learning in the presence of arbitrary device unavailability. Advances in Neural Information Processing Systems **34**, 12052–12064 (2021)
11. Hard, A., Rao, K., Mathews, R., Ramaswamy, S., Beaufays, F., Augenstein, S., Eichner, H., Kiddon, C., Ramage, D.: Federated learning for mobile keyboard prediction. arXiv preprint arXiv:1811.03604 (2018)
12. Horvath, S., Laskaridis, S., Almeida, M., Leontiadis, I., Venieris, S., Lane, N.: Fjord: Fair and accurate federated learning under heterogeneous targets with ordered dropout. Advances in Neural Information Processing Systems **34**, 12876–12889 (2021)
13. Huang, W., Ye, M., Du, B.: Learn from others and be yourself in heterogeneous federated learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10143–10153 (2022)
14. Jhunjhunwala, D., SHARMA, P., Nagarkatti, A., Joshi, G.: Fedvarp: Tackling the variance due to partial client participation in federated learning. In: The 38th Conference on Uncertainty in Artificial Intelligence (2022)

15. Kairouz, P., McMahan, H.B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A.N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al.: Advances and open problems in federated learning. Foundations and Trends® in Machine Learning **14**(1–2), 1–210 (2021)
16. Karimireddy, S.P., Kale, S., Mohri, M., Reddi, S., Stich, S., Suresh, A.T.: Scaffold: Stochastic controlled averaging for federated learning. In: International Conference on Machine Learning. pp. 5132–5143. PMLR (2020)
17. Krizhevsky, A.: Learning multiple layers of features from tiny images. University of Toronto (05 2012)
18. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998). https://doi.org/10.1109/5.726791
19. Li, T., Sanjabi, M., Beirami, A., Smith, V.: Fair resource allocation in federated learning. arXiv preprint arXiv:1905.10497 (2019)
20. Luo, B., Xiao, W., Wang, S., Huang, J., Tassiulas, L.: Tackling system and statistical heterogeneity for federated learning with adaptive client sampling. In: IEEE INFOCOM 2022-IEEE conference on computer communications. pp. 1739–1748. IEEE (2022)
21. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial intelligence and statistics. pp. 1273–1282. PMLR (2017)
22. Mendieta, M., Yang, T., Wang, P., Lee, M., Ding, Z., Chen, C.: Local learning matters: Rethinking data heterogeneity in federated learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8397–8406 (2022)
23. Mohri, M., Sivek, G., Suresh, A.T.: Agnostic federated learning. In: International Conference on Machine Learning. pp. 4615–4625. PMLR (2019)
24. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning. In: NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011 (2011), http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf
25. Shapley, L.S., et al.: A value for n-person games (1953)
26. Shu, J., Zhang, W., Zhou, Y., Cheng, Z., Yang, L.T.: Flas: Computation and communication efficient federated learning via adaptive sampling. IEEE transactions on network science and engineering **9**(4), 2003–2014 (2021)
27. Soltani, B., Zhou, Y., Haghighi, V., Lui, J.: A survey of federated evaluation in federated learning. arXiv preprint arXiv:2305.08070 (2023)
28. Song, T., Tong, Y., Wei, S.: Profit allocation for federated learning. In: 2019 IEEE International Conference on Big Data (Big Data). pp. 2577–2586. IEEE (2019)
29. Wang, G., Dang, C.X., Zhou, Z.: Measure contribution of participants in federated learning. In: 2019 IEEE international conference on big data (Big Data). pp. 2597–2604. IEEE (2019)
30. Wang, S., Ji, M.: A unified analysis of federated learning with arbitrary client participation. arXiv preprint arXiv:2205.13648 (2022)
31. Wang, Z., Fan, X., Qi, J., Jin, H., Yang, P., Shen, S., Wang, C.: Fedgs: Federated graph-based sampling with arbitrary client availability. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 10271–10278 (2023)
32. Yan, Y., Niu, C., Ding, Y., Zheng, Z., Tang, S., Li, Q., Wu, F., Lyu, C., Feng, Y., Chen, G.: Federated optimization under intermittent client availability. INFORMS Journal on Computing **36**(1), 185–202 (2024)
33. Zhou, P., Xu, H., Lee, L.H., Fang, P., Hui, P.: Are you left out? an efficient and fair federated learning for personalized profiles on wearable devices of inferior networking conditions. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies **6**(2), 1–25 (2022)

34. Zhu, L., Lin, H., Lu, Y., Lin, Y., Han, S.: Delayed gradient averaging: Tolerate the communication latency for federated learning. Advances in Neural Information Processing Systems **34**, 29995–30007 (2021)