Robust Speech Command Recognition via Federated Learning on Heterogeneous Data

MD DAUD SHAKIL
Student ID: 102774360
mddaud.shakil@aalto.fi

***Abstract*—This paper investigates the application of Federated Learning (FL) to speech command recognition, addressing the critical challenge of statistical heterogeneity across decentralized clients. We train a Convolutional Neural Network (CNN) on the Google Speech Commands dataset, simulating a realistic FL scenario with 10 clients exhibiting non-IID data. Heterogeneity is introduced through skewed command distributions, varying noise, and distinct speaker populations. We implement and rigorously evaluate several FL algorithms: Standalone (isolated) training, Federated Averaging (FedAvg), and Federated Proximal (FedProx), alongside two personalized FL methods. Our evaluation demonstrates that federated methods significantly outperform isolated training. FedProx emerges as the most effective algorithm, achieving the highest test accuracy (93.51%). Analysis of training and validation losses reveals that FedProx also acts as a regularizer, mitigating local overfitting and showcasing superior robustness to data heterogeneity.**

***Index Terms*—Federated Learning, Speech Recognition, Data Heterogeneity, Non-IID Data, Convolutional Neural Networks, FedAvg, FedProx**

I. Introduction

The proliferation of voice-enabled devices has made speech recognition a cornerstone of modern human-computer interaction. Performance hinges on training models with vast datasets. Traditionally, this required centralizing user voice recordings, a practice that poses significant privacy risks as voice data can be a biometric identifier [1]. These centralized systems, such as the automatic isolated speech recognition (AISR) system described in [8], typically process a unified dataset of audio features with a neural network classifier.

Federated Learning (FL) offers a privacy-preserving paradigm shift [2]. By training models locally on user devices and only aggregating anonymized model updates, FL mitigates privacy concerns. However, this introduces a formidable technical challenge: statistical heterogeneity. In real-world FL, each device's data is not an independent and identically distributed (IID) sample of the overall distribution [3]. This non-IID nature is pronounced in speech applications due to varied accents (feature skew), command frequencies (label skew), and acoustic environments. Standard FL algorithms like Federated Averaging (FedAvg) can suffer from "client drift," where local models diverge and their average results in a suboptimal global model [3].

This paper provides a rigorous empirical investigation into overcoming this challenge for a speech command recognition task. Our contributions are: 1) a comprehensive comparison of standard, robust, and personalized FL algorithms under simulated, multi-faceted heterogeneity; 2) a detailed analysis of how different heterogeneity types impact performance; and 3) a clear demonstration that FedProx acts as a regularizer, reducing local overfitting.

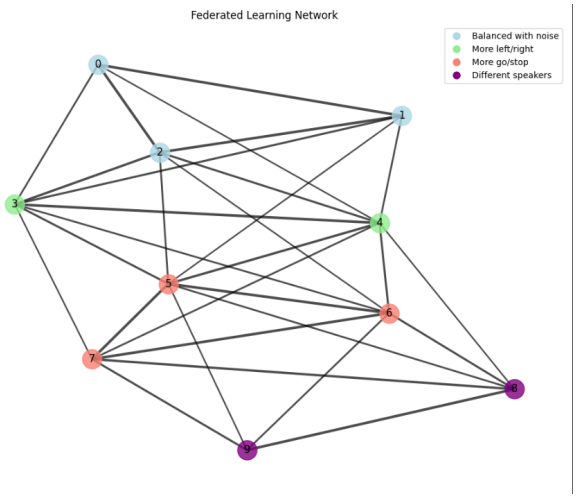


Fig. 1: The federated learning network, showing 10 clients colored by heterogeneity type. Edge thickness represents inter-client similarity for pFedAvg.

II. Problem Formulation

We model the application as an FL network [1, Ch. 3]:

- **Nodes:** $n = 10$ clients $V = \{1, \dots, 10\}$, with local data $\mathcal{D}^{(i)}$.
- **Local Models:** Identical CNN $\mathcal{H}^{(i)}$ with parameters $\mathbf{w}^{(i)} \in \mathbb{R}^d$, processing MFCCs.
- **Loss Functions:** Minimize cross-entropy $L_i(\mathbf{w}^{(i)})$ (Eq. (1)) on $\mathcal{D}^{(i)}$.

$$L_i(\mathbf{w}^{(i)}) = -\frac{1}{|\mathcal{D}^{(i)}|} \sum_{(\mathbf{x},\mathbf{y})\in\mathcal{D}^{(i)}} \mathbf{y}^T \log(\mathbf{p}(\mathbf{x}; \mathbf{w}^{(i)})) \quad (1)$$

FedProx adds a proximal term:

$$\arg\min_{\mathbf{w}} \left\{ L_i(\mathbf{w}) + \frac{\mu}{2} \|\mathbf{w} - \mathbf{w}^{(k)}_{\text{global}}\|_2^2 \right\} \quad (2)$$

- **Edges:** FedAvg/FedProx: star graph. pFedAvg: undirected graph $\mathcal{G} = (V, \mathcal{E})$ with weights $A_{i,i'}$ based on data similarity (Fig. 1).

III. Methodology

We leverage the GTVMin framework as a conceptual basis for collaborative training [1, Ch. 5], which seeks to solve:

$$\min_{\mathbf{w}^{(1)},\ldots,\mathbf{w}^{(n)}} \sum_{i=1}^{n} L_i(\mathbf{w}^{(i)}) + \alpha \sum_{\{i,i'\}\in\mathcal{E}} A_{i,i'}\phi(\mathbf{w}^{(i)} - \mathbf{w}^{(i')}) \quad (3)$$

We use the squared Euclidean norm $\phi(\mathbf{u}) = \|\mathbf{u}\|_2^2$ as our variation measure. Each implemented algorithm represents a different strategy for optimizing or approximating this objective.

1) Standalone: This non-collaborative baseline has each client minimize only its local loss $L_i(\mathbf{w}^{(i)})$, corresponding to GTVMin with $\alpha = 0$. It establishes the performance achievable without any collaboration.

2) FedAvg: This is the standard FL benchmark [2]. In each round k, a subset of clients $C^{(k)}$ receives the global model $\mathbf{w}_{\text{global}}^{(k)}$, performs E_{local} local SGD steps to get an updated $\mathbf{w}^{(i)}$, and returns it to the server for averaging:

$$\mathbf{w}_{\text{global}}^{(k+1)} = \frac{1}{|C^{(k)}|} \sum_{i\in C^{(k)}} \mathbf{w}^{(i)} \quad (4)$$

3) FedProx: To explicitly combat client drift, this algorithm [3] modifies FedAvg by having clients solve the proximally-regularized objective in Eq. (2) during their local training steps.

4) pFedAvg (Neighbor Aggregation): This personalized method uses the graph $\mathcal{G}$. In each round k, clients first train locally to get $\mathbf{w}_{\text{local}}^{(i,k+1)}$. They then form an aggregated neighbor model, $\mathbf{w}_{\text{neigh}}^{(i,k+1)}$, and combine it with their own:

$$\mathbf{w}_{\text{agg}}^{(i,k+1)} = (1-\lambda_{\text{reg}})\mathbf{w}_{\text{local}}^{(i,k+1)} + \lambda_{\text{reg}}\mathbf{w}_{\text{neigh}}^{(i,k+1)} \quad (5)$$

This is followed by a final, single epoch of fine-tuning.

5) pFedAvg V2 (Global Interpolation): This simpler personalization method has clients train locally to get $\mathbf{w}_{\text{local}}^{(i,k+1)}$ and then interpolate with the global model from the start of the round:

$$\mathbf{w}^{(i,k+1)} = (1-\lambda_{\text{reg\_v2}})\mathbf{w}_{\text{local}}^{(i,k+1)} + \lambda_{\text{reg\_v2}}\mathbf{w}_{\text{global}}^{(k)} \quad (6)$$

IV. Experimental Setup

A. Data, Heterogeneity, and Preprocessing

We use the Google Speech Commands v0.02 dataset [4] for a 4-command keyword spotting task ("go", "left", "right", "stop"). All audio is preprocessed into MFCCs. To simulate a realistic FL scenario, the dataset is partitioned across 10 clients, introducing heterogeneity as follows:

- **Balanced with Noise (Clients 0-2):** Balanced commands with varied background noise levels to simulate different acoustic environments.
- **Label Skew (Clients 3-7):** Some clients have a majority of "left/right" commands, while others favor "go/stop".
- **Feature Skew (Clients 8-9):** Data is sourced from a distinct set of speakers, creating a different feature distribution (e.g., pitch, accent).

B. FL Configuration and Evaluation

The training process runs for $R = 20$ communication rounds. Performance is evaluated using average test accuracy, training/validation/test loss curves, and paired t-tests for statistical significance.

C. Reproducibility

The experiment uses standard libraries (TensorFlow, Keras, Librosa, NumPy, Scikit-learn) and the public Google Speech Commands dataset. