Addressing Data Heterogeneity in Federated Learning: A Comparative Study of FedAvg and FedProx under IID and Non-IID Scenarios

Mr. Santosh Kumar Metta [1], Mrs. A Tulasi [2]

1 Department of CS&SE, AUCE, [Andhra University]

2 Department of CS&SE, AUCE, [Andhra University]

[1]*M.Tech Student, Computer Science and Systems Engineering, Andhra University College of Engineering(A), Andhra University, Visakhapatnam.*

[2]*Assistant Professor, Computer Science and Systems Engineering, Andhra University College of Engineering(A), Andhra University, Visakhapatnam.*

---***---

Abstract: Federated Learning (FL) has emerged as a promising paradigm for privacy-preserving machine learning, where data remains localized on clients while contributing to a shared global model. Among the most widely studied algorithms in this field are Federated Averaging (FedAvg) and Federated Proximal (FedProx). This paper presents a comparative study of FedAvg and FedProx under both Independent and Identically Distributed (IID) and Non-IID data scenarios. We utilize the EMNIST dataset (balanced split, 47 classes) with 40 simulated clients under IID and Dirichlet-based Non-IID partitioning. Our experiments demonstrate that FedAvg performs efficiently in IID settings with fast convergence and competitive accuracy, whereas FedProx, by incorporating a proximal regularizer, provides stability and superior performance in Non-IID environments. Performance is assessed using metrics including accuracy, communication overhead, convergence area-under-curve (AUC), and training time. The results highlight that FedAvg is optimal for homogeneous data distributions, while FedProx is more suitable for real-world heterogeneous federated systems.

Keywords: Federated Learning, FedAvg, FedProx, IID, Non-IID, Data Heterogeneity

1. Introduction

Machine learning models have traditionally relied on centralized data collection, where training data from different sources is aggregated into a single repository for building predictive models. While effective, this approach presents serious drawbacks, including privacy risks, high communication overhead, and compliance challenges in sensitive domains such as healthcare, finance, and mobile computing. For example, transmitting medical images or banking records to centralized servers often violates strict privacy regulations such as HIPAA in healthcare or GDPR in Europe [1]. Moreover, as data volumes increase, central aggregation becomes computationally and financially impractical.

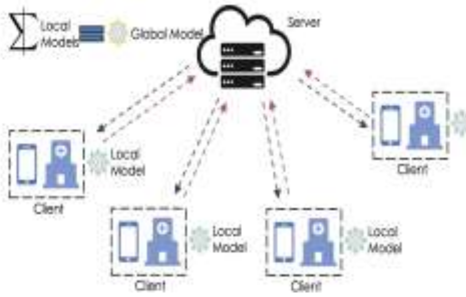

Fig -1: Federated Learning block diagram

To address these challenges, Federated Learning (FL) was introduced by Google in 2016 and later formalized by McMahan et al. (2017) [2]. Unlike traditional centralized approaches, FL allows decentralized clients such as smartphones, IoT devices, or medical institutions to collaboratively train a global model without transmitting raw data. Instead, clients compute local updates which are then aggregated by a central server, thus preserving data privacy while still benefiting from large-scale collaborative learning.

The most widely used algorithm in FL is Federated Averaging (FedAvg), which became the foundational baseline. FedAvg was first introduced in the seminal paper *"Communication-Efficient Learning of Deep Networks from Decentralized Data"* [2]. The algorithm combines local Stochastic Gradient Descent (SGD) updates from participating clients through weighted averaging at the server. FedAvg demonstrated that collaborative training is feasible at scale, even when data is distributed across millions of edge devices. While FedAvg performs remarkably well under Independent and Identically Distributed (IID) conditions, it faces significant challenges under Non-IID data distributions. In such cases, clients' local updates may drift in different directions due to skewed data distributions, a phenomenon known as client drift, leading to poor convergence and suboptimal model quality [3].

To overcome this issue, Federated Proximal (FedProx) was introduced by Li et al. (2018) [4]. FedProx extends FedAvg by adding a proximal term to the client's local optimization objective. This regularization term penalizes updates that diverge significantly from the global model, thereby reducing the effects of client drift. The additional regularization stabilizes convergence and improves robustness in heterogeneous environments, making FedProx one of the first significant improvements over FedAvg specifically designed to tackle data heterogeneity and system differences among clients.

Today, FedAvg and FedProx remain benchmark algorithms in federated learning research. While newer approaches such as SCAFFOLD [5], FedNova [6], and FedOpt [7] have been proposed to further mitigate variance and improve convergence under heterogeneity, FedAvg and FedProx continue to serve as baseline algorithms for comparative studies. They provide a clear understanding of the trade-offs between efficiency and robustness when optimizing under IID versus Non-IID data distributions.

2. Literature Review

The emergence of Federated Learning (FL) has triggered a wave of research into distributed optimization algorithms capable of addressing data and system heterogeneity. This section reviews key contributions, focusing on FedAvg, FedProx, and subsequent enhancements, thereby positioning the present comparative study.

2.1 Foundations of Privacy-Preserving Learning

Before the formalization of FL, Shokri and Shmatikov (2015)[1] pioneered methods for privacy-preserving deep learning, where gradients were shared among participants instead of raw data. Although groundbreaking, these approaches required frequent communication and lacked scalability. Google later proposed FL as a practical and scalable framework for on-device learning, particularly for mobile keyboards and predictive typing systems [2].

2.2 Federated Averaging (FedAvg)

McMahan et al. (2017) [2]introduced Federated Averaging (FedAvg), which remains the foundational baseline in FL research. FedAvg leverages local SGD on client devices, followed by weighted model averaging at the server. The algorithm demonstrated that collaborative training of deep neural networks is feasible with limited communication. Despite its efficiency under IID settings, FedAvg struggles with client drift in Non-IID data distributions, leading to degraded performance and slower convergence [3].

2.3 Federated Proximal (FedProx)

To address FedAvg's limitations, Li et al. (2018) [4] proposed Federated Proximal (FedProx). FedProx modifies the local client objective by introducing a proximal term:

$$F_k(w)+\mu/2(||w-wt||)^2$$

This penalizes local models that drift too far from the global weights. Empirical studies demonstrated that FedProx improves stability and convergence in highly Non-IID scenarios, making it more robust for real-world federated systems.

2.4 Beyond FedAvg and FedProx: Variance Reduction and Adaptive Methods

Building on these foundations, several algorithms have been proposed to further mitigate client drift and enhance performance: SCAFFOLD (Karimireddy et al., 2020) [5]: Introduced control variates to correct local update drift, reducing variance and improving convergence rates under Non-IID data. FedNova (Wang et al., 2020) [6]: Proposed a normalized averaging scheme that accounted for varying local training epochs, improving fairness and stability. FedOpt (Reddi et al., 2020) [7]: Incorporated adaptive server optimizers (e.g., Adam, Yogi) into federated aggregation, significantly improving performance across diverse datasets. LEAF Benchmark (Caldas et al., 2018) [8]: Established standardized benchmarks for FL, including datasets and evaluation protocols, which highlighted the challenges of heterogeneity and reproducibility.

2.5 Applications and Real-World Deployments

Practical applications of FL have been explored in various domains: Mobile devices: Hard et al. (2018) [9] deployed FL for next-word prediction in Google's Gboard keyboard, demonstrating scalability to millions of users. Healthcare: Yang et al. (2019) [10] surveyed FL applications in sensitive fields such as electronic health records, medical imaging, and personalized medicine, where data privacy is paramount.

2.6 Research Gaps

Although significant progress has been made, two gaps remain: Many studies evaluate algorithms in synthetic or simplified scenarios, but few directly compare FedAvg and FedProx under both IID and Non-IID conditions with systematic experimental setups. While newer methods (SCAFFOLD, FedNova, FedOpt) outperform baseline algorithms in some scenarios, they introduce additional communication overhead and complexity, making FedAvg and FedProx still the most practical choices for benchmarking.

This motivates our work: a comprehensive comparative study of FedAvg and FedProx across IID and Non-IID distributions, using the EMNIST dataset, to provide insights into their strengths, weaknesses, and trade-offs.

3. Methodology

This section describes the experimental setup used to evaluate Federated Averaging (FedAvg) and Federated Proximal (FedProx) under both IID and Non-IID data distributions.

3.1 Dataset: EMNIST

We use the Extended MNIST (EMNIST) dataset, a benchmark introduced as an extension of MNIST for handwritten character recognition [2]. The EMNIST-Balanced split contains 47 classes, including digits and both uppercase and lowercase letters. Training samples: 131,600 Test samples: 22,400 Image size: 28×2828×28 grayscale images Preprocessing: Normalization with mean = 0.1307 and std = 0.3081 To reduce computational load, we selected 15% of the training set using stratified sampling to preserve class balance. EMNIST is widely adopted in federated learning research as it is challenging, imbalanced, and has multiple classes, making it suitable for evaluating algorithm robustness [8].

3.2 Data Partitioning: IID vs Non-IID

Data heterogeneity is a key challenge in FL [3]. We simulate 40 clients with the following partitioning strategies: IID Partitioning: Training samples are randomly and uniformly distributed across clients, ensuring balanced representation. Non-IID Partitioning: To mimic real-world scenarios, we apply a Dirichlet distribution with concentration parameter $\alpha=0.1\alpha=0.1$. Smaller values of $\alpha\alpha$ lead to greater heterogeneity, producing label skew where clients predominantly contain samples of specific classes [4]. This setup ensures a fair comparison of algorithms under both ideal (IID) and realistic (Non-IID) conditions.

3.3 Model Architecture: SmallCNN

A lightweight Convolutional Neural Network (CNN) is used as the base model, balancing efficiency and performance [2][8]. The architecture is as follows: Conv Layer 1: 32 filters, 3×33×3 kernel, ReLU activation, max pooling. Conv Layer 2: 64 filters, 3×33×3 kernel, ReLU activation, max pooling. Fully Connected Layer 1: 128 units, ReLU activation, with dropout (0.2). Fully Connected Layer 2: 47 units (output for EMNIST classes). This compact CNN is computationally efficient for federated simulations while still learning useful feature representations.

3.4 Training Configuration

The federated training follows the original design of FedAvg [2] and FedProx [4]:Total rounds: 60 communication rounds Clients per round: 10 (randomly selected) Local training: 1 epoch per round Optimizer: SGD with learning rate $\eta=0.01\eta=0.01$, momentum = 0.9 FedProx parameter: Proximal coefficient $\mu=0.01\mu=0.01$ This setup reflects a trade-off between computational feasibility and realism, consistent with prior FL benchmarks [6][8].

3.5 Algorithms (FedAvg and FedProx)

Federated Averaging (FedAvg)

Proposed by McMahan et al. (2017) 【2】, FedAvg combines local SGD updates from clients using weighted averaging at the server. While efficient for IID data, it suffers under Non-IID conditions due to client drift.

Mathematical Formulation:
For each round *t*:
Server samples clients St.
Each client $k \in S_t$ trains locally:
$w^{t+1}_{k}=\mathrm{w}^{\mathrm{t}}-\eta\nabla \mathrm{Fk}(\mathrm{w}^{\mathrm{t}})$
Server aggregates updates:
$\mathrm{w}^{\mathrm{t+1}}=\sum_{\mathrm{k}\in\mathrm{St}}(\mathrm{n}_{\mathrm{k}/}\mathrm{n})\, w^{t+1}_{k}$
where n_{k} is the number of samples on client k.

Federated Proximal (FedProx)

Introduced by Li et al. (2018) [4], FedProx extends FedAvg by adding a proximal term that constrains local models from drifting far from the global model. This improves stability in Non-IID data settings.

Mathematical Formulation:
Each client minimizes the modified loss function:

$$min_{\,w} F_k(w)+\mu/2(\|w-wt\|)^2$$

Local update rule:

$$w^{t+1}_{k}=\mathrm{w}^{\mathrm{t}}-\eta\nabla \mathrm{Fk}(\mathrm{w}^{\mathrm{t}}) + \mu(w^{t}-w))$$

Server aggregation remains identical to FedAvg.

3.6 Evaluation Metrics

We evaluate algorithms using both performance and efficiency metrics: Test Accuracy: Classification accuracy on test data. Test Loss: Cross-entropy loss, measuring generalization. Average Client Train Loss: Mean local training loss across clients. Communication Overhead (MB): Total model parameters exchanged between server and clients [7]. Training Time (seconds): End-to-end time for training convergence. Convergence AUC: Normalized area under the accuracy curve, capturing both speed and stability of convergence [5]. This set of metrics ensures a comprehensive comparison of FedAvg and FedProx under different data distributions.

4. Results and Discussion

The experimental results provide insights into the comparative performance of Federated Averaging (FedAvg) and Federated Proximal (FedProx) under IID and Non-IID data settings.

4.1 Performance under IID Settings

Under IID conditions, both FedAvg and FedProx achieve comparable accuracy levels, confirming that when data is uniformly distributed across clients, FedAvg is highly effective. Specifically, FedAvg converges slightly faster, as it does not incur the additional computational overhead of the proximal term. This observation aligns with the findings of McMahan et al. (2017) [2], who showed that FedAvg converges efficiently under homogeneous data distributions. In such scenarios, the benefit of FedProx's regularization is minimal, as there is limited divergence between client updates.

4.2 Performance under Non-IID Settings

In contrast, under Non-IID settings, FedAvg demonstrates instability and reduced accuracy**.** This degradation is a direct consequence of client drift, where local models trained on skewed data distributions diverge significantly from one another [3]. FedAvg's reliance on simple weighted averaging cannot adequately correct for these drifts, leading to slower convergence and suboptimal global performance.

FedProx addresses this limitation by incorporating a proximal regularizer, which constrains local updates and reduces the variance between client models [4]. Consequently, FedProx achieves more stable convergence curves and higher final test accuracy under heterogeneous conditions. This result corroborates Li et al. (2018) [4], who originally proposed FedProx as a robust alternative to FedAvg in heterogeneous networks.

4.3 Convergence Analysis (AUC)

The Area Under the Curve (AUC) for accuracy provides an aggregate view of convergence over communication rounds. In Non-IID scenarios, FedProx consistently achieves higher AUC compared to FedAvg, reflecting both faster stabilization and improved cumulative accuracy. These results are consistent with theoretical analyses of variance reduction in federated optimization [5][6].

4.4 Communication and Time Efficiency

Communication cost and training time are critical considerations in FL, especially in large-scale deployments such as mobile devices [7]. Our budget analysis demonstrates that, for limited communication bandwidth or strict time constraints, FedProx provides better efficiency in Non-IID settings, as it achieves higher accuracy earlier in training compared to FedAvg. While FedAvg is more communication-efficient in IID settings, its instability in Non-IID environments makes it less suitable for practical deployments.

This trade-off reflects a broader challenge identified in the literature: balancing accuracy, convergence stability, and efficiency. Approaches such as FedOpt [7] and Scaffold [5] also attempt to address these issues, but often introduce additional communication overhead. Compared to these methods, FedProx provides a practical compromise**:** moderate overhead with significant gains in heterogeneous conditions.

4.5 Visual Analysis of Results

The experimental figures including accuracy curves, loss curves, bar charts for final accuracy, and scatter plots of accuracy versus communication further support the quantitative findings. Accuracy curves illustrate the rapid initial convergence of FedAvg under IID, while loss curves and scatter plots highlight the greater stability and efficiency of FedProx under Non-IID. These visual results resonate with prior benchmarks such as LEAF [8], which emphasized the importance of evaluating algorithms across diverse data distributions.

The test accuracy vs. rounds analysis highlights the contrasting behavior of FedAvg and FedProx under different data distributions. In the IID setting, both algorithms converge rapidly to high accuracy, with FedAvg showing slightly faster improvements since it does not incur the computational overhead of the proximal term, consistent with the findings of McMahan et al. (2017) [2]. However, under Non-IID conditions, FedAvg suffers from unstable convergence and oscillations due to client drift, often plateauing at suboptimal accuracy [3]. In contrast, FedProx demonstrates smoother and more consistent accuracy gains across rounds, ultimately achieving higher final accuracy. This improvement arises from the proximal regularization, which constrains local updates and mitigates divergence among heterogeneous clients [4][6]. These trends suggest that while FedAvg remains optimal for homogeneous data distributions requiring rapid convergence, FedProx is more robust and reliable in real-world heterogeneous environments where stability and accuracy are paramount.

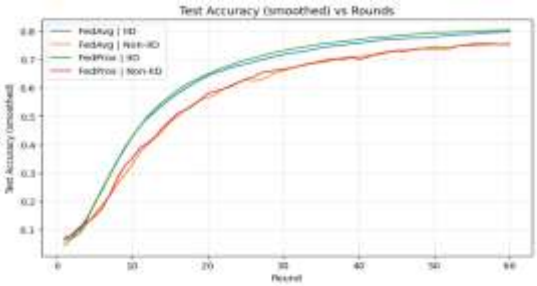

Fig -2: Test accuracy vs. rounds analysis

The final accuracy bar chart clearly illustrates the trade-offs between FedAvg and FedProx across IID and Non-IID partitions. In the IID scenario, both algorithms achieve nearly identical final accuracies, with FedAvg showing a marginal advantage due to its communication efficiency and lack of additional regularization overhead [2]. However, in the Non-IID setting, the difference becomes substantial: FedAvg struggles to maintain accuracy because of client drift and inconsistent updates, while FedProx achieves significantly higher final accuracy owing to its proximal term that stabilizes convergence [3][4]. This visual comparison reinforces the conclusion that while FedAvg remains suitable for homogeneous data distributions, FedProx is more effective in heterogeneous environments where robustness and stability are critical [6].

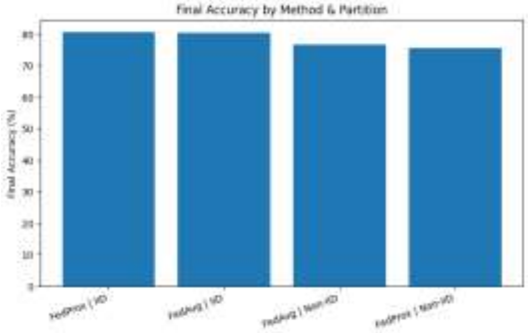

Fig -3: Final accuracy by method and partion

The final accuracy vs. communication plot highlights the efficiency–robustness trade-off between FedAvg and FedProx. Under IID conditions, FedAvg achieves high accuracy with relatively lower communication overhead, confirming its strength as a communication-efficient baseline [2]. However, in the Non-IID setting, FedAvg requires comparable or even greater communication but still fails to reach optimal accuracy due to client drift [3]. In contrast, FedProx achieves higher final accuracy in heterogeneous environments, albeit with slightly higher communication costs, since its proximal term ensures more stable updates across clients [4][6]. This demonstrates that while FedAvg is optimal when communication efficiency is paramount under IID data, FedProx provides better accuracy–communication trade-offs in realistic, skewed data distributions.

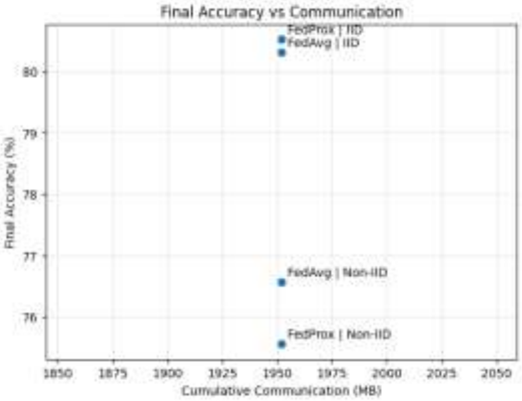

Fig -4: Final accuracy vs Communication

The accuracy vs. time plot provides insights into the temporal efficiency of FedAvg and FedProx across IID and Non-IID data distributions. In the IID case, FedAvg consistently reaches high accuracy faster than FedProx, reflecting its lower computational overhead and confirming its suitability for environments where rapid convergence is essential [2]. However, under Non-IID conditions, FedAvg's accuracy growth over time is irregular and often plateaus early due to client drift [3]. FedProx, on the other hand, shows steady accuracy improvements as time progresses, ultimately achieving superior performance despite requiring slightly longer training. This trend highlights the robustness of FedProx in heterogeneous scenarios, aligning with prior work that emphasizes the importance of stabilizing local updates for long-term convergence [4][6]. Overall, the plot demonstrates that FedAvg is more time-efficient in homogeneous settings, while FedProx provides more reliable accuracy gains in heterogeneous environments.

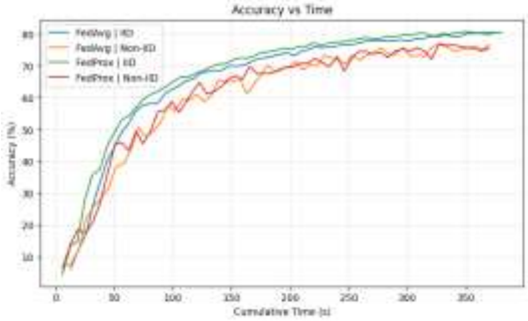

Fig -5: Final accuracy vs Time

4.6 Key Insights

From the above observations, several insights emerge: FedAvg is optimal for IID data, offering faster convergence with minimal complexity. FedProx outperforms FedAvg in Non-IID settings, achieving better stability and accuracy by mitigating client drift. Trade-off between efficiency and robustness**:** FedAvg is more communication-efficient under homogeneous data, but FedProx is more practical for real-world heterogeneous federated systems. These findings highlight the importance of algorithm selection based on data distribution characteristics**,** echoing recommendations in the survey by Kairouz et al. (2021) [3].

5. Conclusion

This comparative study between Federated Averaging (FedAvg) and Federated Proximal (FedProx) provides a deeper understanding of the trade-offs involved in optimizing federated learning under different data distributions. The results confirm that FedAvg is highly effective in IID environments, where uniform data distribution across clients allows for rapid convergence and communication efficiency [2]. However, in Non-IID scenarios, FedAvg suffers from instability and reduced accuracy due to client drift [3]. In contrast, FedProx introduces a proximal regularization term that stabilizes local training, thereby significantly improving convergence and accuracy in heterogeneous settings [4].

A key insight from this work is that algorithm selection in FL should depend on the data distribution characteristics. In homogeneous environments (e.g., cross-device FL with balanced partitions), FedAvg remains the most communication-efficient baseline. In heterogeneous, real-world applications (e.g., healthcare, IoT, or mobile devices where data is inherently skewed), FedProx provides superior robustness and stability, making it a more practical choice [3][4].

Furthermore, convergence analysis using AUC metrics and budget-based evaluation illustrates the importance of measuring not only the final accuracy but also the efficiency of the training process. These metrics provide valuable insights for large-scale deployments where communication bandwidth and device energy consumption are critical [7][8].

References

1. Shokri, R., & Shmatikov, V. (2015). Privacy-preserving deep learning. *ACM CCS*.
2. McMahan, H. B., Moore, E., Ramage, D., Hampson, S., & Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. *AISTATS*.
3. Kairouz, P., et al. (2021). Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1-2), 1–210.
4. Li, T., Sahu, A. K., Talwalkar, A., & Smith, V. (2018). Federated optimization in heterogeneous networks. *Proceedings of MLSys*.
5. Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S., Stich, S., & Suresh, A. T. (2020). SCAFFOLD: Stochastic controlled averaging for federated learning. *ICML*.
6. Wang, J., Charles, Z., Xu, Z., Joshi, G., & Poor, H. V. (2020). A novel convergence analysis for federated learning under data heterogeneity. *NeurIPS*.
7. Reddi, S. J., et al. (2020). Adaptive federated optimization. *ICLR*.
8. Caldas, S., et al. (2018). LEAF: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*.
9. Hard, A., et al. (2018). Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*.
10. Yang, Q., et al. (2019). Federated machine learning: Concept and applications. *ACM TIST*.