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

Chao Song[1], Jianfeng Huang[1], **Jie Wu**[2] and Li Lu[1]

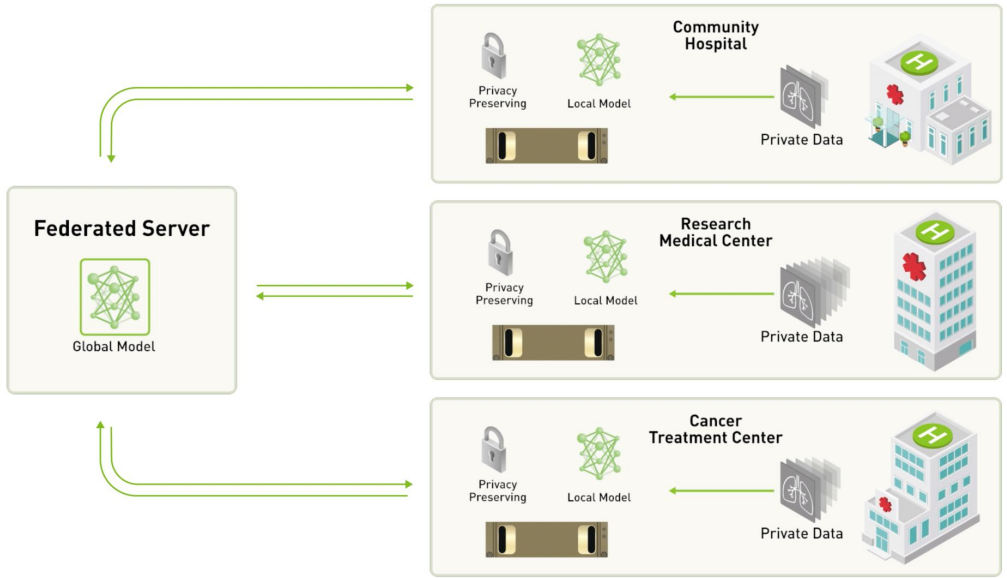
[1]University of Electronic Science and Technology of China, China

[2]Temple University, US

IEEE GLOBECOM®

Research Background

- Federated Learning (FL)
  - A machine learning approach that is distributed and privacy-preserving
  - Suitable for handling big data
- Streaming Federated Learning
  - Deals with real-time data streams
  - Faces challenges of data heterogeneity and imbalance

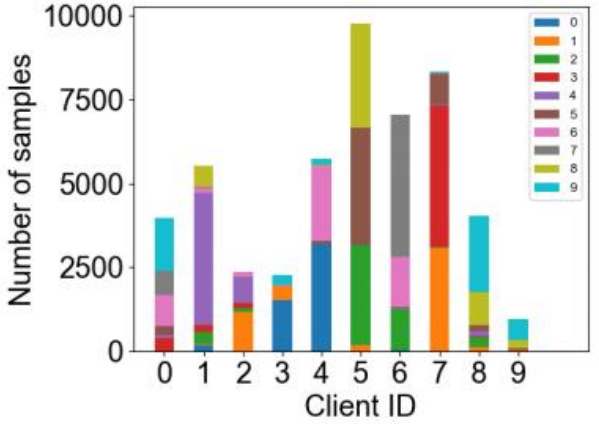


Federated learning

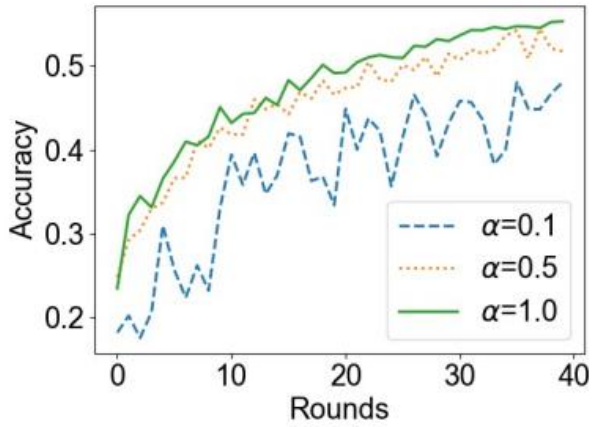
Streaming data

Data Distribution: Heterogeneity

- In FL data, **heterogeneity** means different clients have data with different characteristics.
- Regarding the number of colors and their values, it implies that the distribution of colors and related values differs among clients.
- The impact of client data distribution heterogeneity on the accuracy of models in FL, where the heterogeneity is induced by varying the parameter α in the Dirichlet distribution.



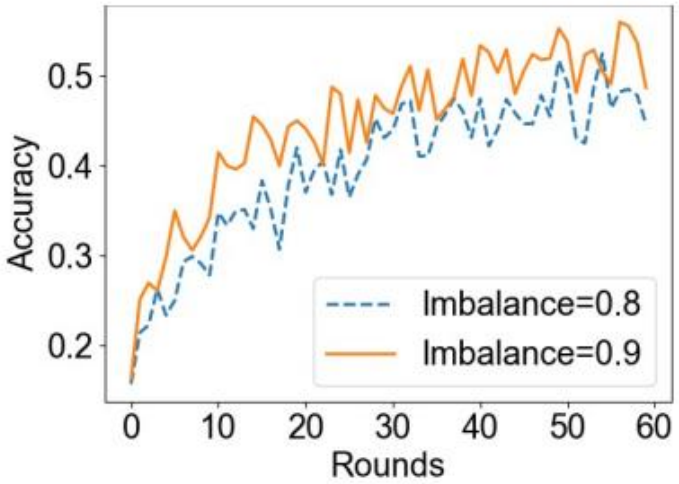
(a) Data distribution ($\alpha = 0.2$)



(b) Heterogeneity

Data Distribution: Imbalance

- **Imbalance**, related to the variance of a distribution over time, shows that the data distribution changes unevenly as time passes.

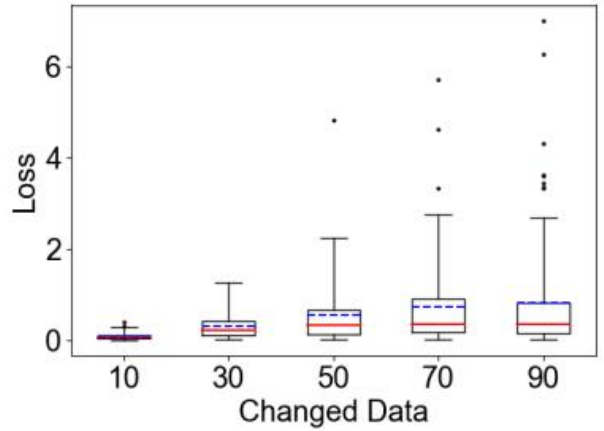


(c) Imbalance

- The impact of client data distribution imbalance on the accuracy of FL. The imbalance is constructed by setting different imbalance factors and observing the resulting accuracy.
- A higher degree of imbalance in the label set across all clients, as indicated by a lower imbalance factor, leads to a degradation in model performance.

Impact of Latency on Local Data Distribution

- To validate latency's impact on local data distribution in training, we calculate loss pre & post local training per round in streaming FL.
- The results show loss difference, and there's a loss discrepancy in updated data. Using pre-training loss for client sampling is inaccurate.



