A Data-aware Probabilistic Client Sampling Scheme in Streaming Federated Learning

Chao Song[†], Jianfeng Huang[†], Jie Wu[‡] and Li Lu[†]
[†]School of Computer Science and Engineering, University of Electronic Science and Technology of China, China
[‡]Department of Computer and Information Sciences, Temple University, US
Email: {chaosong, luli2009}@uestc.edu.cn, huangjianfeng@std.uestc.edu.cn, jiewu@temple.edu.cn

***Abstract*—In streaming federated learning, where data on each client is received in the form of a data stream, the distribution of data on the clients has a significant impact on the performance of the federated learning model. The continuous influx of streaming data on the clients leads to real-time changes in the local data distribution, which in turn affects the performance of the federated learning model. Furthermore, the heterogeneity in data distribution among clients exacerbates this impact. In this paper, to address these challenges, we propose a Data-aware Probabilistic Client Sampling scheme (DPCS) for selecting appropriate clients to participate in model training in each round of federated learning. DPCS begins with a method for real-time monitoring of local data distributions on the clients. Based on these observations, the central server adopts a probability-based client sampling strategy. Through extensive experimentation, we demonstrate that our client sampling scheme offers higher timeliness and enhances the performance of federated learning compared to traditional methods.**

***Index Terms*—client sampling, data heterogeneity, data imbalance, streaming federated learning**

I. Introduction

Federated Learning (FL), a machine learning approach that has gained significant attention in recent years, is particularly well-suited for handling big data in a distributed and privacy-preserving manner [1]–[3]. Big data analysis of streaming data is driven by the exponential growth of real-time data generated from various sources [4]. Streaming data refers to a continuous flow of data that is generated in real-time and needs to be processed as it arrives. Unlike traditional batch processing, where data is collected and processed in large chunks at regular intervals, streaming data processing requires real-time or near-real-time analysis. Streaming federated learning is an innovative approach that addresses the challenges of privacy, scalability, and real-time processing in a distributed environment [5]. It is an active area of research with potential applications in various domains, including the Internet of Things, smart cities, and personalized services.

In streaming FL, the problems of data heterogeneity and data imbalance present unique challenges that can significantly impact the performance and effectiveness of the learning models. Data heterogeneity refers to the situation where different clients in the federated system have data that is not identically distributed. In the context of streaming data, this means that the data characteristics, such as feature distributions, data scales, and even the underlying data-generating processes, can vary greatly across clients. Data imbalance occurs when the distribution of different classes (or labels) in the data is uneven. In federated learning, this problem is caused by the fact that some clients may have a large number of samples from certain classes, while others may have very few or none at all. This imbalance can result in a model that is biased towards the majority classes and performs poorly on the minority classes.

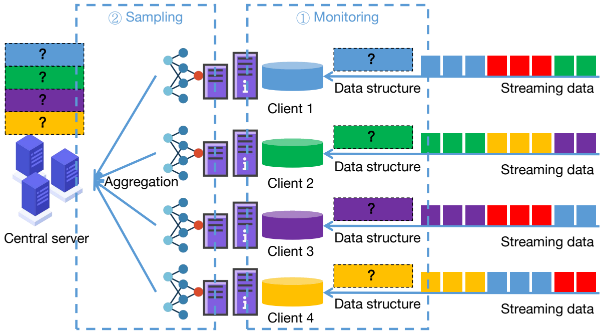
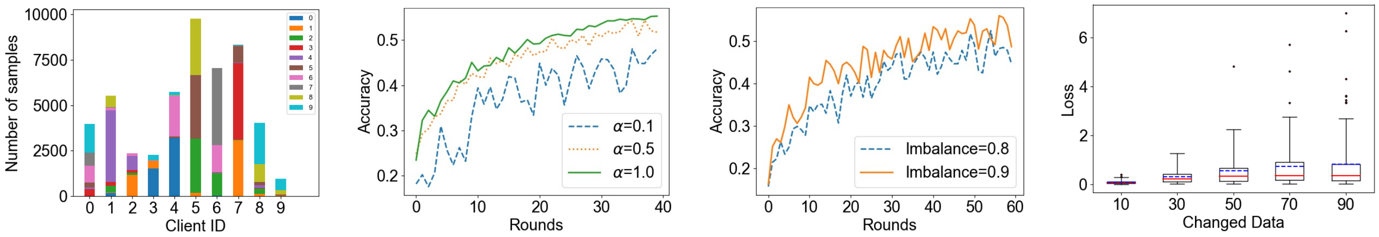


Fig. 1: Local data distribution aware client sampling scheme in streaming federated learning.

To tackle data heterogeneity and imbalance in streaming federated learning, researchers and practitioners employ client sampling and selection strategies, which carefully select a diverse and representative subset of clients for each training round to ensure that the global model benefits from a wide range of data distributions. McMahan et al. in [6] propose an approach randomly selects the subset of clients for training in FL. Yae Jee Cho et al. in [7], [8] discuss the issue of client selection in FL, particularly focusing on strategies that bias the selection towards clients with higher local loss to accelerate convergence speed. To address the challenge of selecting appropriate devices and excluding unnecessary model updates to save resources, Yibo Jin et al. in [9] formulate an online optimization problem and design an algorithm to solve it, achieving resource efficiency and model convergence. The loss is measured for each device and across all devices, and the training aims for convergence of local and global models.

In federated learning, client sampling methods are crucial for the efficiency and effectiveness of the learning process [10], [11]. However, some client sampling methods do not take into account the data imbalance issue. Data imbalance occurs when the data contributed by different clients have



(a) Data distribution ($\alpha = 0.2$)

(b) Heterogeneity

(c) Imbalance

(d) Latency

Fig. 2: Influence of local data distribution in streaming federated learning.

significantly different distributions or quantities. This can lead to a model that is biased towards the majority class or the data from clients with more representation, resulting in poor generalization to underrepresented classes or clients. Furthermore, some other methods design their client sampling strategies based on gradients or the model itself. These approaches assume that the model has already been trained for a certain period and can accurately represent the data. In streaming FL scenarios, the data distribution can change rapidly. If the sampling strategy relies on the model to select clients, it may not be up-to-date with the current data distribution, leading to a significant sampling bias.

In this paper, we propose a Data-aware Probabilistic Client Sampling scheme (DPCS) for data heterogeneity and imbalance in streaming FL. DPCS contains a monitor of local data distribution and a probabilistic client sampling strategy, as shown in Fig. 1. The monitor of local data distribution involves the data structure of counter for the data stream to track the local data distribution at each client. The proposed scheme also includes a theoretical framework for modeling the client sampling problem. An optimization model for client sampling is constructed, which leverages convex optimization techniques to calculate a probability-based client sampling strategy. This ensures that the client selection for model training is not only efficient but also optimized to account for the dynamic nature of streaming data. We conduct extensive experiments on three publicly available datasets, and the experimental results reveal that DPCS offers higher timeliness and enhances the performance of federated learning compared to traditional methods.

II. Streaming Federated Learning

A. Motivation

In order to analyze the impact of data distribution changes in streaming FL, we discuss it from the three perspectives of heterogeneity, imbalance and latency through experimental data analysis. In our experiment focusing on the heterogeneity of client data distribution, we utilized the Dirichlet distribution to construct the local data on the clients. Specifically, we examined data across 10 different class labels and illustrated the distribution of labels on the clients' local data for varying values of the parameter α in the Dirichlet distribution. As the value of α increases, the distribution of data across clients becomes more uniform. This trend is evident when comparing the distributions at different α values. For instance, when α is set to 0.2, the data distribution is non-uniform as shown in Fig. 2a. This suggests that a smaller α value leads to a more diverse distribution of labels, potentially reflecting a more realistic scenario in federated learning where clients may hold significantly different data.

In our experimental investigation of client data distribution heterogeneity in federated learning, as depicted in Fig. 2b, we utilized the random sampling strategy from FedAvg [6] to select clients for each training round. By varying the parameter α in the Dirichlet distribution used to partition the data, we induced different levels of heterogeneity among the clients. The impact of these varying degrees of heterogeneity on the model's accuracy was then assessed. The experimental results clearly demonstrate a relationship between the heterogeneity of client data distribution and the accuracy of the models generated. As the value of α decreases, leading to greater heterogeneity in the data distribution among clients, the accuracy of the models tends to diminish. This suggests that when clients possess more dissimilar data, the federated learning process becomes more challenging, potentially due to the increased difficulty in aggregating diverse updates into a coherent global model.

This imbalance is characterized by a non-uniform distribution of samples across different labels within the dataset, which can affect the model's ability to learn effectively from all classes. The imbalance factor in our experiment is defined as the ratio of data for each label in the sum of all clients' datasets. We construct the number of samples for each label in decreasing fashion with the imbalance factor. For instance, the imbalance factor is set to 0.8, the initial sample count for the first label is set to 10,000. Subsequently, the sample count for the second label is reduced to 8,000, which is 80% of the first label's sample count. This pattern continues with each subsequent label's sample count being reduced by the imbalance factor compared to the previous label's count. In our experimental study on the impact of client data distribution imbalance in federated learning, we employed a random sampling method for the federated learning process. The results of this experiment are presented in Fig. 2c. It is evident from the findings that the accuracy rate for a dataset with an imbalance factor of 0.8 is significantly lower compared to a dataset with an imbalance factor of 0.9. This observation indicates that a higher degree of imbalance in the label set across all clients leads to a degradation in model performance. The FL model may struggle to learn from the underrepresented labels, leading to a biased model that performs poorly on the test data.

To validate the impact of latency in local training on the distribution of local data, we conducted an experimental analysis. In each round of selection, we calculated the loss for the local data before and after local training based on the samples. Fig. 2d displays the difference in loss between the local training before and after each round. In the experiment, we selected one client to calculate the loss, where 'before' indicates the loss before the data update, and 'after' indicates the loss after the data update. The experiment includes variations in the local data distribution established by controlling the rate of change in the client's cache labels, with different growth rates of the data. The results show that the updated data before and after local training have a discrepancy in loss. Therefore, using the loss calculated from the data before local training to formulate a client sampling strategy is inaccurate.

B. Theoretical Analysis

In this section, we analyze the relationship between the model obtained from the federated learning in the m-th round and the model trained using a centralized approach on an uniform-distribution data set. Let $\mathbf{r}_i$ denote the distribution vector of data labels on the i^{th} client by its counter, and is normalized with the C classes, i.e., $\mathbf{r}_i = (r_{i,1}, r_{i,2}, \cdots, r_{i,C})$. $\mathbf{a}$ is the sampling probability of the n client by the central server, i.e., $\mathbf{a} = (a_1, a_2, \cdots, a_n)$. Let w^f_{mT} denote the parameters of global model after m rounds training, and T denotes the number of local updates in each round. w^b_{mT} denotes the parameters of global model trained on a dataset with a target (balance) distribution after m rounds training. w^{center}_{mT} denotes the model parameters of global model trained in a centralized manner according to the expected distribution after m rounds training. The theorem is as follows:

Theorem 1. *There exists an upper bound between the model obtained from the federated learning in the m-th round and the model trained using a centralized approach on a balanced data set, as described below:*

$$\begin{aligned}
&\parallel E[w^f_{mT}] - w^b_{mT} \parallel \\
&\leq \sum\nolimits_{i=1}^{n} a_i[(1+\eta\lambda)^T \parallel w^i_{(m-1)T} - w^{center}_{(m-1)T} \parallel \\
&+ \eta \parallel \mathbf{p}^{center} - r_i \parallel_1 \sum\nolimits_{j=2}^{T} g(w^{center}_{mT-j})(1+\eta\lambda)^{j-1}] \qquad (1)\\
&+ (1+\eta\lambda)^T \parallel w^{center}_{(m-1)T} - w^b_{(m-1)T} \parallel \\
&+ \eta \parallel \mathbf{p}^{center} - \mathbf{p}^{goal} \parallel_1 (\sum\nolimits_{j=1}^{T} (1+\eta\lambda)^{j-1}) g(w^b_{mT-j}),
\end{aligned}$$

where $\mathbf{p}^{center} = (p^{center}_1, p^{center}_2, \cdots, p^{center}_C)$ denotes the distribution vector of the data labels calculated according to such sampling probabilities, which is calculated as follows: $\mathbf{p}^{center} = \sum_{i=1}^{n} a_i \cdot \boldsymbol{r}_i$. Let $\mathbf{p}^{goal}$ denote the distribution vector of the target data labels used for comparison. For training, $\mathbf{p}^{goal}$ is the normalized sum of all client data distributions $\sum_{i=1}^{n} \boldsymbol{r}_i$. For testing, $\mathbf{p}^{goal}$ adopts the uniform distribution.

