3FL: Improving Cross-Device Federated Learning Through Straggler Mitigation and Warm Start

Kaylani Bochie[1], Anna Glownia[2], Matteo Sammarco[2], and Miguel Elias M. Campista[1]

[1]*Universidade Federal do Rio de Janeiro – UFRJ*, Brazil
[2]*Stellantis*, Poissy, France
{kaylani,miguel}@gta.ufrj.br, {anna.glownia,matteo.sammarco}@stellantis.com

***Abstract*—This paper proposes four new techniques to increase federated learning performance and decrease its overall training latency. The proposals, called Fastest-First Federated Learning (3FL), Data-Oriented Federated Learning (DOFL), Adaptive Fastest-First Federated Learning (Adaptive-3FL), and Hybrid Federated Learning (Hybrid-FL), reduce training latency by leveraging client selection based on training speed, data volume, and a combination of both, respectively. The proposals' assessments involve simulations using realistic data distributions and client configurations for cross-device federated learning scenarios. The results demonstrate the potential to achieve reductions in training latency of up to 35% compared with traditional federated learning. Moreover, the experiments confirm that the model accuracy reaches similar or even superior results, achieving values of up to 98% for image classification problems.**

***Index Terms*—Federated learning, wireless mobile networks.**

I. Introduction

The exponential growth of edge data has catalyzed deep learning deployment [1], while privacy concerns [2] have driven interest in Federated Learning (FL) [3], [4]. However, cross-device FL suffers from key operational bottlenecks: non-IID client datasets that impair global convergence [5] and straggler devices caused by hardware heterogeneity and unstable networks that severely delay training rounds [6].

To mitigate stragglers and communication overhead, recent research explores client selection, dynamic participation, and model pruning frameworks [7], [8]. Coded learning schemes [9] and invariant dropout methods (e.g., FLuID [10]) attempt to reduce client computation workload, yet they impose additional local encoding complexity or require partial model synchronization. Reputation-based scoring protocols (e.g., CEP [11]) and dynamic participation algorithms (e.g., FLANP [12], ACSP-FL [4]) dynamically scale the active client pool based on historical training delays or loss values. However, they frequently penalize or permanently exclude slower clients, risking severe model divergence under non-IID conditions. Furthermore, while generative frameworks (e.g., FedGCS [13]) and hierarchical edge evaluations [14] optimize client subsets using multi-attribute metrics, they introduce high optimization overhead at the central or edge servers. Crucially, most existing schemes treat compute latency and network transmission in isolation, failing to account for how heterogeneous local dataset volumes jointly impact transport delay and bandwidth consumption in cross-device environments.

Unlike prior work that relies on complex server-side optimization, intrusive local encoding, or permanent client exclusion, this paper introduces four lightweight client selection strategies designed to jointly optimize training latency, dynamic participation, and network bandwidth in non-IID cross-device FL environments. The first one, 3FL (Fastest-First Federated Learning), directly addresses compute stragglers by selecting clients based on individual execution speed without introducing complex server-side overhead. The second one, A-3FL (Adaptive Fastest-First FL) aims to mitigate the risk of model bias seen in static exclusion schemes by dynamically monitoring real-time dropout rates to safely re-select stragglers during ongoing training. DOFL (Data-Oriented FL) addresses the overlooked network transmission bottleneck by leveraging local dataset size heterogeneity to drastically reduce network bandwidth consumption. Finally, Hybrid-FL bridges the gap of isolated metrics by jointly balancing compute latency and data volume into a unified, low-complexity selection rule. Extensive simulations on non-IID benchmark datasets (MNIST and CIFAR-10) demonstrate that our strategies reduce training latency by up to 35% compared to FedAvg, while preserving or enhancing global model accuracy (reaching up to 97%).

In summary, the main contributions of this work are: (i) we formulate a comprehensive client selection taxonomy targeting straggler mitigation and communication efficiency in non-IID cross-device scenarios; (ii) we design and evaluate four complementary client selection mechanisms (3FL, A-3FL, DOFL, and Hybrid-FL) that significantly reduce latency and network footprint without compromising accuracy.

The remainder of this paper is organized as follows: Section II details the proposed algorithms, Section III describes the experimental setup, Section IV presents the performance analysis, and Section V concludes the paper.