(d) Latency

- **Conclusion**

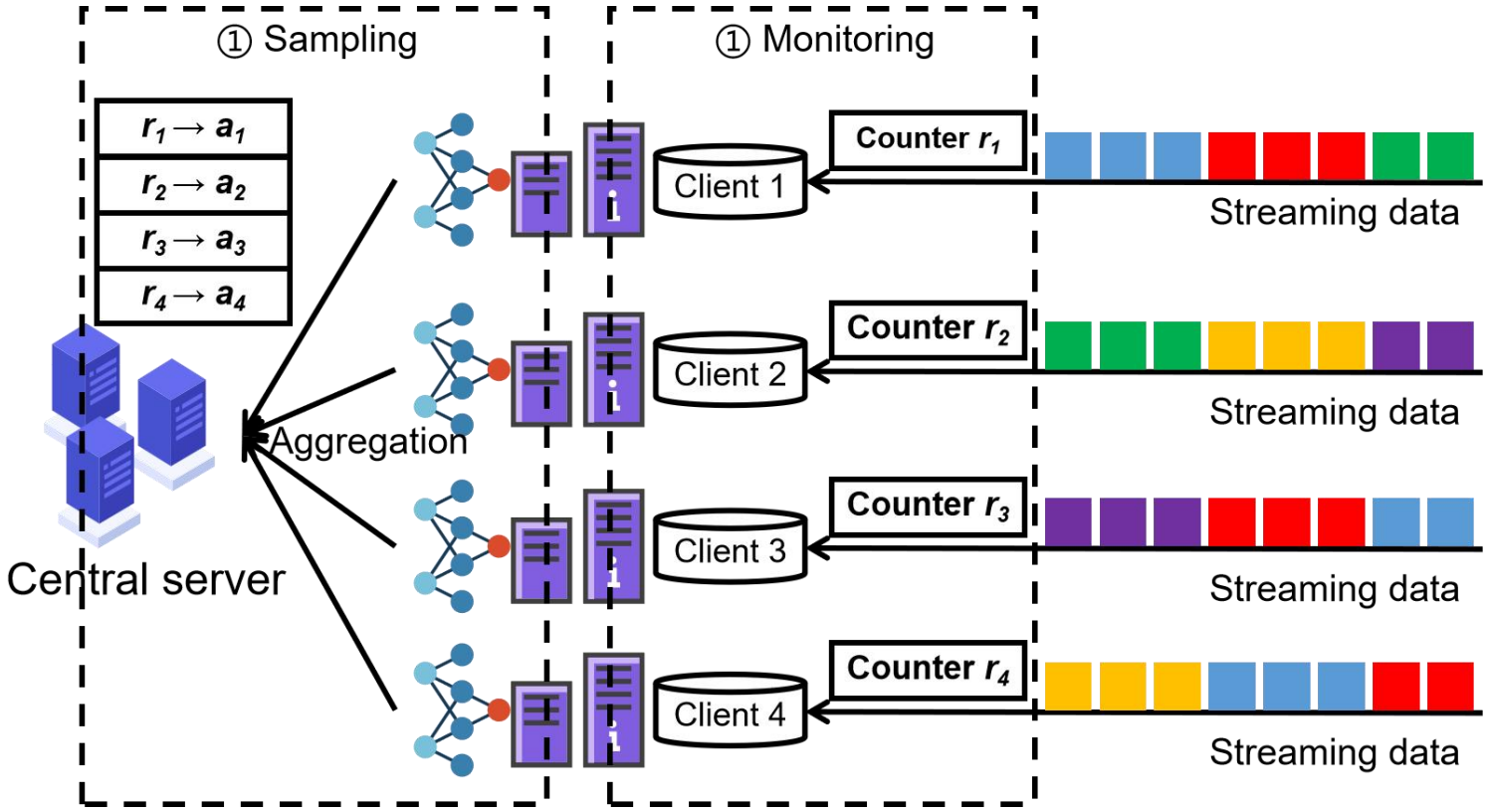
① Client data distribution heterogeneity reduces model accuracy, justifying sampling partial clients in each round.

② Data imbalance impacts model performance, highlighting the need to optimize client sampling for imbalance.

③ Local training changes data distribution, suggesting client sampling should be done before local training in streaming FL.

Research Objectives and Contributions

- We propose a Data-aware Probabilistic Client Sampling (DPCS) for streaming FL:
  - Real-time monitoring of local data distributions on clients
  - A probability-based client sampling strategy
  - Improvement in the timeliness and performance of



Theorem 1 in Streaming Federated Learning

- **Theorem 1**: There exists an upper bound between the model obtained from FL in the m-th round and the model trained using a centralized approach on a balanced data set. The inequality is following

This affects the client sampling in streaming FL.

$$
\begin{aligned}
&\| E[w_{mT}^{f}] - w_{mT}^{b} \| \\
&\leq \sum_{i=1}^{n} a_i [(1+\eta\lambda)^T \| w_{(m-1)T}^{i} - w_{(m-1)T}^{center} \| \\
&+ \eta \| \mathbf{p}^{center} - r_i \|_1 \sum_{j=2}^{T} g(w_{mT-j}^{center})(1+\eta\lambda)^{j-1}] \\
&+ (1+\eta\lambda)^T \| w_{(m-1)T}^{center} - w_{(m-1)T}^{b} \| \\
&+ \eta \| \mathbf{p}^{center} - \mathbf{p}^{goal} \|_1 (\sum_{j=1}^{T} (1+\eta\lambda)^{j-1}) g(w_{mT-j}^{b}).
\end{aligned}
$$

- r_i is the normalized distribution vector of data labels on the i^{th} client.
- a_i is the sampling probability of the i^{th} client.
- w_{mT}^{f} represents the global model parameters after m rounds of training with T local updates each.
- w_{mT}^{b} is the global model trained on a dataset with a target (balance) distribution.
- w_{mT}^{center} is the model trained in a centralized manner according to the expected distribution.
- p^{center} is the calculated distribution vector of data labels based on sampling probabilities $p^{center} = \sum_{i=1}^{n} a_i \cdot r_i$.
- p^{goal} is the distribution vector of the target data labels (sum of all client data distributions for training, uniform distribution for testing).

