Federated Learning with Dynamic Data Distribution and Client Participation

Chris Harry Patrick
University of Wisconsin-Milwaukee
Milwaukee, Wisconsin, USA
patric55@uwm.edu

Tian Zhao*
University of Wisconsin-Milwaukee
Milwaukee, Wisconsin, USA
tzhao@uwm.edu

Abstract

Federated Learning (FL) enables collaborative model training across distributed clients while preserving data privacy. However, real-world deployments continue to face persistent challenges including (1) evolving, non-stationary and non-IID client data; (2) heterogeneous device capabilities and intermittent client availability; and (3) communication overhead from frequent aggregation.

This paper presents an adaptive FL framework that addresses these challenges. We propose an evaluation setup that induces dynamic class rotation to simulate temporal data drift and capability- and performance-aware client participation to reflect realistic device variability. The framework further employs a stagnation-aware aggregation policy that reduces communication overhead without sacrificing accuracy and a rarity-aware, capability-weighted aggregation mechanism for heterogeneous and non-uniform clients. Our framework is evaluated on CIFAR-10 dataset with 20 heterogeneous clients, which achieves a peak accuracy of 87.53%, outperforming FedAvg (82.60%), FedProx (82.74%), and FedDyn (80.92%).

CCS Concepts

• **Computing methodologies** → **Machine learning**; • **Computer systems organization** → *Distributed architectures.*

Keywords

Federated Learning, Dynamic Data, Client Participation, Non-IID, Communication Efficiency

ACM Reference Format:
Chris Harry Patrick and Tian Zhao. 2026. Federated Learning with Dynamic Data Distribution and Client Participation. In *The 41st ACM/SIGAPP Symposium on Applied Computing (SAC '26), March 23–27, 2026, Thessaloniki, Greece.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3748522.3779922

1 Introduction

Federated Learning (FL) is a decentralized learning paradigm in which a central server trains a global model using data that remains distributed across client devices [6]. Each client performs local optimization and periodically uploads model updates, which the server aggregates to update the global model. This approach preserves data privacy and reduces the risks associated with central data storage, making FL attractive for privacy-sensitive domains such as mobile personalization, healthcare, and sensor networks [4, 5].

ACM ISBN 979-8-4007-2294-3/2026/03
https://doi.org/10.1145/3748522.3779922

Despite its appeal, many studies evaluate FL under idealized conditions. Common simplifying assumptions include static client datasets, fixed non-IID partitions, uniform participation probability, and constant aggregation intervals [1, 8]. In practice, FL deployments face challenges such as temporal drift in client data due to evolving user behavior and environments, heterogeneous device capabilities and irregular participation, and costly server–client communication, where either overly frequent or infrequent aggregation can degrade convergence and stability.

In this work, we propose an evaluation framework that closes this gap by extending existing FedAvg-style optimization with a richer simulation layer. Our framework models (1) time-varying data distributions via class rotation, (2) capability- and performance-aware client participation, and (3) stagnation-aware aggregation frequency. By focusing on interpretable, lightweight, and reproducible mechanisms, rather than novel optimization methods, we provide a practical tool for analyzing FL under more realistic deployment conditions while retaining compatibility with standard FL algorithms. Experiments on CIFAR-10 demonstrate improved convergence stability, accuracy, and communication efficiency compared to conventional baselines.

In the rest of the paper, we discuss related work in Section 2, explain our method in Section 3, detail our experiment in Section 4, and analyze the results in Section 5. The code and results of this paper are available at https://github.com/uwm-se/FedStg.

2 Related Work

FL research spans several core challenges, including data heterogeneity, client participation variability, and communication efficiency. Early studies established that data heterogeneity remains one of the most critical challenges in FL. Zhao *et al.* [9] demonstrated that training on non-IID data severely degrades global accuracy when clients hold data from disjoint label sets. McMahan *et al.* [6] introduced FedAvg, the foundational algorithm for FL, which performs local training followed by periodic global aggregation. Although effective under IID conditions, FedAvg suffers convergence instability in non-IID settings because local model updates drift apart. Li *et al.* [5] addressed this issue through FedProx, adding a proximal term to the loss function that regularizes local updates and stabilizes convergence. However, FedProx primarily handles system heterogeneity (differences in compute power and local objectives) rather than temporal data evolution.

