FedSTaS: Client Stratification and Data Level Sampling for Efficient Federated Learning

Jordan Slessor[1], Dezheng Kong[1], Rui Hu[2], Linglong Kong[†1], and Bei Jiang[1]

[1] *University of Alberta at Edmonton,* [2] *MacEwan University at Edmonton*

Abstract: Federated learning (FL) enables collaborative model training across multiple clients while preserving data privacy. However, existing FL approaches suffer from inefficient client selection and data sampling, leading to slow convergence, high communication costs, and reduced model accuracy - especially in heterogeneous, non-IID settings. We introduce *FedSTaS*, a novel federated sampling framework that strategically integrates client stratification from *FedSTS* with centralized data sampling from *FedSampling* to enhance training efficiency while ensuring privacy. Our method provides theoretical guarantees on unbiased client and data sampling, variance reduction, and improved convergence rates, all while achieving strong privacy protection via differential privacy techniques. Extensive experiments on IID and non-IID datasets show that *FedSTaS* accelerates training convergence and improves final model accuracy, outperforming *FedSTS* and setting a new standard for efficient and privacy-preserving FL.

[†]Corresponding Author. Email: lkong@ualberta.ca

Key words and phrases: Client sampling, data sampling, differential privacy, federated learning, online learning.

1. Introduction

Federated Learning (FL) allows multiple clients to collaboratively train a global model while keeping their local data private (Kairouz et al., 2021). Rather than exchanging raw data, FL relies on clients sharing model updates, which are aggregated by a central server over multiple training rounds (McMahan et al., 2023). While this approach has significant privacy benefits, it also introduces several fundamental challenges that impact the efficiency and effectiveness of training.

One of the biggest challenges in FL is client heterogeneity. Clients often have non-IID data distributions, varying computational resources, and inconsistent participation rates, all of which make it difficult to train a well-generalized model (Nishio and Yonetani, 2019; Zhang et al., 2023). This heterogeneity can lead to biased model updates, slower convergence, and degraded performance (Qi et al., 2023). Another key limitation is communication efficiency - since only a subset of clients can participate in each round, how those clients are selected has a major impact on training speed and final accuracy (Abebe et al., 2024; Zhang et al., 2023). Without effective selection strategies, redundant updates and poorly representative training can significantly slow progress (Yadav and Bor-Yaliniz, 2024; Jiménez et al., 2024).

To improve efficiency, client selection methods have been proposed to reduce redundancy and ensure better representation of the overall data distribution. Traditional approaches such as *FedAvg* (McMahan et al., 2023), rely on random client sampling, which can exacerbate bias and inefficiencies (Fraboni et al., 2021; Cha and Chang, 2024). More recent work, such as *FedSTS*, improves client selection by using stratified sampling based on gradient similarity (Gao et al., 2025), ensuring that selected clients contribute more representative updates. Additionally, *FedCS* prioritizes clients based on their resource availability, further improving convergence speed (Nishio and Yonetani, 2019).

While client selection optimizes who participates, data sampling strategies determine how much and which data is used for training. Many FL methods allow clients to independently sample from their local data, which can introduce sampling bias and inefficiencies (Lu et al., 2022). *FedSampling* addresses this by incorporating a centralized data sampling mechanism to better approximate a global data distribution (Qi et al., 2023). However, existing approaches treat client selection and data sampling as independent problems, missing the opportunity to jointly optimize both.

FL also raises serious privacy concerns. Model updates can be exploited through inference attacks, allowing adversaries to reconstruct private training data (Dwork, 2006; Qammar et al., 2023). Differential privacy (DP) provides a formal way to limit these risks, but excessive noise injection can harm model accuracy (Li et al., 2020). Balancing privacy, efficiency, and accuracy remains an open challenge (Zhang et al., 2023).

In this work, we propose *FedSTaS*, a novel client and data sampling framework that builds on *FedSTS* and *FedSampling* to improve both training efficiency and privacy. Unlike prior work, *FedSTaS* jointly optimizes client selection and data sampling, addressing three key challenges in FL:

1. **More efficient client selection:** Using stratification, *FedSTaS* selects representative clients while reducing bias (Gao et al., 2025).

2. **Smarter data sampling:** Instead of relying on clients to sample their own data, *FedSTaS* incorporates centralized data selection for more efficient training (Zhang et al., 2023; Jiménez et al., 2024; Qi et al., 2023).

3. **Privacy protection:** *FedSTaS* integrates a differentially private sampling mechanism, maintaining privacy while limiting accuracy loss

(Dwork, 2006; Qammar et al., 2023).

We provide theoretical guarantees on unbiased client and data sampling, variance reduction, and improved convergence rates Yadav and Bor-Yaliniz (2024). Our experiments on both IID and non-IID datasets show that *FedSTaS* achieves faster convergence and higher final model accuracy compared to *FedSTS*, particularly in settings with high client heterogeneity.

2. Model Aggregation

Federated learning (FL) aims to train a global model by aggregating updates from multiple distributed clients while keeping their data decentralized. Given N clients, the optimization objective is to minimize a weighted sum of local loss functions:

$$\begin{aligned}\min_{\boldsymbol{w}\in\mathbb{R}^d} F(\boldsymbol{w}) &= \min_{\boldsymbol{w}\in\mathbb{R}^d} \sum_{k=1}^{N} \omega_k F_k(\boldsymbol{w}) \\ &= \min_{\boldsymbol{w}\in\mathbb{R}^d} \sum_{k=1}^{N} \omega_k \frac{1}{n_k} \sum_{j=1}^{n_k} f(\boldsymbol{w}; x_{k,j}),\end{aligned} \tag{2.1}$$

where $F_k(\boldsymbol{w})$ represents the local objective function of client k, and ω_k is a weight proportional to the client's data contribution. The standard FL process involves an iterative exchange where the central server distributes the current global model, clients perform local updates, and the server aggregates these updates to refine the model.

During each communication round t, a subset S_t of clients is selected to participate. Each client $k \in S_t$ optimizes its model using E rounds of stochastic gradient descent (SGD):

$$\boldsymbol{w}_{t+1}^k = \boldsymbol{w}_t^k - \eta_t \nabla F_k(\boldsymbol{w}_t, \xi_t^k), \tag{2.2}$$

where η_t is the learning rate and ξ_t^k is a sample of the kth client's training data, $\mathcal{D}_k$. After local updates, clients send their new model parameters back to the server, which aggregates them using a weighted averaging scheme:

$$\boldsymbol{w}_{t+E} = \sum_{k \in S_t} \frac{N}{m} \omega_k \boldsymbol{w}_{t+E}^k. \tag{2.3}$$

Since only a fraction of clients participate in each round, the aggregated update should remain representative of the entire client population. A client sampling scheme is considered unbiased if the expected update is equal to the full population update:

$$\mathbb{E}\left[\sum_{k \in S_t} \frac{N}{m} \omega_k \boldsymbol{w}_{t+E}^k\right] = \sum_{k=1}^{N} \omega_k \boldsymbol{w}_{t+E}^k, \tag{2.4}$$

where the expectation is taken with respect to the client sampling S_t.

To address these limitations, *FedSTaS* enhances client selection by integrating stratified client sampling and centralized data sampling, which are detailed in the following subsections.

3. Client Level Sampling

In traditional FL, clients are often selected randomly or based on availability constraints in each training round. However, random selection can lead to inefficient updates, especially in non-IID settings, where some clients contribute disproportionately more valuable updates than others. To adress this, *FedSTS* introduces a stratified client selection strategy, which improves training efficiency by selecting clients based on gradient similarity.

3.1 Stratified Client Selection

FedSTS groups clients into H non-overlapping strata according to the similarity of their Information Squeezed (IS) gradients Gao et al. (2025). Instead of selecting clients uniformly, this approach prioritizes those with higher gradient norms, ensuring that the most informative updates are incorporated into the global model.