This work was supported by CNPq (310234/2025-5, 407304/2025-8, 408255/2023-4, and 405940/2022-0), Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) – Funding Code 001 and project numbers 88887.954253/2024-00, 88887.987121/2024-00, and 88881.987122/2024-01, COFECUB project number Ma1074/25, FAPERJ (E-26/200.438/2026 and E-26/210.778/2025), FAPESP (2023/00673-7 and 2023/00811-0), Fundação de Desenvolvimento da Pesquisa (Fundep) Rota 2030 and our partners Stellantis and Mobway.

II. Proposed Client Selection Schemes

Computational heterogeneity and dynamic network conditions in cross-device FL often turn straggler devices into bottlenecks. Since client subsets exert a disproportionate influence on global convergence and system overhead [15], strategic client selection is key to optimizing latency and bandwidth. This section details four complementary client selection techniques designed to accelerate convergence and reduce network resource consumption. They operate on a two-phase paradigm: an initial *warm start* using an optimized client subset, followed by *fine tuning* that incorporates all available clients to ensure data diversity and mitigate model bias.

A. Fastest-First Federated Learning (3FL)

Fastest-First Federated Learning (3FL) aims to minimize round latency by prioritizing computationally capable and low-latency clients during the early stages of training. Because metadata such as CPU clock rate or network interfaces is rarely collected in standard FL (e.g., FedAvg) due to compatibility and privacy constraints [16], 3FL infers client throughput directly via a lightweight empirical Calibration Phase. In the following, 3FL executes a Warm-Start Phase and a Fine-Tuning Phase, as depicted in Figure 1.

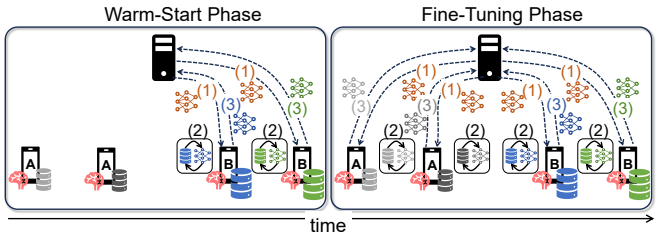


Fig. 1. 3FL: Fast devices ("B") complete the Calibration Phase to build a warm-start model before slower devices ("A") join for full-data fine-tuning.

In the Calibration Phase, even before global training begins, the server broadcasts an untrained probing model, sharing the exact neural network architecture of the target model with all available clients, along with an initial response timeout threshold T_{out}. Clients execute one local training epoch and return their updates. If the proportion of responding clients is below a target threshold τ (e.g., $\tau = 25\%$), the server discards the received updates, doubles the timeout window ($T_{out} \leftarrow 2 \times T_{out}$) until a maximum T_{out}, and repeats the probing round until at least τ clients successfully respond. Assuming that the threshold is achieved, the responsive clients form the initial active subset $\mathcal{C}_{fast}$. In the Warm-Start Phase, for the first K global rounds, training proceeds exclusively among clients in $\mathcal{C}_{fast}$. This rapidly establishes a well-behaved global model without being delayed by stragglers. Finally, in the Fine-Tuning Phase, after K rounds, the server expands participation to all available clients. The fast-trained model serves as a pre-trained "warm start" for resource-constrained devices, preserving data diversity without neglecting straggler datasets. Because calibration lasts for only a single probing round and the total training process spans hundreds of rounds, the time spent in calibration is negligible compared to the cumulative latency savings.

B. Adaptive 3FL (A-3FL)

A key assumption of static 3FL is that device execution speed remains stationary. However, in mobile and wireless environments, client connectivity and processing availability fluctuate due to mobility, battery constraints, or transient network congestion. To address non-stationary behavior, Adaptive 3FL (A-3FL) introduces dynamic recalibration. Training begins identically to 3FL. However, the server continuously monitors client completion rates across rounds. If the proportion of active clients completing their local updates within the calibrated window falls below a dropout threshold δ (e.g., $\delta = 75\%$), the server triggers a new Calibration Phase. Recalibration updates the fast client set $\mathcal{C}_{fast}$ to reflect real-time network conditions, preventing newly emerged stragglers from degrading training speed.