Subsequent work expanded the understanding of data heterogeneity. Zhu *et al.* [10] surveyed non-IID partitioning schemes, emphasizing that most benchmarks use static label assignments, which fail to capture realistic user behavior over time. Similarly, Jimenez *et al.* [3] categorized various types of non-IIDness, which include label skew, feature skew, and concept drift, and emphasized the need for standardized benchmarks that reflect data dynamics. Inspired by these observations, our framework introduces a *cosine-based label rotation* mechanism that simulates evolving client data distributions across rounds. This approach models temporal drift in a controllable, reproducible manner, eliminating the need for centralized data exchange or artificial re-sampling.

Beyond data imbalance, system heterogeneity poses additional challenges due to varying device capabilities and unreliable client availability. Nishio and Yonetani [7] proposed *FedCS*, a client scheduling framework that profiles device resources, such as bandwidth, latency, and compute capacity, to select an optimal subset of clients for each round. While *FedCS* improves training efficiency, it relies on centralized profiling and frequent communication with clients, which may be impractical for large-scale or privacy-sensitive deployments. Chen et al. [2] introduced *ACSFed*, which selects statistically diverse clients based on the Earth Mover's Distance (EMD) between local and global distributions. Although this enhances statistical representativeness, it ignores hardware variability and does not adapt client participation frequency over time.

3 Methodology

3.1 Simulation Environment

Our framework enhances realistic FL evaluation by introducing two mechanisms around a standard FL loop. These mechanisms allow the study of FL under conditions closer to real deployments while keeping FedAvg-style optimization unchanged.

Cosine-Based Class Rotation. $p_{i,c}(R)$ is the probability of sampling class c at client i in round R, which simulates the evolving data distributions.

$$\text{phase}_i(R) = \frac{2\pi R}{T} \cdot \text{rotation\_speed} + \text{offset}_i$$
$$w_{i,c}(R) = (\cos(\text{phase}_i(R) + c \cdot \delta) + 1.1)^2$$
$$p_{i,c}(R) = \frac{w_{i,c}(R)}{\sum_{c'} w_{i,c'}(R)}$$

where

- δ is a small phase shift between classes;
- offset_i ensures clients are out-of-phase;
- clients sample 6–8 classes per round for local training.

The purpose of R is to drive time-dependent rotation of class distributions. As the federated round R increases, the cosine-based function shifts sampling probabilities, simulating realistic deterministic data drift.

Client Participation Simulation. $P_i(R)$ is the participation probability that combines capability, performance, and decay:

$$P_i(R) = \min\left(0.95, \max\left(0.3, \text{base} \cdot c_i \cdot \text{performance}_i(R) \cdot \text{decay}(R)\right)\right)$$

- *Capability factor* $c_i \in \{0.8, 0.9, 1.0\}$ is a synthetic metric that combines relative compute speed, network reliability, and likelihood of completing local training.
- *Performance factor* is the mean validation accuracy over the last 3 local epochs.
- *Decay factor* performs linear reduction from 1.0 to 0.5 to model gradual client disengagement.

This produces realistic participation patterns: fast, reliable clients appear more often; stable learners remain engaged; overall availability gradually decreases.

3.2 Aggregation Strategy

The server aggregates client updates using a three–component weighting scheme. For each client i, the weight w_i is defined below.

$$w_i = n_i \cdot s_i \cdot r_i,$$

where n_i is the number of local samples, s_i is the client–speed weight with values 0.8, 0.9, and 1 for slow, medium, and fast client respectively, and r_i is the rarity-aware weight. Speed weight discourages slow/unstable clients from dominating the update. Rarity-aware weight gives more model weights to clients with rare classes and it is calculated for each round as follows.