To achieve optimal stratified sampling, *FedSTS* applies Neyman allocation, which minimizes the variance due to client sampling by allocating more sampling weight to strata with higher variance. Specifically, if we sample m clients, the optimal number of selected clients from each stratum is determined as

$$m_h = m \cdot \frac{N_h S_h}{\sum_{h=1}^{H} N_h S_h}, \tag{3.5}$$

where N_h and S_h are the number of clients in and sample variance of the hth stratum. By allocating more clients to high variance strata, *FedSTS* ensures that the selected updates lead to faster and more stable convergence.

3.2 Importance-Weighted Sampling

Beyond stratified sampling, *FedSTS* further improves client selection by incorporating importance-weighted sampling, where the probability of selected a client is proportional to its gradient norm:

$$p_t^k = \frac{||IS(G_t^k)||}{\sum_{k=1}^{N_h} ||IS(G_t^k)||}, k = 1, \ldots, N_h,$$

where $||IS(G_t^k)||$ is the norm of the IS gradient for the kth client. Clients with larger gradient magnitudes are more likely to be sampled, as their updates have a greater impact on the global model.

3.3 Algorithm Implementation

The full *FedSTS* stratified client selection process is detailed in Algorithm 2. The algorithm first clusters clients into strata based on gradient similarity, then allocates sample sizes using Neyman allocation, and finally performs weighted client sampling to select participants for the next training round.

By combining stratified selection and importance-weighted sampling, *FedSTS* significantly reduces the variance in model updates and accelerates convergence, particularly in highly heterogeneous FL settings. The next section introduces data-level sampling, which further optimizes how data is selected for training.

4. Data Level Sampling

In standard FL, each client independently selects a subset of its local dataset for training. However, random local sampling can introduce bias, particularly in non-IID settings, where client datasets may differ significantly in size and distribution. To address this issue, *FedSampling* Qi et al. (2023) introduces a centralized data sampling mechanism, which ensure that selected training data across clients better represents the overall global distribution.

4.1 Centralized Data Sampling

Instead of allowing clients to arbitrarily sample from their own data, *FedSampling* selects data proportional to the total number of samples available across all clients. This ensures that clients with larger datasets do not contribute more training samples, leading to a more balanced global update.

The probability of selecting a data sample from client k is given by:

$$p_k = n^*/n,$$

where n^* and $n = \sum_{k=1}^{N} n_k$ are the desired and total sample size, respectively. Here $n_k = |\mathcal{D}_k|$ is the size of the kth client's training data.

4.2 Privacy-Preserving Data Size Estimation

Since dataset sizes are privacy-sensitive, directly sharing n_k with the central server could expose private client information. To mitigate this, *FedSampling* employs a differentially private mechanism to estimate dataset sizes while preserving privacy. This procedure is outlined in the subsequent section, in which we alter it to facilitate only the participating clients.

5. FedSTaS: Federated Stratification and Sampling

FedSTaS is designed to jointly optimize client selection and data sampling in FL, integrating the strengths of *FedSTS* and *FedSampling* into a single, efficient framework. The core idea behind *FedSTaS* is to ensure that both the clients chosen for training and the data they contribute are representative of the overall distribution, reducing bias, variance, and inefficiencies commonly found in FL settings. The full outline of our method is described in Algorithm 3.

5.1 Client Selection in *FedSTaS*

For client selection, *FedSTaS* follows the stratified client selection approach introduced in *FedSTS* Gao et al. (2025). This procedure is described in section 3 and the full stratified selection process is outline in Algorithm 2.

5.2 Data Sampling in *FedSTaS*

Our data level sampling is closely based on *FedSampling* Qi et al. (2023), but it is adapted to mimic centralized learning across the set of participating clients rather than the entire population. Suppose that in round t, the server aims to sample n^* observations from the m sampled clients. Let n_{hi} denote the training dataset size for the ith client in stratum h, and let the total number of observations across all participating clients be:

$$n = \sum_{h=1}^{H} \sum_{i=1}^{m_h} n_{hi}.$$

Without privacy constraints, we propose uniform sampling across the participating clients, where each sample is selected with probability n^*/n.

To preserve client privacy, we adopt the *FedSampling* Qi et al. (2023) approach, where each client first clips its sample size to ensure no individual dataset is overly large:

$$n_{hi,c} = \min(n_{hi}, M - 1),$$

for some threshold M. A random sample size is then drawn from a uniform Multinomial distribution, $\hat{n}_{h_i} \sim \mathcal{P}(M)$. Next, we introduce a privacy-preserving mechanism by drawing x_{hi} from a Bernoulli distribution with parameter α, generating a protected client data size:

$$r_{h_i} = x_{h_i} n_{h_i,c} + (1 - x_{h_i}) \hat{n}_{h_i}. \tag{5.6}$$

From this, we compute a private estimate of the total participating data size:

$$\tilde{n} = \left(R - \frac{(1-\alpha)Mm}{2} \right) / \alpha, \tag{5.7}$$

where $R = \sum_{h=1}^{H} \sum_{i=1}^{m_h} r_{h_i}$. The final data sampling step is then performed uniformly across participating clients, with probabilities adjusted using the privacy preserving estimate:

$$p_k = \frac{n^*}{\tilde{n}}.$$

5.3 Summary of Our Approach

Unlike conventional FL methods that uniformly sample ξ_t^k from each client's local training data in Equation 2.2, our approach enforces uniform sampling across all participating clients. Additionally, we apply privacy constraints to ensure that client dataset sizes remain private.

6. Theoretical Results

In this section, we establish the theoretical guarantees of our proposed method. Specifically, we analyze the unbiasedness of both client and data level sampling, derive a variance reduction and convergence bound following the results in Gao et al. (2025), present a variance reduction result for the client sampling, and prove the privacy protection for our private sample size estimator.

6.1 Unbiasedness

Since *FedSTaS* adopts the client sampling strategy from *FedSTS* Gao et al. (2025), it preserves unbiased client selection, as established in Lemma 1. This result ensure that, even though only a subset of clients participates in training, the expected aggregated update remains representative of the entire population.

Lemma 1 (Client Level Unbiased-ness)**.** *Let* $\boldsymbol{w}_{t+1}$ *be computed via Algorithm 3. Then*

$$\mathbb{E}_{S_t}[\boldsymbol{w}_{t+1}] = W(\mathcal{K}),$$

where $W(\mathcal{K})$ *represents the model aggregation computed using all clients.*

Next, we establish the asymptotic unbiasedness of our privacy preserving ratio estimator. Specifically, the following lemma states that the mean squared error (MSE) between the privacy preserving ratio and the actual ratio converges to zero as the number of sampled clients increases. The proof follows directly from Qi et al. (2023), with the key adjustment that our analysis focuses on participating clients rather than the entire client set.

Lemma 2. *Let $p = n^*/n$ and $\tilde{p} = n^*/\tilde{n}$ denote the data level sampling probabilities for centralized learning on the sampled clients, and the proposed method, respectively. Then,*

$$\lim_{m \to \infty} \mathbb{E}[(\tilde{p} - p)^2] = 0. \tag{6.8}$$

6.2 Variance Reduction and Convergence Bound

Beyond maintaining unbiased client selection, *FedSTaS* further reduces variance compared to standard sampling approaches such as stratified or simple random sampling. This result follows directly from Theorem 1 in Gao et al. (2025).

Theorem 1 (Client Sampling Variance Reduction)**.** *If the population is large compared to the subset, $(m/N), (m_h/N_h), (1/m_h)$, and $(1/N)$ are negligible, then the selection (or cross-client) variance of different client sam-*

pling schemes satisfies:

$$\mathbb{V}(\mathbf{w}_{sts}) \leq \mathbb{V}(\mathbf{w}_{stradiv}) \leq \mathbb{V}(\mathbf{w}_{strat}) \leq \mathbb{V}(\mathbf{w}_{rand}) \tag{6.9}$$

where $\mathbf{w}_{sts}$*,* $\mathbf{w}_{stradiv}$*,* $\mathbf{w}_{strat}$*, and* $\mathbf{w}_{rand}$ *denote the model updates aggregated from the subset* S_t *that is generated by FedSTS, stratified selection under re-allocation, stratified selection under plain allocation, and simple random selection scheme, respectively.*

This variance result has direct implications for model convergence. Building on this, a convergence bound can be established for *FedSTaS*'s client sampling strategy, leveraging the theoretical framework in Gao et al. (2025), which itself follows the convergence analysis of *FedAvg* McMahan et al. (2023); Li et al. (2020).

To restate the convergence result, we first introduce the following standard assumptions.

Assumption 1 (L-Smooth)**.** For all $\mathbf{v}$ and $\mathbf{w}$, $k = 1, \ldots, N$,

$$F_k(\mathbf{v}) \leq F_k(\mathbf{w}) + (\mathbf{v} - \mathbf{w})^\top \nabla F_k(\mathbf{w}) + \frac{L}{2}\|\mathbf{v} - \mathbf{w}\|_2^2,$$

where $\mathbf{v}$ and $\mathbf{w}$ are different model parameters.

Assumption 2 (Strongly Convex)**.** For all $\mathbf{v}$ and $\mathbf{w}$, $k = 1, \ldots, N$,

$$F_k(\mathbf{v}) \geq F_k(\mathbf{w}) + (\mathbf{v} - \mathbf{w})^\top \nabla F_k(\mathbf{w}) + \frac{\mu}{2}\|\mathbf{v} - \mathbf{w}\|_2^2,$$

where $\mathbf{v}$ and $\mathbf{w}$ are different model parameters.

Assumption 3 (Bounded Variance)**.** Let ξ_k be sampled from the kth device's local data uniformly at random. The variance of stochastic gradients in each device is bounded:

$$\mathbb{E}\|\nabla F_k(\mathbf{w}_k, \xi_k) - \nabla F_k(\mathbf{w}_k)\|^2 \leq \sigma^2, \quad k = 1, \ldots, N.$$

Assumption 4 (Bounded Expectation)**.** The expectation of stochastic gradients in squared norm is bounded by G^2, i.e.,

$$\mathbb{E}\|\nabla F_k(\mathbf{w}_k, \xi_k)\|^2 \leq G^2, \quad \forall k = 1, \ldots, N.$$

Assumption 5 (Stratified Effect Lower Bound)**.** $\forall h \in H$, we have $||\boldsymbol{W}_h - \boldsymbol{W}(\mathcal{K})|| \geq \omega \geq 0$, where $\boldsymbol{W}_h = \sum_{i=1}^{N_h}(\boldsymbol{w}h_i/N_h)$ is the averaged model update of the hth stratum, $\boldsymbol{W}(\mathcal{K}) = \sum_{h=1}^{H}(\boldsymbol{w}h_i/N)$ is the averaged model update of the entire set $\mathcal{K}$.

Assumption 6 (Stratified Effect Lower Bound)**.** $\forall i, j \in H$, we have $|S_i - S_j| \geq \chi \geq 0$ where S_i is the intra-variability of the ith stratum.

We now restate the convergence bound from Gao et al. (2025). Let F^* and F_k^* denote the minimum of F and F_k, respectively, and define

$$\Gamma = F^* - \sum_{k=1}^{N} p_k F_k^*.$$

Then, the convergence bound on the client sampling under *FedSTS*, as established in Gao et al. (2025), is given in Theorem 2. For a comprehensive discussion and detailed derivation of this result, we refer readers to the original work.

Theorem 2 (Convergence Bound of Unbiased Selection Strategy)**.** *Let Assumptions 1-6 hold and $L, \mu, \sigma_k, G, \omega, \chi$ be defined therein. Consider random client sampling methods for FedAvg when sampling m clients. Then the client sampling from FedSTS satisfies*

$$
\begin{aligned}
&E[F(\boldsymbol{w}_T)] - F^* \\
\leq &O\left(\frac{\sum_{k=1}^{N} p_k^2\sigma_k^2 + E^2G^2 + \gamma G^2}{\mu T}\right) + O\left(\frac{L\Gamma}{\mu T}\right) \\
&+ O\left(\frac{E^2G^2}{m\mu T}\right) - O\left(\frac{\omega^2}{m}\right) - O\left(\frac{\chi^2}{m}\right).
\end{aligned}
$$

6.3 Impact of Data Sampling

In this subsection, we analyze the impact of *FedSTaS* on the conditional variance of the gradient updates, as defined in Equation 2.2. Specifically, we compare simple random sampling (SRS) against the non-private version of *FedSTaS* to assess its effect on variance reduction.