Overview of the DPCS Scheme

- Monitor of local data distribution： The monitor provides real-time data insights
- Probabilistic client sampling strategy: The sampling strategy calculates selection probabilities based on data distribution reports.

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

Input: initial global weight w_0^f, learning rate η, 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 $w_{(m+1)T}^f = 1/|S| \sum w_{(m+1))T}^c$;

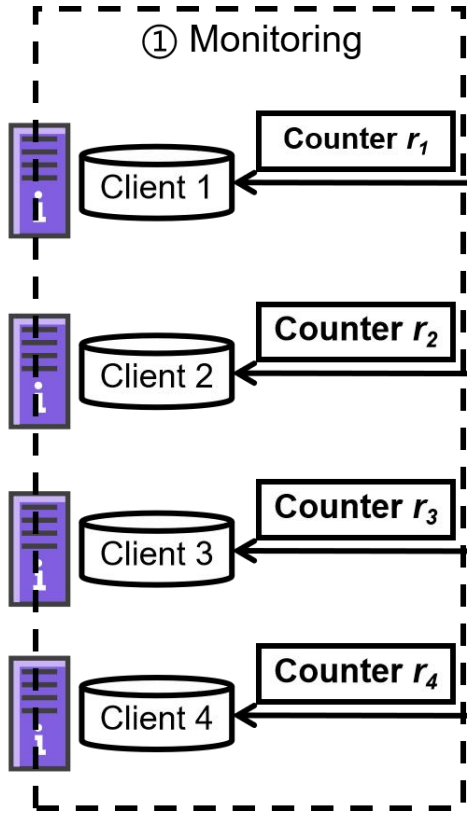
① **Client Sampling and Initialization**: Responsible for sampling clients and initializing the local parameters for the selected clients.

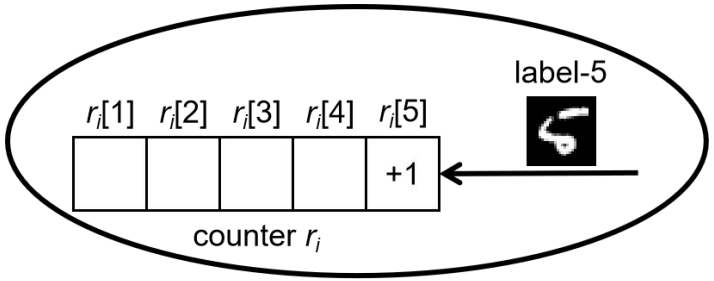
② **Local Training on Selected Clients**: It performs local training on the selected clients in parallel.

③ **Server Aggregation of Model Parameters**: The server aggregates the local model updates received from the selected clients to update the global model parameter.

Monitor of Local Data Distribution

- Data Structure
  - Designs a data structure (such as a counter) at the client side
  - Tracks the frequency and distribution of data instances in real-time
- Privacy Protection
  - Does not store raw data, only stores aggregated statistical information



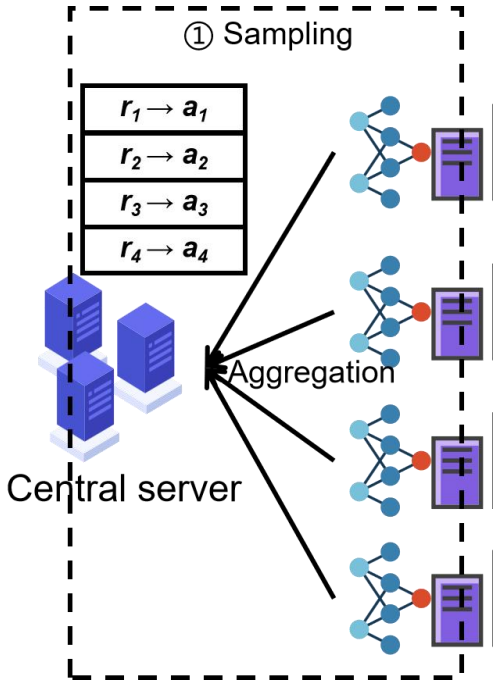


Counters

Probability Model-based Client Sampling Strategy

- Optimization Objective
  - Minimizes the upper bound in Theorem 1
  - Solves a convex optimization problem to get sampling probabilities
- Adaptive Sampling
  - Adjusts probabilities based on previous rounds' results
  - Solve the problem to get sampling probality:

$$\min_{a} \| \ a \cdot r - p^{goal} \|$$



- $a = [a_1, a_2, \cdots, a_n]$: the vector of sampling probabilities of the n clients.
- $r = [r_1, r_2, \cdots, r_n]$: the matrix of the distribution of data labels on the n clients.
- p^{goal}: the distribution vector of the target data labels (sum of all client data distributions for training).

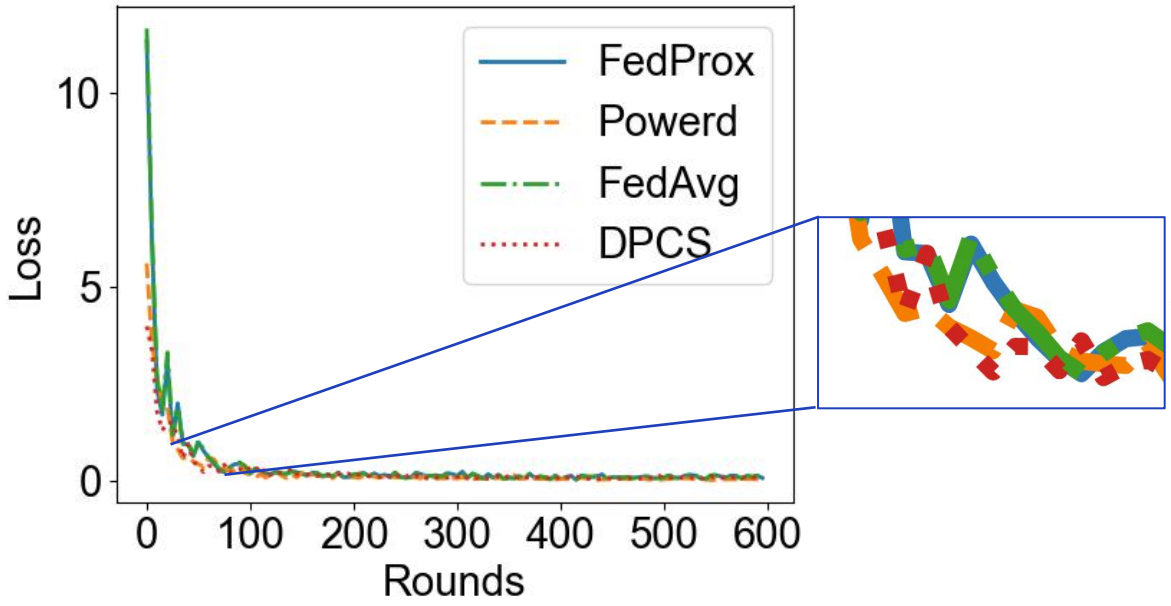
Experimental Setup

- Parameter Settings
  - Number of clients, number of local updates, etc.
  - Momentum, batch size, learning rate, etc.
- Model
  - CNN: The model architecture consists of two convolutional layers followed by two fully connected layers.
- Datasets
  - MNIST, Fashion MNIST (FMNIST), CIFAR10
- Baseline Algorithms

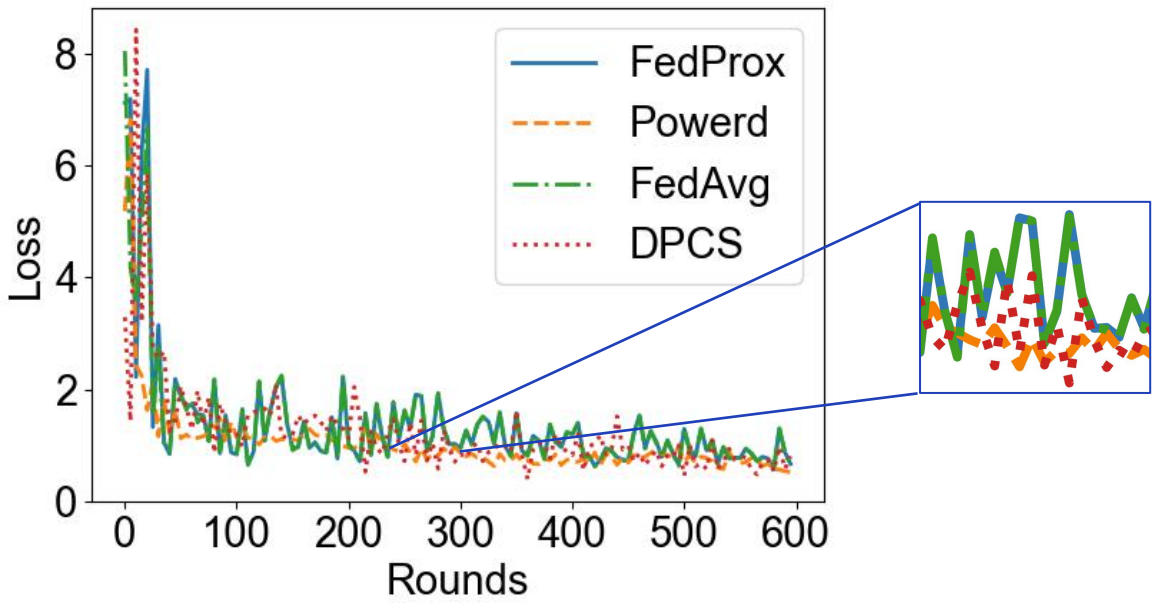
Algorithms	Sampling	Aggregation
FedAvg	Randomly select clients	Average model updates
FedProx	Randomly select clients	Aggregate considering local model proximity
Powerd	Select clients with high loss values	Aggregate based on loss values (favor high loss)

Experimental Results – Training Loss

- Training Loss
  - The training loss of all algorithms decreases as the number of rounds increases.
  - The proposed DPCS algorithm can ensure the convergence of federated learning in training.



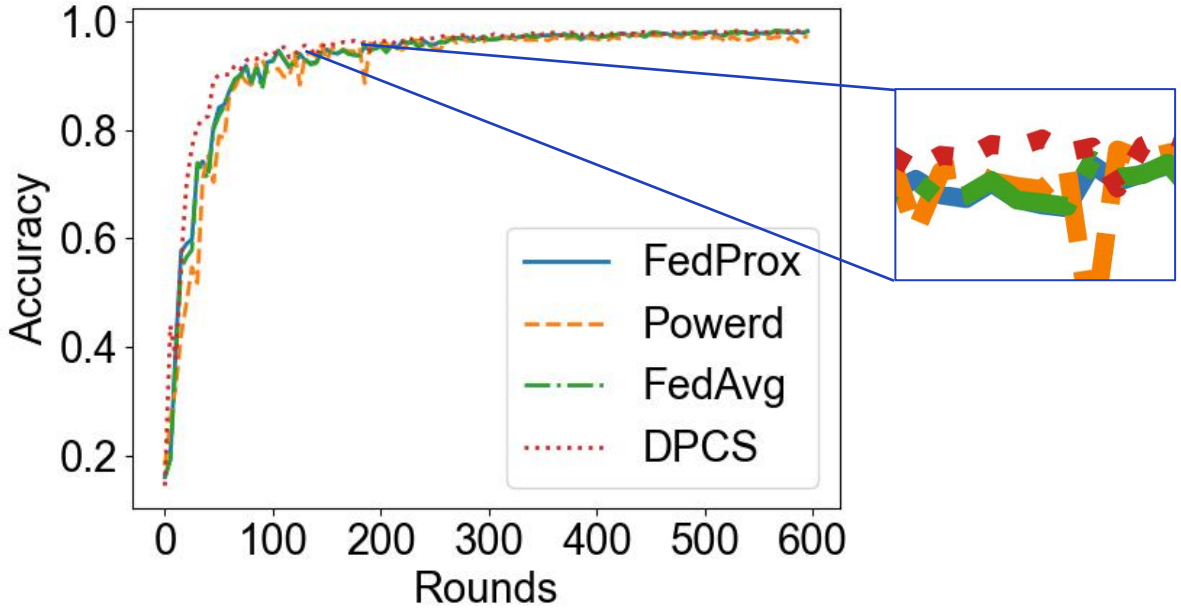
MNIST loss



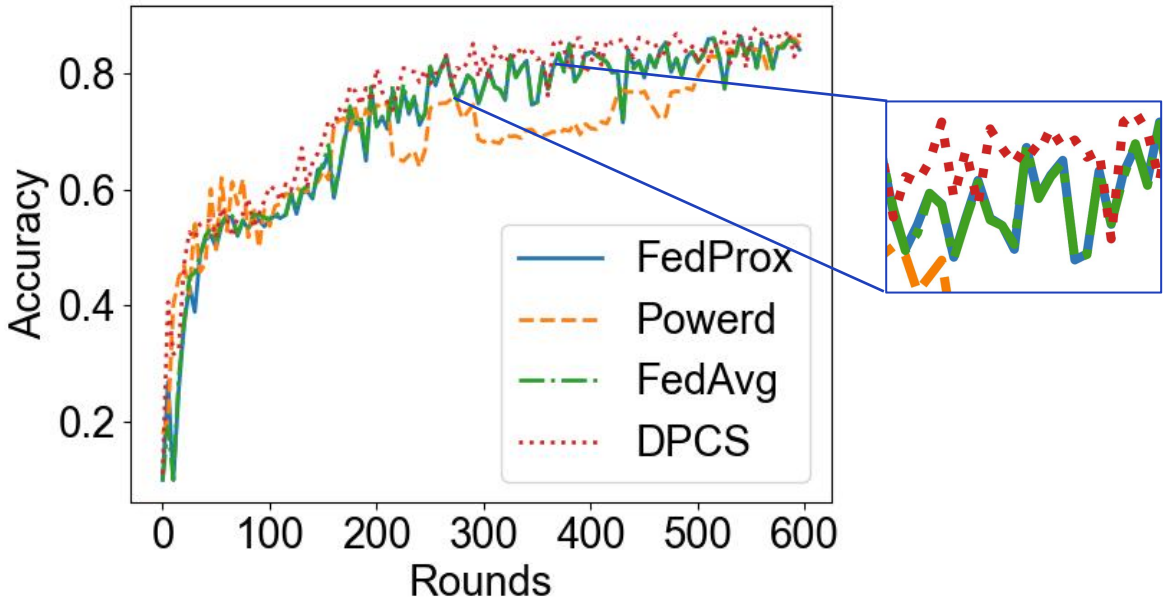
FMNIST loss

Experimental Results - Accuracy

- Accuracy
  - The accuracy of all algorithms increases as the number of rounds increases.
  - During the training process, DPCS can obtain a high-precision model earlier than other algorithms.