C. Data-Oriented Federated Learning (DOFL)

While 3FL prioritizes computational throughput, it may initially omit clients with large local datasets if they have longer training times. To maximize early information gain, Data-Oriented Federated Learning (DOFL) selects clients based on their local data volume. Thus, instead of calibrating, DOFL executes Metadata Reporting and Volume-Based Selection before it starts training. In Metadata Reporting, at system initialization, clients send a lightweight metadata scalar to the server indicating their total number of local samples N_i. This reporting involves no model training or heavy payload transmission. In Volume-Based Selection, the server ranks clients by dataset size and selects the top fraction τ holding the largest local datasets to perform the initial K warm-start rounds. Afterward, the remaining clients join to fine-tune the model, as shown in Figure 2. By prioritizing data-rich devices in early rounds, DOFL accelerates initial convergence per round while reducing unnecessary network bandwidth consumption from data-scarce devices.

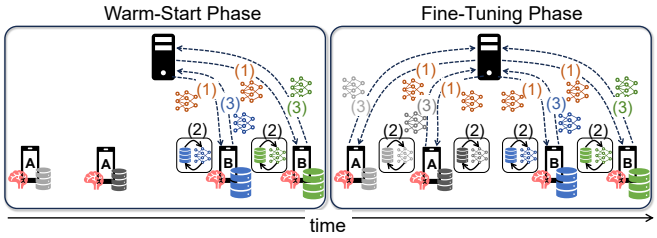


Fig. 2. DOFL scheme: Clients holding larger datasets ("B") are prioritized during early rounds to maximize information density before full federation.

D. Hybrid Federated Learning (Hybrid-FL)

To capture the dual benefits of execution speed (3FL) and information volume (DOFL), Hybrid Federated Learning (Hybrid-FL) unifies processing throughput and dataset size into a single client-selection metric. During initialization, clients undergo both a calibration round (to measure t_i) and

a metadata report (to obtain N_i). The server computes a composite score $S_i = \alpha \cdot \hat{N}_i + (1 - \alpha) \cdot (1 - \hat{t}_i)$ for each client i by normalizing and weighting both attributes, where $\hat{N}_i$ and $\hat{t}_i$ represent min-max normalized dataset sizes and response latencies, respectively, and $\alpha \in [0, 1]$ balances data volume versus speed. The server ranks clients according to S_i and selects the top fraction τ for the initial K training rounds before opening participation to all devices.

III. Experimental Methodology and Setup

This section outlines the experimental environment, dataset partitioning schemes, deep learning model architectures, and simulation parameters used in our evaluations.

A. Hardware Framework and Latency Profiling

To profile hardware heterogeneity in cross-device FL, benchmark experiments were conducted across heterogeneous physical hardware nodes. Local training execution times were empirically measured on systems ranging from low-end edge nodes (Intel Core i5-10400 @ 2.90GHz, 32GB RAM) to high-performance compute servers (AMD Epyc 7452 @ 2.35GHz, 378GB RAM). The empirical execution times obtained from these distinct hardware tiers served as baseline parameters for Monte Carlo latency simulations, allowing realistic profiling of computational speed variance across heterogeneous client pools. Network dynamics and transmission latencies were modeled based on payload sizes (model weights in MB) and empirical link statistics measured via `iPerf3`. The core FL environment was implemented using the Flower framework [17] with TensorFlow/Keras backends.

B. Datasets and Heterogeneous Data Distribution

Evaluations were performed on two standard image classification benchmarks: MNIST and CIFAR-10. MNIST [18] comprises 60,000 training and 10,000 test grayscale images (28×28 pixels) across 10 handwritten digit classes. CIFAR-10 [19] comprises 50,000 training and 10,000 test RGB images ($32 \times 32 \times 3$ pixels) across 10 object classes. Given the classification nature of both tasks, the global model accuracy is used as the primary statistical performance metric.

To reflect realistic cross-device heterogeneity, training samples were partitioned among clients in a non-IID manner, following established practices in the literature [3], [9]. Specifically, local datasets were restricted so that each client had samples from at most two distinct class labels, thereby inducing severe statistical heterogeneity. For MNIST, the total dataset sizes per client were further unbalanced, with no sample overlap, to simulate real-world data distribution skew.