(1) Compute class rarity score rel_c for class c, where N is the number of clients in this round and f_c of them have class c.
$$rel_c = \frac{N}{f_c}.$$
(2) Compute client rarity score b_i for client i with class set C_i.
$$b_i = \frac{1}{|C_i|} \sum_{c \in C_i} rel_c.$$
(3) Convert to rarity weight r_i, which is a softened and clipped version of the above score.
$$r_i = \min(1.5,\ 1 + 0.25(b_i - 1))$$

Below are the steps of our stagnation-aware aggregation strategy that adapts to the training progress.

(1) *Warm-up (rounds 1–4):* aggregate every round to stabilize initial convergence.
(2) *Reduced frequency (rounds 5+):* aggregate every other round to save communication.
(3) *Fallback:* if validation accuracy stagnates across two aggregation rounds, aggregation returns to every round until improvement resumes.

4 Experimental Setup

The framework is evaluated on CIFAR-10 using a neural network with three convolutional layers followed by two fully connected layers. The dataset is partitioned among 20 simulated clients. Class rotation is driven by a cosine schedule with two to three full cycles over 100 federated rounds, providing sufficient time for the model to experience multiple drift phases.

Client heterogeneity is implemented by assigning each client a capability factor, a local epoch count between three and five, and a batch size between 32 and 64. The participation mechanism uses a base rate $p_0 = 0.8$ and a slow exponential decay that halves the effective rate over the course of training while preserving a non-zero minimum.

FedAvg, FedProx, and FedDyn are used as baselines, all sharing the same backbone architecture, learning rate, local optimization settings, and drift schedule. Each method is trained for 100 communication rounds. Our experiments use the same optimizer as FedAvg but replace the static data and participation assumptions with the temporal drift and adaptive participation. Our aggregation method

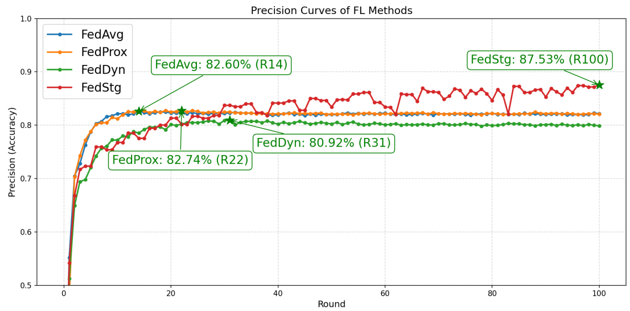

Figure 1: Test accuracy on CIFAR-10 over federated rounds for FedAvg, FedProx, FedDyn, and FedStg (ours) under cosine-based data drift and adaptive participation. Our method attains both higher final accuracy and smoother convergence.

FedStg uses rarity and stagnation-aware strategies described above. The primary metric is test accuracy as a function of rounds.

5 Results and Discussion

Figure 1 shows test accuracy as a function of federated rounds for FedAvg, FedProx, FedDyn, and our method FedStg under the dynamic conditions described above. The baselines converge quickly in early rounds but exhibit oscillations and mild degradation when the class rotation causes shifts in the dominant labels. In contrast, FedStg maintains a upward trajectory and continues to improve as drift cycles progress.

FedStg has 87.53% test accuracy at round 100. Under the same conditions, FedAvg attains 82.60%, FedProx 82.74%, and FedDyn 80.92%. The improvement suggests that adaptive aggregation yields more robust generalization under temporal drift and dynamic participation.

Communication efficiency is also improved. During the reduced-frequency phase, the server aggregates only every second round, and in some configurations every third round, subject to the stagnation criterion. Over 100 rounds, this reduces the number of performed aggregations by approximately 35% compared with methods that aggregate every round.

Table 1: Ablation study results (100 rounds, 20 clients).