We begin by establishing two key results regarding the conditional variance of gradient updates under these two sampling methods. The variance expressions for simple random sampling and *FedSTaS* are given in Lemma

3 and 4, respectively.

Lemma 3 (SRS Gradient Variance). *Let ξ_{SRS} denote a simple random sample from $\mathcal{D}_k$, and let $\boldsymbol{g}_i = \nabla f(\boldsymbol{w}_t, x_{ki})$. Then,*

$$\begin{aligned} V_{\xi_{SRS}}[\boldsymbol{w}_{t+1}|\boldsymbol{w}_t] &= \frac{\eta_t^2}{n_k \bar{n}} \left(1 - \frac{\bar{n}-1}{n_k - 1}\right) \sum_{i=1}^{n_k} \boldsymbol{g}_i \boldsymbol{g}_i^T \\ &+ \frac{\eta_t^2}{n_k \bar{n}} \left(\frac{\bar{n}-1}{n_k-1} - \frac{\bar{n}}{n_k}\right) \left(\sum_{i=1}^{n_k} \boldsymbol{g}_i\right) \left(\sum_{j=1}^{n_k} \boldsymbol{g}_j\right)^T. \end{aligned} \tag{6.10}$$

Lemma 4 (*FedSTaS* Gradient Variance). *Let ξ_{STaS} denote a sample taken from $\mathcal{D}_k$ with sampling probabilities equal to n^*/n, and let $\boldsymbol{g}_i = \nabla f(\boldsymbol{w}_t, x_{ki})$. Then,*

$$\begin{aligned} V_{\xi_{STaS}}[\boldsymbol{w}_{t+1}|\boldsymbol{w}_t] &= \frac{\eta_t^2 n^*}{n \bar{n}} \left(1 - \frac{n^*(\bar{n}-1)}{n - 1}\right) \sum_{i=1}^{n_k} \boldsymbol{g}_i \boldsymbol{g}_i^T \\ &+ \frac{\eta_t^2 n^*}{n \bar{n}} \left(\frac{n^*(\bar{n}-1)}{n-1} - \frac{\bar{n} n^*}{n}\right) \left(\sum_{i=1}^{n_k} \boldsymbol{g}_i\right) \left(\sum_{j=1}^{n_k} \boldsymbol{g}_j\right)^T. \end{aligned} \tag{6.11}$$

Next, we formally establish that *FedSTaS* provides lower gradient variance compared to SRS. Theorem 3 states that the conditional variance of gradient updates under *FedSTaS* is less than or equal to that of SRS in the Loewner ordering. This result highlights that *FedSTaS* leads to more stable and efficient updates at the client level.

Theorem 3 (Data Sampling Variance Reduction). *Let $V_{\xi_{SRS}}[\boldsymbol{w}_{t+1}|\boldsymbol{w}_t]$ and $V_{\xi_{STaS}}[\boldsymbol{w}_{t+1}|\boldsymbol{w}_t]$ denote the gradient variances under simple random sam-*

pling and the FedSTaS sampling scheme, respectively. Then,

$$V_{\xi_{SRS}}[\mathbf{w}_{t+1}|\mathbf{w}_t] \succeq V_{\xi_{STaS}}[\mathbf{w}_{t+1}|\mathbf{w}_t]$$

in the Loewner ordering.

6.4 Privacy Protection

Our final theoretical result concerns the privacy guarantees of *FedSTaS*, specifically regarding the protection of client local sample sizes. This result is derived from a slight modification of Lemma 3.2 in Qi et al. (2023). The key takeaway is that the estimated total sample size $\tilde{n}$, computed using Equation 5.6, satisfies $\epsilon-$local differential privacy (ϵ-LDP).

We begin by formally defining ϵ-LDP.

Definition 1 (ϵ-LDP)**.** A random mechanism $\mathcal{M}$ satisfies ϵ-LDP if and only if for two arbitrary inputs x and x', and any output y in the image of $\mathcal{M}$,

$$\frac{\mathbb{P}(\mathcal{M}(x) = y)}{\mathbb{P}(\mathcal{M}(x') = y)} \leq e^{\epsilon}. \tag{6.12}$$

We now establish that our privacy preserving sample size estimation achieves ϵ-LDP under an appropriately chosen parameter α.

Lemma 5. *Given an arbitrary size threshold M, our privacy preserving* *estimation of the client's local sample sizes achieves ϵ-LDP, when*

$$\alpha = \frac{e^{\epsilon} - 1}{e^{\epsilon} + M - 2}. \tag{6.13}$$

This result ensures that client dataset sizes remain private, preventing adversaries from inferring sensitive information about individual clients based on their local sample sizes.

7. Experiments

PUT EXPERIMENTS HERE

8. Conclusion and Future Work

PUT CONCLUSION AND FUTURE WORK HERE

Supplementary Materials

DESCRIPTION OF SUPPS HERE

Acknowledgements

ACKS HERE

References

Abebe, W., P. Munoz, and A. Jannesari (2024). Lefl: Low entropy client sampling in federated learning.

Cha, N. and L. Chang (2024). Distributed client selection with multi-objective in federated learning assisted internet of vehicles.

Dwork, C. (2006). Differential privacy. In *International Colloquium on Automata, Languages, and Programming*, pp. 1–12. Springer.

Fraboni, Y., R. Vidal, L. Kameni, and M. Lorenzi (2021). Clustered sampling: Low-variance and improved representativity for clients selection in federated learning.

Gao, D., D. Song, G. Shen, X. Cai, L. Yang, G. Liu, X. Li, and Z. Wang (2025). Fedsts: A stratified client selection framework for consistently fast federated learning. *IEEE Transactions on Neural Networks and Learning Systems 36*(5), 9180–9194.

Jiménez, A. C. C., E. C. Kaya, L. Ye, and A. Hashemi (2024). Submodular maximization approaches for equitable client selection in federated learning.

Kairouz, P., H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, and et. al. (2021). Advances and open problems in federated learning.

Li, T., A. K. Sahu, A. Talwalkar, and V. Smith (2020, May). Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine 37*(3), 50–60.

Li, T., A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith (2020). Federated optimization in heterogeneous networks.

Li, X., K. Huang, W. Yang, S. Wang, and Z. Zhang (2020). On the convergence of fedavg on non-iid data.

Lu, C., W. Ma, R. Wang, S. Deng, and Y. Wu (2022, 11). Federated learning based on stratified sampling and regularization. *Complex & Intelligent Systems 9*, 1–19.

McMahan, H. B., E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas (2023). Communication-efficient learning of deep networks from decentralized data.

Nishio, T. and R. Yonetani (2019, May). Client selection for federated learning with heterogeneous resources in mobile edge. In *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*. IEEE.

Qammar, A., A. Naouri, J. Ding, and H. Ning (2023). Blockchain-based optimized client selection and privacy preserved framework for federated learning.

Qi, T., F. Wu, L. Lyu, Y. Huang, and X. Xie (2023). Fedsampling: A better sampling strategy for federated learning.

Yadav, S. and I. Bor-Yaliniz (2024). Bacsa: A bias-aware client selection algorithm for privacy-preserving federated learning in wireless healthcare networks.

Zhang, J., A. Li, M. Tang, J. Sun, X. Chen, F. Zhang, C. Chen, Y. Chen, and H. Li (2023). Fed-cbs: A heterogeneity-aware client sampling mechanism for federated learning via class-imbalance reduction.

Zhang, Y., C. Xu, H. H. Yang, X. Wang, and T. Q. S. Quek (2023). Dpp-based client selection for federated learning with non-iid data.

9. Appendix

In this appendix we present the algorithms referenced in the article.

Algorithm 1: ClientUpdate

Input: Client k

Input: Model parameter $\boldsymbol{w}_t$

Input: Sampling ratio $n^*/\tilde{n}$

Initialize $\boldsymbol{w}_t^k = \boldsymbol{w}_t$;

$\xi_t^k \leftarrow$ a sample of $\lfloor n^*/m \rfloor$ observations from $\mathcal{D}_k$ taken with sampling probabilities $p_{k_j} = n^*/\tilde{n}$, $j = 1, \dots, n_k$;

foreach *epoch* $i = 1, \dots, E$ **do**

$\quad \boldsymbol{w}_{t+i+1}^k = \boldsymbol{w}_{t+i}^k - \eta \nabla F_k(\boldsymbol{w}_{t+i}^k, \xi_t^k)$;

end

Output: $\boldsymbol{w}_{t+1}^k = \boldsymbol{w}_{t+E}^k, X_t, \lambda_t = IS(G_t^k)$

Algorithm 2: ClientStratification

Input: Compressed gradient of all clients in the tth round

$\{X_k^t\}_{k=1}^N$ and corresponding cluster index $\{\lambda_k^t\}_{k=1}^N$

Input: The number of strata H

Initialize Use λ_k^t to restore $\{X_k^t\}_{k=1}^N$ as $\{Z_k^t\}_{k=1}^N$;

Initialize Randomly select H clients as group or stratum centers

$\{\mu_1, \mu_2, \ldots, \mu_H\}$;

Initialize $C_i = \emptyset$ $(1 \leq i \leq H)$;

repeat

  foreach *client* $k \leq N$ **do**

    $\epsilon_k = \arg\min_{i=1,2,\ldots,H} \|Z_k^t - \mu_i\|_2$;

    $C_{\epsilon_k} = C_{\epsilon_k} \cup \{k^{th}$ client with $Z_k^t\}$;

  end

  foreach *stratum* $C_i,\ i \leq H$ **do**

    New center $\mu_i' = \frac{1}{|C_i|} \sum_{Z_k^t \in C_i} Z_k^t$;

    $\mu_i \leftarrow \mu_i'$;

  end

until $\forall\ i = \{1, 2, \ldots, H\}, \mu_i' = \mu_i$;

Output: $\mathcal{G} = \{C_1, C_2, \ldots, C_H\}$

Algorithm 3: FedSTaS

Input: Updates in the tth round of all clients $\{G_t^k\}_{k=1}^N$; Desired client sample size m; desired data sample size n^*

Initialize $\boldsymbol{w}_0$; $S_t = \emptyset$;

foreach *round* $t = 1, 2, \dots$ **do**

  $\mathcal{G} = \text{ClientStratification}(\boldsymbol{X}_t, \lambda_t, H)$;

  foreach *stratum* $h \leq H$ **do**

    $m_h = m \cdot \frac{N_h S_h}{\sum_{h=1}^{H} N_h S_h}$;

    $p_t^k = \frac{||IS(G_t^k)||}{\sum_{k=1}^{N_h} ||IS(G_t^k)||}, k = 1, \dots, N_h$;

    $S_t = S_t \cup m_h$ clients sampled with $\{p_t^k\}_{k=1}^{N_h}$;

  end

  foreach *sampled client* $i \in S_t$ **do**

    compute r_{h_i} using 5.6;

  end

  compute $\tilde{n}$ using 5.7;

  foreach *sampled client* $k \in S_t$ **do**

    $\boldsymbol{w}_{t+1}^k = \text{ClientUpdate}(k, \boldsymbol{w}_t, n^*/\tilde{n})$;

  end

  $\boldsymbol{w}_{t+1} = \frac{1}{N} \sum_{h=1}^{H} N_h \frac{1}{m_h} \sum_{k=1}^{m_h} \boldsymbol{w}_{t+1}^k$

end