C. Model Architectures and Hyperparameters

Two Convolutional Neural Networks (CNN), tailored to their respective benchmark datasets, were employed. For MNIST, we used a lightweight architecture with 2D convolutional layers, ReLU activations, max-pooling, and a dense softmax output layer. For CIFAR-10, we used a deeper CNN architecture with sequential convolutional blocks, dropout layers for regularization, and a multi-class softmax head.

Prior to evaluating our client selection strategies, we conducted initial sensitivity experiments across candidate client pool sizes (N) to establish a stable baseline FL configuration. A pool of $N = 25$ clients was selected to strike a balanced tradeoff between device heterogeneity and convergence stability. Local training epochs were fixed to $E = 1$ to emphasize per-round convergence dynamics. All hyperparameter configurations are summarized in Table I.

TABLE I
System Evaluation Hyperparameters.

Category	Hyperparameter	Value
FL Framework	Total Client Pool (N)	25
	Global Training Rounds (R)	500
	Local Epochs (E)	1
	Local Batch Size (B_c)	64
	Optimizer & Learning Rate (η)	Adam ($\eta = 10^{-2}$)
Proposed Schemes	Target Selected Fraction (τ)	25%
	Warm-Start Duration (K)	250 rounds
	A-3FL Dropout Threshold (δ)	75%

IV. Experimental Results and Discussion

This section evaluates the proposed client selection strategies (3FL, A-3FL, DOFL, and Hybrid-FL) against standard FedAvg [3]. First, baseline behavior is established across varying network topologies to justify the candidate simulation environment. Subsequently, accuracy trajectories, per-round latency savings, and tradeoff dynamics are analyzed across both MNIST and CIFAR-10 benchmarks.

A. Baseline FedAvg Sensitivity to Non-IID Data Partitioning

To establish a competitive baseline, FedAvg was evaluated under non-IID conditions on MNIST across varying client pool sizes ($N \in \{2, 5, 10, 15, 25, 50\}$). To simulate severe statistical heterogeneity, local datasets were partitioned without overlaps. For $N = 2$, each client held 5 classes; for $N = 5$, 2 classes; for $N = 10$, 1 single class (extreme non-IID); and for $N > 10$, each client received partial data from exactly 2 classes, with a volume scaled inversely to N.

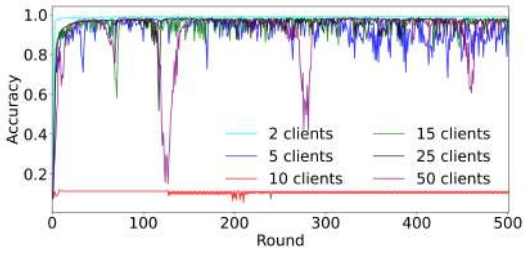


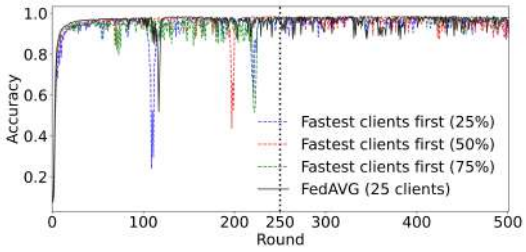
Fig. 3. Baseline FedAvg classification accuracy on MNIST under non-IID partitioning across client pool sizes.

As illustrated in Figure 3, model convergence remains robust across most configurations, demonstrating strong generalization. The extreme 10-client case exhibits significant convergence degradation because single-class partitioning induces localized optimization trajectories that conflict during global

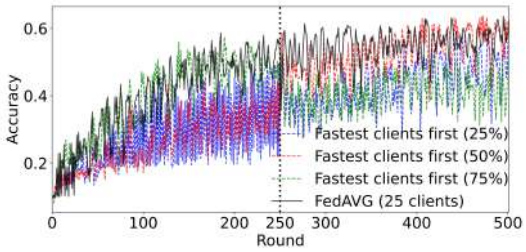
aggregation. In contrast, $N = 50$ demonstrates accuracy drops (e.g., rounds 120, 280, and 460) caused by transient optimization trapping in non-convex local minima under high client fragmentation. The $N = 25$ scenario strikes a balance between cross-device scale, statistical heterogeneity, and convergence stability. Consequently, $N = 25$ was selected as the reference network topology for all subsequent proposal evaluations.