Variant (disabled component)	Peak Acc.	Best Round
No dynamic data rotation	88.93%	100
No adaptive participation	85.19%	92
No stagnation-aware aggregation	86.86%	83
Full FedStg (all components)	87.53%	100

We conducted an ablation analysis, as shown in Table 1, that confirms the importance of each component. Lack of dynamic rotation increases accuracy slightly but lack of stagnation-aware aggregation allows early convergence at lower accuracy.

We also evaluated the performance of FedStg on CIFAR-100. The results (Table 2) indicate a much smaller advantage of FedStg over FedAvg. We speculate that the usefulness of rarity-aware weighting is limited since each client observes a small fraction of the classes per round. Our simple model may also limit the attainable accuracy. The cosine-based class rotation is more volatile in a 100-class space.

Table 2: Accuracy and convergence round of different FL methods on CIFAR-100 with 20 clients after 100 rounds

Method	Accuracy (%)	Convergence Round
FedStg	64.78	94
FedAvg	64.48	100
FedDyn	54.86	44
FedProx	57.41	16

6 Conclusion

This paper presented a FL evaluation framework with temporal data drift and dynamic client participation and FedStg, a rarity and stagnation-aware aggregation method. The framework is intentionally simple and does not require changes to the underlying optimization algorithm. Instead, it focuses on making the simulation environment more faithful to real-world deployments. Experiments on CIFAR-10 with 20 heterogeneous clients demonstrate that FedStg improves in accuracy, convergence stability, and communication efficiency compared with FedAvg, FedProx, and FedDyn when all methods are evaluated under the same dynamic conditions.

Acknowledgments

This work is partially supported by the Northwestern Mutual Data Science Institute (NMDSI) under grant number SF17.

References

[1] Durmus Acar, Yue Zhao, Richard Matas, Matthew Mattina, Paul Whatmough, and Venkatesh Saligrama. 2021. Federated Learning Based on Dynamic Regularization. In *International Conference on Learning Representations (ICLR)*.

[2] A. Chen, Y. Fu, Z. Sha, and G. Lu. 2022. An EMD-Based Adaptive Client Selection Algorithm for Federated Learning in Heterogeneous Data Scenarios. *Frontiers in Plant Science* 13 (2022), 908814. doi:10.3389/fpls.2022.908814

[3] Daniel M. Jimenez G., David Solans, Mikko Heikkila, Andrea Vitaletti, Nicolas Kourtellis, Aris Anagnostopoulos, and Ioannis Chatzigiannakis. 2024. Non-IID data in Federated Learning: A Survey with Taxonomy, Metrics, Methods, Frameworks and Future Directions. arXiv:2411.12377 [cs.LG] https://arxiv.org/abs/2411.12377

[4] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. 2020. SCAFFOLD: Stochastic Controlled Averaging for Federated Learning. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*. 5132–5143.

[5] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. Federated Optimization in Heterogeneous Networks. In *Proceedings of Machine Learning and Systems*, Vol. 2. 429–450.

[6] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Artificial Intelligence and Statistics*. PMLR, 1273–1282.

[7] Teruo Nishio and Ryo Yonetani. 2019. Client Selection for Federated Learning with Heterogeneous Resources in Mobile Edge. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, 1–7.

[8] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H. Vincent Poor. 2020. Tackling the Objective Inconsistency Problem in Heterogeneous Federated Optimization. In *Proceedings of the 34th International Conference on Neural Information Processing Systems* (Vancouver, BC, Canada) *(NIPS '20)*. Curran Associates Inc., Red Hook, NY, USA, Article 638, 13 pages.

[9] Yue Zhao, Meng Li, Ruoxi Lai, Naveen Suda, David Civin, and Vikas Chandra. 2018. Federated Learning with Non-IID Data. *arXiv preprint arXiv:1806.00582* (2018).

[10] Hangyu Zhu, Jinjin Xu, Shiqing Liu, and Yaochu Jin. 2021. Federated Learning on Non-IID Data: A Survey. *Neurocomputing* 465 (2021), 371–390.