Proof. Due to the page limitations, we briefly describe the proof process here. We expand the equation into two parts.

$$\begin{aligned}
&\parallel E[w^f_{mT}] - w^b_{mT} \parallel \\
&\leq \parallel E[w^f_{mT}] - w^{center}_{mT} + w^{center}_{mT} - w^b_{mT} \parallel \\
&\leq \parallel E[w^f_{mT}] - w^{center}_{mT} \parallel + \parallel w^{center}_{mT} - w^b_{mT} \parallel .
\end{aligned}$$

For the first part of the equation $\parallel E[w^f_{mT}] - w^{center}_{mT} \parallel$, the upperbound is as follows:

$$\begin{aligned}
&\parallel E[w^f_{mT}] - w^{center}_{mT} \parallel \\
&\leq \sum\nolimits_{i=1}^{n} a_i (1+\eta\lambda) \parallel w^i_{mT-1} - w^{center}_{mT-1} \parallel \\
&\leq \sum\nolimits_{i=1}^{n} a_i[(1+\eta\lambda)^T \parallel w^i_{(m-1)T} - w^{center}_{(m-1)T} \parallel \\
&+ \eta \parallel \mathbf{p}^{center} - r_i \parallel_1 \sum\nolimits_{j=2}^{T} g(w^{center}_{mT-j})(1+\eta\lambda)^{j-1}].
\end{aligned}$$

For the second part of the equation $\parallel w^{center}_{mT} - w^b_{mT} \parallel$, the upper bound is calculated as follows:

$$\begin{aligned}
&\parallel w^{center}_{mT} - w^b_{mT} \parallel \\
&= \parallel w^{center}_{(m-1)T} - \eta \sum\nolimits_{i=1}^{C} p^{center}_i \nabla_w \mathbb{E}_{x|y=i}[-\log f_i(x, w^{center}_{(m-1)T})] \\
&- w^b_{mT-1} + \eta \sum\nolimits_{i=1}^{C} \frac{1}{C} \nabla_w \mathbb{E}_{x|y=i}[-\log f_i(x, w^{center}_{(m-1)T})] \parallel \\
&\leq (1+\eta\lambda)^T \parallel w^{center}_{(m-1)T} - w^b_{(m-1)T} \parallel \\
&+ \eta \parallel \mathbf{p}^{center} - \mathbf{p}^{goal} \parallel_1 (\sum\nolimits_{j=1}^{T} (1+\eta\lambda)^{j-1}) g(w^b_{mT-j}),
\end{aligned}$$

where η is the learning rate. $g(\cdot)$ is a function of w, and $\nabla_w \mathbb{E}_{x|y=i}[-\log f_i(x, w)]$ is $\lambda_{x|y=i}$-Lipschitz for each class $i \in [C]$ introduced in [12].

Thus, we obtain the upper bound of $\parallel E[w^f_{mT}] - w^b_{mT} \parallel$ by combining the two parts of the above equations.

□

We notice that both of the equations $\sum_{i=1}^{n} a_i \parallel \mathbf{p}^{center} - \mathbf{r}_i \parallel$ and $\parallel \mathbf{p}^{center} - \mathbf{p}^{goal} \parallel_1$ are in the upper bound, and $\sum_{i=1}^{n} a_i \parallel \mathbf{p}^{center} - \mathbf{r}_i \parallel$ is related to $\parallel \mathbf{p}^{center} - \mathbf{p}^{goal} \parallel_1$. This can be affected by the sampling probability $\mathbf{a}$, and we investigate the client sampling scheme based on $\mathbf{a}$.

III. Data-aware Probabilistic Client Sampling

To address the dynamic nature of streaming data in federated learning, we propose a Data-aware Probabilistic Client Sampling scheme (DPCS) as shown in Algorithm 1. DPCS is designed to address the challenges of data heterogeneity and dynamic data distributions typical in streaming data scenarios, and the scheme operates through two main components. The first one is the monitor of local data distribution. At the client level, this component continuously monitors and assesses the distribution of the incoming streaming data. It provides real-time insights into the local data characteristics, which is crucial for making informed decisions about client participation in the federated learning process. The second component is the probability model-based client sampling strategy. The central server employs this strategy to determine the probability of each client being selected for participation in each round of federated learning. It does so by analyzing the local data distribution reports submitted by the clients. The probability model evaluates the relevance and diversity of each client's data and calculates the selection probability accordingly, ensuring that the clients chosen contribute to the robustness of the global model. By integrating these components, the DPCS scheme ensures that the client sampling process is both data-aware and probabilistically driven. This leads to a more strategic and efficient selection of clients, which is essential for optimizing the performance of federated learning models, especially when dealing with large-scale and heterogeneous datasets.

A. Monitor of Local Data Distribution

At each client, a data structure is designed to monitor the local data distribution as it changes in real-time, as shown in Fig. 1. The data structure added to each client serves acts as a local monitoring system, which is capable of tracking the frequency and distribution of data instances as they arrive in a streaming fashion. Each client is equipped with a data structure (such as a counter) that can efficiently store and process the incoming data stream to record the frequency [13], [14]. This could be a simple counter with the size of C (the number of classes) to record the frequency of each class. As new data arrives, the data structure updates to reflect the current distribution of data across different categories or features. This monitoring happens in real-time, allowing the system to quickly respond to changes in the data flow. With an up-to-date understanding of the local data distribution, the client sampling strategy based on $\mathbf{a}$ can be adjusted accordingly. This means that the selection of clients for the next round of model training is informed by the most recent data distribution, ensuring that the model is trained on a representative and balanced subset of the data. It's important to note that while this method enhances the adaptability of the FL process, it also needs to maintain the privacy of the clients' data. The data structures should be designed in a way that they do not store raw data but rather aggregate statistics that can be used for sampling purposes without compromising privacy.

B. Probability Model-based Client Sampling Strategy

At the central server, the strategy leverages the power of probabilistic models to make informed decisions about which clients to sample for each round of model training. The core idea is to construct a model that captures the underlying distribution of the data across all clients and uses this information to guide the sampling process.

To minimize the upper bound proven in Theorem 1, we adopt $\parallel \mathbf{p}^{center} - \mathbf{p}^{goal} \parallel_1 = \parallel \sum_{a\in\mathbf{a}} a \cdot \mathbf{r}_i - \mathbf{p}^{goal} \parallel$ as the objective of optimization in this paper as follows:

$$\min_{\mathbf{a}} \quad \parallel \sum\nolimits_{a\in\mathbf{a}} a \cdot \mathbf{r}_i - \mathbf{p}^{goal} \parallel . \tag{2}$$

This is a convex optimisation problem that we can solve using the CVXPY solver library to find the sampling probability for each client a. Based on the sampling probability of each client, the central service uses a sampling method without replacement to select the clients that participate in the current round of federated learning. The sampling strategy is adaptive, meaning it learns from the outcomes of previous rounds. If certain clients consistently provide updates that lead to model improvements, the model may adjust the probabilities to reflect this.

Algorithm 1: Data-aware Probabilistic Client Sampling scheme (DPCS).

```
Input: initial global weight $w_0^f$, learning rate $\eta$,
       number of local updates $T$, number of training
       rounds $R$
Output: trained weights $w_{mT}^f$
for round $m = 0, \cdots, R-1$ do
    Sampling clients, get client S:
    All clients upload $\mathbf{r}_i$;
    Server computes sampling probability;
    Sampling clients according to probability;
for each client $c \in S$, in parallel do
    $w_{mT}^c = w_{mT}^f$;
    for $k = 0, \cdots, T-1$ do
        Compute $\Delta_{mT,k}^c = \nabla F_c(w_{mT,k}^c, \xi_{mT,k}^c)$;
        Local update: $w_{mT,k+1}^c = w_{mT,k}^c - \eta\Delta_{mT,k}^c$;
    Upload to server: $w_{(m+1))T}^c = w_{mT,k+1}^c$;
At server:
Receive $w_{(m+1))T}^c$, $c \in S$;
Let $w_{(m+1)T}^f = 1/|S| \sum w_{(m+1))T}^c$;
```

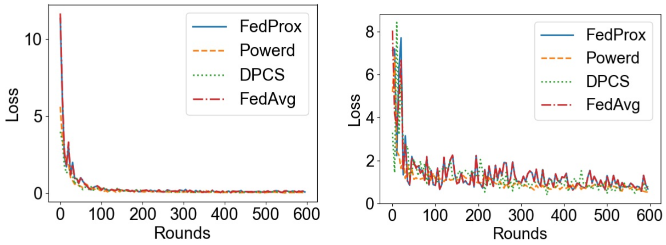
IV. Experiment Results

A. Experiment Setup

1) Parameters: In our experiments, the related parameters are as follows: the number of the clients is 20, and the number of local updates (T) is 5. In local training, the parameters of momentum, batch size and learning rate is 0.5, 64 and 0.03, respectively. The parameter α in the Dirichlet distribution is 0.1. The sampling ratio of clients in each round is 0.3, and the parameter of imbalance is 0.8.

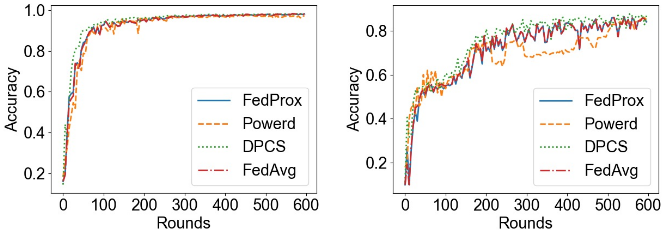
2) Datasets: We adopt three different datasets as follows: (1) **MNIST** [15]: It comprises a total of 70,000 images, with 60,000 images for the training set and 10,000 images for the test set. Each image is a 28 × 28 pixel grayscale representation of a hand-written digit from zero to nine. (2) **Fashion MNIST (FMNIST)** [16]: This is a collection of 70,000 grayscale images, consisting of a training set of 60,000 examples and a testing set of 10,000 examples. Each example is a 28x28 grayscale image, and there are ten classes of labels. (3) **CIFAR10** [17]: This is a widely used dataset in the machine learning and computer vision communities, consisting of 60,000 color images, each with a resolution of 32x32 pixels. The dataset is divided into 50,000 training samples and 10,000 testing samples.

Training and test sets are provided in these datasets. We have done federated learning partitioning and imbalance



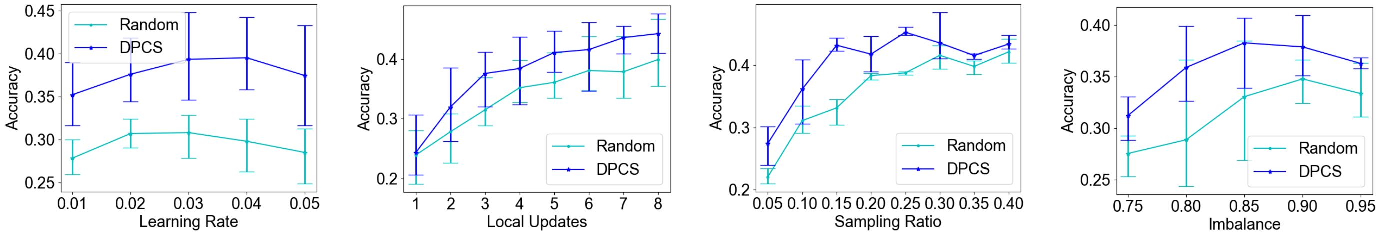
(a) MNIST (b) FMNIST

Fig. 3: Comparison of loss in training.



(a) MNIST (b) FMNIST

Fig. 4: Comparison of accuracy in training.



(a) Learning rate (b) Local updates (c) Sampling ratio (d) Imbalance

Fig. 5: Sensitivity analysis compared with random sampling strategy.

operations on the training set according to the Dirichlet distribution.

3) Baseline Algorithms: The algorithms compared in our experiments are as follows: (1) **FedAvg** [6]: The FedAvg algorithm performs stochastic gradient descent (SGD) locally on each client's data and then aggregates the model updates from all clients to form a global model. In each round, a subset of clients is selected, typically via random sampling. (2) **FedProx** [18]: In the FedProx algorithm, the server aggregates the local updates from the clients by considering both the magnitude of the updates and the proximity of the local models to each other, encouraging consensus among clients. (3) **Powerd** [8]: The algorithm is a client selection method for federated learning that intentionally biases the choice of clients towards those with higher local loss values.

B. Loss and Accuracy in Training

In the experimental study of federated learning training loss, we have conducted a comparative analysis of four distinct algorithms across two different datasets in 600 rounds. As illustrated in Fig. 3, the training loss for all algorithms decreases with an increase in the number of training rounds. Notably, the proposed DPCS demonstrates a superior performance by achieving lower loss values compared to existing algorithms. This innovative strategy not only reduces loss more effectively but also expedites the convergence process during training.

In the experimental evaluation of federated learning on test set accuracy, we have compared and tested four different algorithms across two distinct datasets in 600 rounds. As depicted in Fig. 4, there is a clear upward trend in the training accuracy for all algorithms as the number of training rounds increases. This indicates that as the models are further refined over successive rounds, they become more adept at making accurate predictions on unseen data. The proposed DPCS leverages counter statistics to understand the distribution of data, stands out by achieving higher accuracy rates than the existing algorithms. This strategy demonstrates a more effective utilization of the diverse data present across different clients, leading to a model that generalizes better to new data.

C. Sensitivity Analyses

In our investigation into the parameter sensitivity of client sampling strategies within federated learning, we conducted an experiment comparing the proposed DPCS with the random sampling approach used in FedAvg [6] in 200 rounds with the dataset CIFAR10. The results are presented in Fig. 5. Specifically, Fig. 5a illustrates the impact of the learning rate on the accuracy of both strategies. It is observed that as the learning rate increases, the accuracy of both client sampling strategies initially rises and then declines, with a peak in accuracy around the learning rate of 0.03 to 0.04. This suggests that there exists an optimal range for the learning rate where the models are most effective in terms of accuracy.

Fig. 5b presents a comparison between DPCS and the random sampling strategy in terms of their accuracy as local updates vary. Local updates in federated learning refer to the number of times each client performs gradient descent (or a variant like mini-batch gradient descent) to update the model parameters on their local device. The results indicate a positive trend in accuracy for both strategies with an increase in the number of local updates. The proposed counter-based DPCS strategy consistently demonstrates superior performance over the random sampling strategy.

As depicted in Fig. 5c, we examined the impact of varying sampling ratios on the accuracy of the two client sampling

TABLE I: Comparison of algorithm accuracy

Algorithms	MNIST	FMNIST	CIFAR10
FedAvg [6]	0.9541 / 0.9819	0.7074 / 0.8391	0.3717 / 0.5252
FedProx [18]	0.9532 / 0.9810	0.7083 / 0.8402	0.3743 / 0.5051
Powerd [8]	0.9580 / 0.9704	0.7453 / 0.85	0.3909 / 0.5411
DPCS	**0.9654 / 0.9826**	**0.8074 / 0.8677**	**0.4745 / 0.6127**

strategies. The sampling ratio in federated learning is a critical parameter that defines the proportion of clients selected to participate in each round of the training process relative to the total number of available clients. Expressed as a percentage, the sampling ratio determines how many clients out of the entire network will contribute their local data for the model's training in a given round. The results reveal that the accuracy for both strategies increases with the growth of the sampling ratio, showing a notable upward trend. However, this trend tends to stabilize when the sampling ratio exceeds 0.2. This inflection point suggests that beyond a certain threshold, increasing the sampling ratio does not significantly contribute to further improvements in accuracy.

D. Accuracy of Algorithm

Table I provides a comparative analysis of the accuracy performance of four distinct algorithms, across three different datasets. The accuracy is measured at two training milestones, specifically after 200 and 600 rounds, to evaluate the learning progress and convergence of the algorithms. Each cell in the table represents the accuracy achieved by the respective algorithm on a particular dataset at the specified number of training rounds, denoted as 'a / b' where 'a' corresponds to the accuracy after 200 rounds and 'b' after 600 rounds. On all the three datasets, DPCS has achieved an average improvement of 10.52% over FedAvg, 11.13% over FedProx, and 7.84% overPowerd. This consistent outperformance indicates that the DPCS algorithm is more effective in leveraging the data distributed across the clients in a federated learning setting.

V. Conclusion

In conclusion, this paper has presented an innovative client sampling scheme tailored for streaming federated learning environments, where data is continuously and dynamically received by clients in the form of streams. Recognizing the challenges posed by the real-time fluctuations in local data distributions and the inherent heterogeneity among clients, our proposed solution incorporates a real-time monitoring mechanism to assess the data distribution on each client. This information is then utilized by the central server to implement a probability-based client sampling strategy, ensuring that each training round is informed by the most current and representative data. Our experimental results have validated the effectiveness of this approach, demonstrating that it significantly improves the timeliness and overall performance of federated learning models in the face of streaming data.

Acknowledgments

This work is supported by the National Key R&D Program of China under Grant 2021YFB3101303; the Natural Science Foundation of Sichuan Province of China No.2024NSFSC0492; the National Natural Science Foundation of China under Grant No. 62472070, 82241060, 62020106013.

References

[1] G. Wu, J. Li, Z. Ning, Y. Wang, and B. Li, "Federated learning enabled credit priority task processing for transportation big data," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 1, pp. 839–849, 2024.

[2] P. Kulkarni, A. Kanhere, P. H. Yi, and V. S. Parekh, "Optimizing federated learning for medical image classification on distributed non-iid datasets with partial labels," *CoRR*, vol. abs/2303.06180, 2023.

[3] A. Fu, X. Zhang, N. Xiong, Y. Gao, H. Wang, and J. Zhang, "VFL: A verifiable federated learning with privacy-preserving for big data in industrial iot," *IEEE Trans. Ind. Informatics*, vol. 18, no. 5, pp. 3316–3326, 2022.

[4] D. K. Babu, C. R. Raman, and D. V. D. Rao, "Deep residual network-based data streaming approach for soil type application under iot-based big data environment," *Wirel. Networks*, vol. 29, no. 4, pp. 1751–1769, 2023.

[5] H. Wang, J. Bian, and J. Xu, "On the local cache update rules in streaming federated learning," *IEEE Internet Things J.*, vol. 11, no. 6, pp. 10 808–10 816, 2024.

[6] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of AISTATS*, 2017.

[7] Y. J. Cho, J. Wang, and G. Joshi, "Client selection in federated learning: Convergence analysis and power-of-choice selection strategies," *CoRR*, vol. abs/2010.01243, 2020. [Online]. Available: https://arxiv.org/abs/2010.01243

[8] ——, "Towards understanding biased client selection in federated learning," in *Proceedings of AISTATS*, 2022.

[9] Y. Jin, L. Jiao, Z. Qian, S. Zhang, S. Lu, and X. Wang, "Resource-efficient and convergence-preserving online participant selection in federated learning," *Proceedings of IEEE ICDCS*, 2020.

[10] F. Wu, S. Guo, Z. Qu, S. He, Z. Liu, and J. Gao, "Anchor sampling for federated learning with partial client participation," in *Proceedings of ICML*, 2023.

[11] M. Ribero and H. Vikalo, "Reducing communication in federated learning via efficient client sampling," *Pattern Recognit.*, vol. 148, p. 110122, 2024.

[12] S. Zhang, Z. Li, Q. Chen, W. Zheng, J. Leng, and M. Guo, "Dubhe: Towards data unbiasedness with homomorphic encryption in federated learning client selection," in *Proceedings of ICPP*, 2021.

[13] N. Kakimura and R. Nitta, "Randomized counter-based algorithms for frequency estimation over data streams in O(loglogN) space," *Theor. Comput. Sci.*, vol. 984, p. 114317, 2024.

[14] Q. Shi, Y. Xu, J. Qi, W. Li, T. Yang, Y. Xu, and Y. Wang, "Cuckoo counter: Adaptive structure of counters for accurate frequency and top-k estimation," *IEEE/ACM Trans. Netw.*, vol. 31, no. 4, pp. 1854–1869, 2023.

[15] L. Deng, "The MNIST database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[16] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *CoRR*, vol. abs/1708.07747, 2017. [Online]. Available: http://arxiv.org/abs/1708.07747

[17] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," *Handbook of Systemic Autoimmune Diseases*, vol. 1, no. 4, 2009.

[18] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proceedings of MLSys*, I. S. Dhillon, D. S. Papailiopoulos, and V. Sze, Eds., 2020.