B. Fastest-First FL (3FL) Convergence and Latency Profiling

The performance of 3FL was evaluated by executing the initial $K = 250$ rounds with a fast client subset $\mathcal{C}_{fast}$ (governed by selection fraction $\tau \in \{25\%, 50\%, 75\%\}$), followed by full client participation for the remaining 250 rounds.



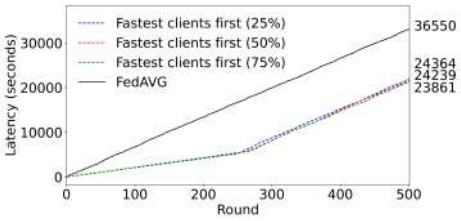
(a) MNIST accuracy trajectory.



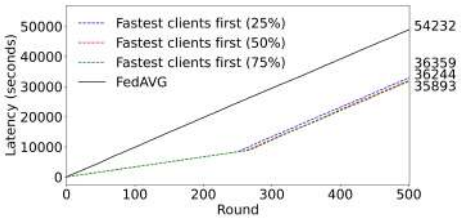
(b) CIFAR-10 accuracy trajectory.

Fig. 4. 3FL classification accuracy trajectories across selection fractions τ compared to baseline FedAvg.

Figures 4(a) and 4(b) show model accuracy over global rounds for MNIST and CIFAR-10, respectively. During the initial warm-start phase ($r \leq 250$), 3FL configurations experience a slight accuracy gap ($\leq 5\%$) relative to the baseline FedAvg due to reduced dataset diversity in $\mathcal{C}_{fast}$. However, once all clients join at round 251, the model rapidly incorporates the missing class distributions, achieving final classification performance equivalent to that of baseline FedAvg across both benchmarks. On CIFAR-10, selecting $\tau = 50\%$ even achieves a slightly superior final accuracy compared to the baseline FedAvg, demonstrating that preconditioning model weights via fast devices reduces early optimization noise. Figure 5 shows that parity in final accuracy is achieved with a considerable reduction in execution latency. By eliminating stragglers during early rounds, we reduce initial-phase training time by up to 62%. Local update times reduce from 61s to 23s on MNIST and from 92s to 33s on CIFAR-10, confirming that 3FL accelerates model warm-starting without sacrificing long-term generalization.



(a) Cumulative latency on MNIST.

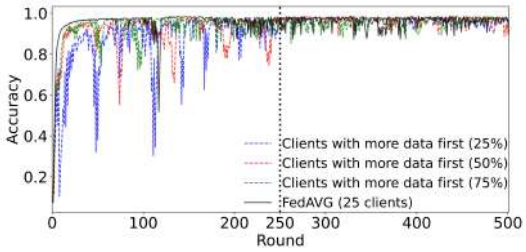


(b) Cumulative latency on CIFAR-10.

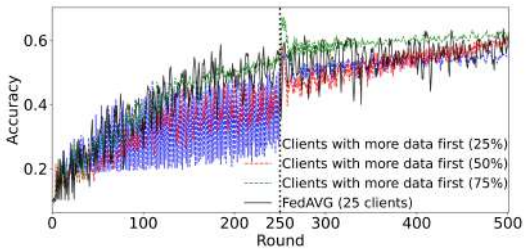
Fig. 5. Cumulative training latency: baseline FedAvg vs. 3FL variants.

C. Data-Oriented FL (DOFL) Evaluation

DOFL prioritizes information density by selecting clients with the largest local datasets during the initial $K = 250$ rounds. Local dataset sizes were varied up to $2\times$ baseline volume to model data volume skew.



(a) MNIST accuracy trajectory.



(b) CIFAR-10 accuracy trajectory.