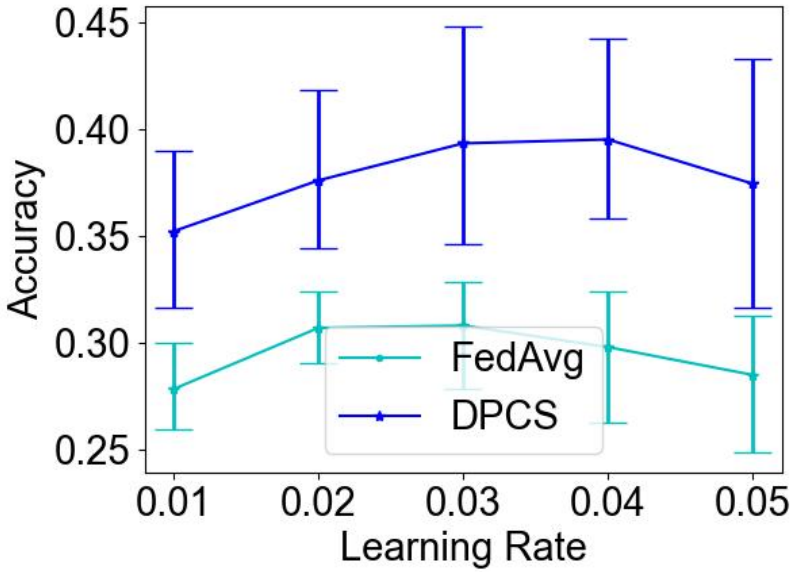
MNIST accuracy

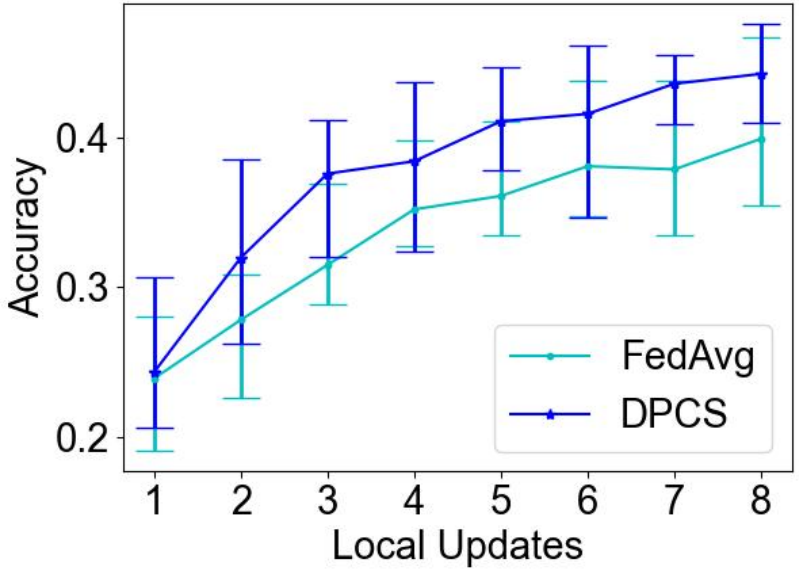


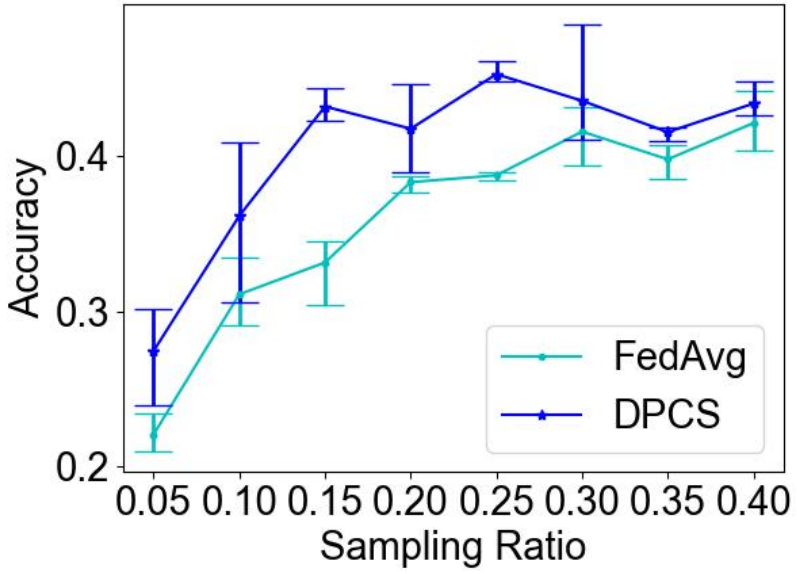
FMNIST accuracy

Experimental Results – Sensitivity Analysis (CIFAR10)

- Learning Rate (η)
  - As the learning rate increases, the accuracy first increases and then decreases.
- Number of Local Updates (T)
  - As the number of local updates increases, the accuracy improves, and DPCS is better.
- Sampling Ratio
  - The proportion of clients selected relative to the total number of available clients.
  - As the sampling ratio increases, the accuracy improves and stabilizes after exceeding 0.2.







Comparison of Algorithm Accuracy

- Comparison on Different Datasets
  - DPCS outperforms other algorithms on three datasets.
  - The average improvement rates are 10.52%, 11.13%, and 7.84% respectively.

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**

'a / b': 'a' corresponds to the accuracy at the 200th round and 'b' at the 600th round.

Conclusions and Future Work

- Conclusions
  - DPCS effectively addresses streaming FL challenges with local data distribution monitoring and probability-based sampling.
  - Experiments on datasets prove DPCS outperforms others in accuracy and loss reduction.
  - Data distribution impact analysis validates importance of handling heterogeneity and imbalance.
- Future Work
  - Explore enhanced sampling strategies adapting to dynamic data distributions.
  - Focus on privacy-preserving and communication optimization in client sampling.

Thank You

- Q&A

- Further questions: Chao Song: chaosong@uestc.edu.cn