On the Effectiveness of Federated Aggregation Methods Across Different Model Types

Cole Feuer, Tyler Thraves, Sahithi Kamireddy

***Abstract*—Federated learning has emerged as a powerful technique for decentralized model training, enabling privacy-preserving and communication-efficient learning across distributed data sources. Central to the effectiveness of federated learning is the choice of aggregation method, which directly impacts convergence speed, model generalization, and resilience to non-iid data. In this work, we conduct an empirical study comparing three prominent aggregation methods: FedAvg, FedLAMA, and FedDist, across different model architectures and data modalities. Using a convolutional neural network (CNN) trained on the CIFAR-10 dataset, as well as a multi-layer perceptron (MLP) trained on the Mimic3 dataset and a Transformer trained on the Sentiment140 dataset, we benchmark these methods in terms of classification accuracy, F1 score, and training loss. Our results reveal significant trade-offs between methods, particularly in terms of convergence and overfitting tendencies, offering insights into the selection of aggregation strategies for practical federated deployments.**

I. Introduction

With the increasing prevalence of massive, distributed datasets, centralized machine learning faces significant scalability, efficiency, and privacy challenges. Centralized training requires aggregating all data to a single server, creating storage bottlenecks, longer training times, and critical privacy vulnerabilities. For sensitive domains such as healthcare, finance, and personal devices, users often cannot or will not share raw data due to privacy concerns [3], [6].

Federated learning (FL) addresses these issues by training models locally on decentralized devices and aggregating only model updates, rather than raw data [5], [4]. This paradigm improves data privacy, reduces communication overhead, and enables learning from otherwise siloed datasets. However, the performance of a federated system heavily depends on the aggregation method used to combine updates from local clients.

II. Related Work

FedAvg [1], [5] introduced the foundational idea of simple averaging of local model weights and demonstrated its scalability. Despite its simplicity, FedAvg can suffer under data heterogeneity (non-iid distributions).

Recent improvements include FedLAMA [2], [7], which proposes layer-wise adaptive aggregation to address the imbalance of updates across model layers. FedLAMA dynamically adjusts aggregation frequency based on the significance of updates, improving convergence and communication efficiency.

FedDist [8] diverges from weight aggregation entirely, instead aggregating predictions on a shared public dataset to guide the central model's updates. This ”output-level” aggregation can improve robustness to client drift but raises questions about overfitting and generalization.

III. Federated Aggregation Methods

A. FedAvg

FedAvg operates by averaging model parameters across all participating clients after each training round [5]:

$$\theta \leftarrow \frac{1}{N}\sum_{i=1}^{N}\theta_i \tag{1}$$

where θ_i represents the parameters from the i-th client.

FedAvg aims to minimize the global objective:

$$\min_{\theta}\sum_{i=1}^{N}\frac{n_i}{n}\mathcal{L}_i(\theta) \tag{2}$$

where $\mathcal{L}_i$ is the local loss for client i, n_i is the number of data points on client i and $n = \sum_i n_i$ is the total number of samples.

B. FedLAMA

FedLAMA adapts FedAvg by introducing adaptive layer-wise aggregation [2]. It monitors discrepancy per layer l:

$$d_l = \frac{\|\theta^l - \bar{\theta}^l\|_2}{|\theta^l|} \tag{3}$$

where θ^l denotes the parameters of layer l in the client and $\bar{\theta}^l$ denotes the corresponding parameters in the core model. If $d_l < \epsilon$, the aggregation interval is increased by a factor γ. Otherwise, the layer is aggregated normally.

This mechanism selectively synchronizes only significant changes, reducing communication cost while maintaining model quality.

C. FedDist

FedDist shifts aggregation from the parameter space to the prediction space [9]. After local training, each client generates softmax outputs $p_i(x)$ for a shared public dataset D_p. The core model aggregates these predictions[8]:

$$\bar{p}(x) = \frac{1}{N}\sum_{i=1}^{N}p_i(x) \tag{4}$$

where $\bar{p}(x)$ represents the pseudo-label distribution for sample x. The central model is then trained to minimize the KL divergence between its predictions and the aggregated pseudo-labels:

$$\min_{\theta}\mathbb{E}_{x\sim D_p}\left[\mathrm{KL}(\bar{p}(x)\,\|\,p_c(x))\right] \tag{5}$$

This method can tolerate diverse model architectures but depends heavily on the quality of the public dataset and the coherence of pseudo-labels.

IV. Experimental Setup

To evaluate the comparative effectiveness of federated aggregation methods, we implemented a comprehensive experimental framework encompassing a variety of model architectures and data modalities. Our experiments span image classification, tabular data classification, and sentiment analysis tasks. We detail the datasets, model configurations, and training protocols used for each setting.

A. Datasets

We selected three diverse datasets to test each aggregation method under different data modalities:

- **CIFAR-10** [9]: A benchmark dataset for image classification consisting of 60,000 32x32 color images across 10 classes.
- **MIMIC-III**: A widely used dataset for healthcare research comprising de-identified clinical data from intensive care unit patients. We process it for binary classification tasks on tabular data.
- **Sentiment140**: A large dataset of 1.6 million tweets labeled for sentiment analysis (positive or negative), used here to evaluate the transformer-based model.

B. Model Architectures

CNN for CIFAR-10: Our convolutional neural network includes three convolutional layers with 32, 64, and 64 filters respectively (kernel size 3x3), each followed by ReLU activations. Two max-pooling layers (2x2) reduce spatial dimensions. The final layers include a dense layer with 64 units and an output softmax layer for 10-class classification.

MLP for MIMIC-III: The multi-layer perceptron consists of three dense layers with 128, 256, and 1 unit respectively. ReLU activations are used for the first two layers, while the final output layer uses a tanh activation for binary classification.

Transformer for Sentiment140: We utilize DistilBERT (distilbert-base-uncased) as a lightweight transformer model. Input tweets (up to 128 tokens) are encoded via token IDs and attention masks. The [CLS] token's hidden state passes through a dropout layer and two fully connected layers (768→768 with ReLU, and 768→2 output logits).

Model code and training scripts are available at: https://github.com/thravt/FederatedAggregation.

C. Training Protocols

Each federated training setup involved multiple clients performing local training and a central server aggregating updates. We adopted the following configurations:

- **CNN:** 5 clients, each trained for 25 local epochs per round. Optimized using Adam with sparse categorical cross-entropy.
- **MLP:** 10 clients, each trained for 25 local epochs (except FedDist, which uses 5 due to pseudo-label overfitting). Optimized with Adam and binary cross-entropy loss.
- **DistilBERT:** Fine-tuned using balanced mini-batches for 10 local epochs. Cross-entropy loss is optimized using Adam with a learning rate of 2e-5. Aggregation occurs across 5 simulated clients.

These configurations aim to simulate practical federated deployments while exposing differences in aggregation effectiveness across heterogeneous tasks.

V. Results

Method	Accuracy	F1 Score (%)	Loss
FedAvg	64.44%	**19.56%**	1.2690
FedDist	59.56%	19.56%	4.4741
FedLAMA	**68.86%**	19.55%	**1.0266**

TABLE I
Performance of Aggregation Methods for CNN

Method	Accuracy	F1 Score (%)	Loss
FedAvg	**92.64%**	**55.97%**	0.4906
FedDist	90.07%	0.13%	0.5470
FedLAMA	91.57%	53.46%	**0.3160**

TABLE II
Performance of Aggregation Methods for MLP

Method	Accuracy	F1 Score (%)	Loss
FedAvg	62.00%	61.60%	0.6590
FedDist	51.25%	34.73%	4.6817
FedLAMA	**63.00%**	**62.78%**	**0.6784**

TABLE III
Performance of Aggregation Methods for DistilBERT Sentiment Classification

VI. Discussion

Our results underscore the variability in performance of aggregation methods across different model architectures and data modalities.

In the image classification task (CNN on CIFAR-10), FedLAMA outperformed both FedAvg and FedDist in terms of accuracy and loss. This highlights the benefit of adaptive synchronization in settings where local updates differ significantly between layers. The ability of FedLAMA to selectively reduce communication while preserving crucial updates likely contributed to its superior performance. FedAvg performed moderately well but struggled to generalize as effectively. FedDist underperformed significantly, suffering from a high loss likely caused by pseudo-label noise and poor generalization.

For the tabular classification task using MLP on MIMIC-III, FedAvg yielded the highest accuracy and F1 score, suggesting that simple averaging can be effective in low-dimensional, less hierarchical settings. However, FedLAMA still maintained the lowest loss, indicating better convergence in terms of optimization even if final predictive performance was marginally behind FedAvg. FedDist performed poorly, with near-zero F1 scores, indicating its difficulty in handling tabular data without strong output label consistency.

In the text-based sentiment classification task using DistilBERT, FedLAMA again proved to be the most robust, delivering the highest accuracy and F1 score. This suggests that its adaptive approach effectively addresses the gradient variance issues inherent in transformer-based models. FedAvg was close behind, whereas FedDist showed a dramatic drop in both accuracy and F1, with the highest loss, likely due to the incompatibility between pseudo-labeling and the high-dimensional representation space of transformers.

These observations highlight that while FedAvg is generally reliable, FedLAMA's adaptability makes it more effective in complex models and non-iid environments. FedDist's dependence on pseudo-label aggregation severely limits its performance unless supported by a highly representative public dataset and well-aligned client outputs.

VII. Conclusion

This study presents a comparative evaluation of three federated aggregation methods—FedAvg, FedLAMA, and FedDist—across image, tabular, and textual data domains. Our experiments reveal that no single method universally outperforms the others across all settings; rather, the optimal choice is task- and architecture-dependent.

FedLAMA emerged as the most consistently strong performer, excelling in both CNN and transformer-based settings due to its adaptive synchronization strategy. FedAvg, while simpler, was competitive in less complex MLP-based tasks and text classification. Conversely, FedDist showed limited effectiveness, particularly under heterogeneous client behavior or when the shared public dataset lacked sufficient representativeness.

In practice, these results suggest that adaptive aggregation strategies like FedLAMA can offer substantial benefits in realistic federated environments with diverse clients and model types. Future work will explore the effects of client sampling strategies, data heterogeneity, and adversarial robustness. We also plan to extend our study to larger datasets, cross-device scenarios, and other aggregation paradigms including Bayesian and attention-based methods.

References

[1] L. Collins et al., ”FedAvg with Fine Tuning: Local Updates Lead to Representation Learning,” 2022.

[2] S. Lee et al., ”Layer-Wise Adaptive Model Aggregation for Scalable Federated Learning,” 2022.

[3] G. Long et al., ”Federated Learning for Privacy-Preserving Open Innovation Future on Digital Health,” 2021.

[4] S. Bharati et al., ”Federated Learning: Applications, Challenges and Future Directions,” 2022.

[5] B. McMahan et al., ”Communication-Efficient Learning of Deep Networks from Decentralized Data,” 2017.

[6] P. Voigt and A. von dem Bussche, ”The EU General Data Protection Regulation (GDPR),” 2017.

[7] S. Karimireddy et al., ”SCAFFOLD: Stochastic Controlled Averaging for Federated Learning,” 2020.

[8] Y. Lin et al., ”Ensemble Distillation for Robust Model Fusion in Federated Learning,” 2020.

[9] A. Krizhevsky, ”Learning Multiple Layers of Features from Tiny Images,” 2009.