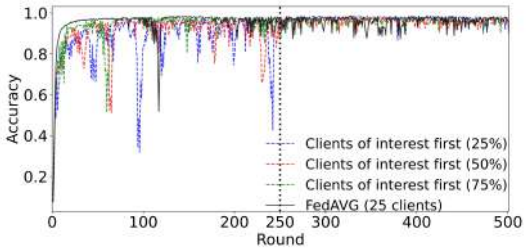
Fig. 6. DOFL classification accuracy trajectories on MNIST and CIFAR-10.

As depicted in Figure 6, DOFL exhibits pronounced accuracy oscillations during the warm-start phase. Because data-rich clients hold concentrated class subsets in non-IID settings, training exclusively on them can induce temporary overfitting to specific local labels. However, upon transitioning to full

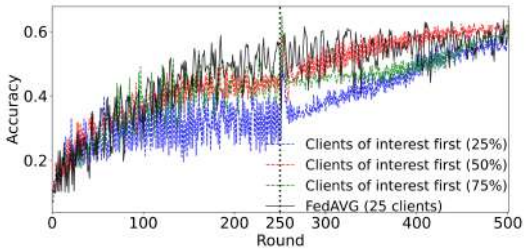
client participation at round 251, accuracy quickly stabilizes, yielding final performance comparable to (MNIST) or marginally higher than (CIFAR-10) standard FedAvg. Because DOFL selects devices based solely on sample volume N_i rather than on execution throughput, its latency savings relative to FedAvg are modest (primarily attributable to fewer active network connections during warm-start rounds). Thus, DOFL serves primarily as an information-maximizing strategy rather than a speed-optimization tool.

D. Hybrid-FL Tradeoff Analysis

Hybrid-FL combines execution speed and sample volume into a composite selection metric S_i. Figures 7(a) and 7(b) illustrate its classification accuracy trajectories.



(a) MNIST accuracy trajectory.



(b) CIFAR-10 accuracy trajectory.

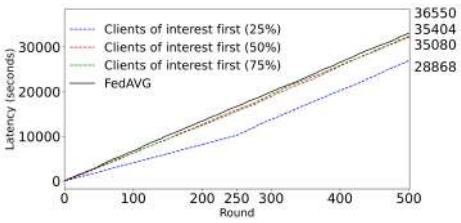
Fig. 7. Hybrid-FL classification accuracy across selection fractions τ.

By jointly weighting speed and dataset size, Hybrid-FL eliminates the warm-start accuracy drops observed in DOFL while achieving superior trajectory stability compared to 3FL. On CIFAR-10 (Figure 7(b)), Hybrid-FL ($\tau = 75\%$) achieves the smoothest convergence profile and reaches baseline performance significantly earlier than FedAvg.

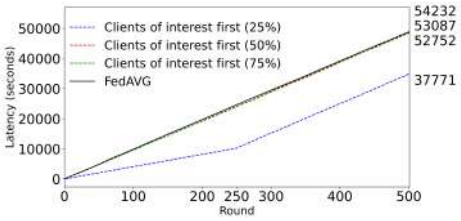
As shown in Figure 8, Hybrid-FL yields latency reductions intermediate between 3FL and FedAvg. Because some fast devices are traded off for slower, data-rich clients, total latency is higher than 3FL but remains substantially lower than FedAvg, establishing Hybrid-FL as a balanced tradeoff scheme.

E. Adaptive 3FL (A-3FL) Dynamics under Channel Variation

To evaluate A-3FL under time-varying conditions, transient stragglers and connection dropouts were simulated by assigning per-round failure probabilities to active devices. Reselection was triggered whenever client availability dropped below $\delta = 75\%$ for three consecutive rounds.



(a) MNIST latency profile.



(b) CIFAR-10 latency profile.

Fig. 8. Hybrid-FL cumulative training latency across selection fractions τ.

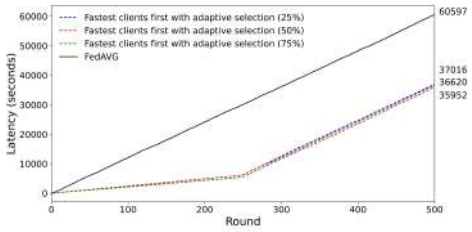


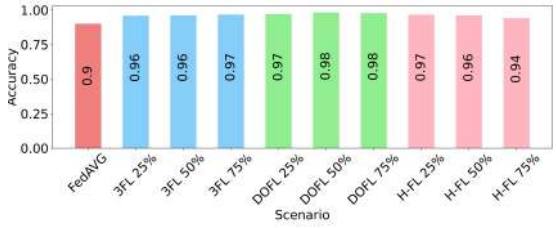
Fig. 9. A-3FL cumulative training latency under time-varying straggler probabilities on MNIST.

Figure 9 demonstrates that static 3FL suffers severe latency degradation when previously fast clients degrade into stragglers. In contrast, A-3FL dynamically detects non-responsive clients and re-executes calibration, recovering optimal training speed. While recalibration introduces a negligible single-round overhead, the long-term latency savings under non-stationary environments outweigh the recalibration cost.

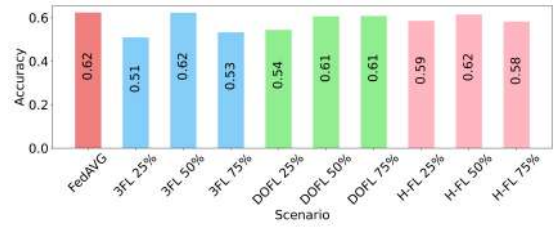
F. Comparative Summary and Tradeoff Analysis

Figures 10 and 11 summarize the final classification accuracies and cumulative latencies across all evaluated schemes and client selection fractions $\tau \in \{25\%, 50\%, 75\%\}$.

Key insights from our evaluation include preserving accuracy, reducing latency, and tuning the selection fraction. Across all variants, initial warm-starting on client subsets does not compromise final model accuracy. After full federation ($r > 250$), all proposed techniques match baseline FedAvg performance within a $\pm 1.2\%$ margin. In addition, 3FL achieves the maximum latency reduction (up to 45% overall runtime savings on CIFAR-10 with $\tau = 25\%$). Hybrid-FL provides a flexible tradeoff, attaining a 28% latency reduction while offering smoother convergence dynamics. Finally, $\tau = 50\%$ offers the most robust tradeoff across both datasets,

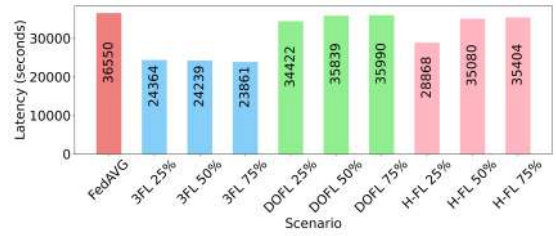


(a) Final accuracy on MNIST.

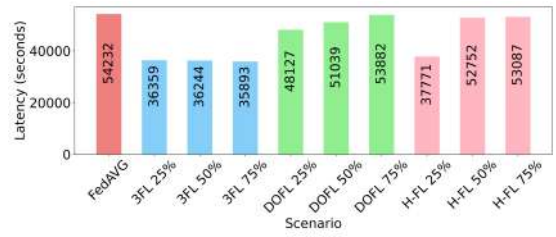


(b) Final accuracy on CIFAR-10.

Fig. 10. Final classification accuracy comparison across all evaluated selection schemes and active client fractions τ.



(a) Cumulative latency on MNIST.



(b) Cumulative latency on CIFAR-10.

Fig. 11. Cumulative execution latency comparison across all evaluated selection schemes and active client fractions τ.

providing substantial latency speedup while maintaining sufficient statistical representation during early training.

V. Conclusion and Future Work

This paper introduced four techniques to reduce resource consumption and latency while maintaining good performance on federated learning tasks, namely, 3FL, A-3FL, DOFL, and Hybrid-FL. The proposed schemes can be selected based on different requirements, such as using 3FL for scenarios where training latency is critical and A-3FL when latency is critical and client performance can fluctuate. Additionally, DOFL has been shown to leverage different data distributions across clients, while Hybrid-FL can be selected as an across-the-board solution. As future work, we intend to expand the simulations to encompass different tasks, namely word prediction and intrusion detection.

References

[1] H. A. Selmy, H. K. Mohamed, and W. Medhat, "Big data analytics deep learning techniques and applications: A survey," *Inf. Syst.*, vol. 120, no. C, Feb. 2024.

[2] A.-T. Tran, T.-D. Luong, and V.-N. Huynh, "A comprehensive survey and taxonomy on privacy-preserving deep learning," *Neurocomput.*, vol. 576, no. C, Apr. 2024.

[3] 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 the 20th International Conference on Artificial Intelligence and Statistics*, vol. 54, Apr. 2017, pp. 1273–1282.

[4] A. M. de Souza, F. Maciel, J. B. da Costa, L. F. Bittencourt, E. Cerqueira, A. A. Loureiro, and L. A. Villas, "Adaptive client selection with personalization for communication efficient federated learning," *Ad Hoc Networks*, vol. 157, p. 103462, Apr. 2024.

[5] Kairouz et al., "Advances and Open Problems in Federated Learning," *Found. Trends Mach. Learn.*, vol. 14, no. 1–2, Jun. 2021.

[6] A. Hard, A. M. Girgis, E. Amid, S. Augenstein, L. McConnaughey, R. Mathews, and R. Anil, "Learning from straggler clients in federated learning," 2024. [Online]. Available: https://arxiv.org/abs/2403.09086

[7] L. Fu, H. Zhang, G. Gao, M. Zhang, and X. Liu, "Client selection in federated learning: Principles, challenges, and opportunities," *IEEE Internet of Things Journal*, vol. 10, no. 24, Dec. 2023.

[8] C. Smestad and J. Li, "A systematic literature review on client selection in federated learning," in *27th International Conference on Evaluation and Assessment in Software Engineering*, Jun. 2023, p. 2–11.

[9] S. Kumar, R. Schlegel, E. Rosnes, and A. G. i. Amat, "Coding for straggler mitigation in federated learning," in *ICC 2022 - IEEE International Conference on Communications*, May 2022, pp. 4962–4967.

[10] I. Wang, P. J. Nair, and D. Mahajan, "FLuID: Mitigating stragglers in federated learning using invariant dropout," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. [Online]. Available: https://openreview.net/forum?id=rG1M3kOVba

[11] M. Asad, S. Otoum, and S. Shaukat, "Resource and heterogeneity-aware clients eligibility protocol in federated learning," in *IEEE Global Communications Conference (GLOBECOM 2022)*, Dec. 2022, pp. 1140–1145.

[12] A. Reisizadeh, I. Tziotis, H. Hassani, A. Mokhtari, and R. Pedarsani, "Straggler-resilient federated learning: Leveraging the interplay between statistical accuracy and system heterogeneity," *IEEE Journal on Selected Areas in Information Theory*, vol. 3, no. 2, pp. 197–205, 2022.

[13] Z. Ning, C. Tian, M. Xiao, W. Fan, P. Wang, L. Li, P. Wang, and Y. Zhou, "FedGCS: A generative framework for efficient client selection in federated learning via gradient-based optimization," in *Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3-9, 2024*, Aug. 2024, pp. 4760–4768.

[14] S. Trindade and N. L. S. da Fonseca, "Client selection in hierarchical federated learning," *IEEE Internet of Things Journal*, vol. 11, no. 17, pp. 28 480–28 495, 2024.

[15] J. Zhao, S. Bagchi, S. Avestimehr, K. Chan, S. Chaterji, D. Dimitriadis, J. Li, N. Li, A. Nourian, and H. Roth, "The federation strikes back: A survey of federated learning privacy attacks, defenses, applications, and policy landscape," *ACM Comput. Surv.*, vol. 57, no. 9, Apr. 2025.

[16] Q. Zhang, P. Palacharla, M. Sekiya, J. Suga, and T. Katagiri, "Demo: A blockchain based protocol for federated learning," in *IEEE 28th International Conference on Network Protocols (ICNP)*, Oct. 2020.

[17] D. J. Beutel, T. Topal, A. Mathur, X. Qiu, J. Fernandez-Marques, Y. Gao, L. Sani, H. L. Kwing, T. Parcollet, P. P. d. Gusmão, and N. D. Lane, "Flower: A friendly federated learning research framework," *arXiv preprint arXiv:2007.14390*, 2020.

[18] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[19] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," University of Toronto, Toronto, Ontario, Tech. Rep. 0, 2009. [Online]. Available: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf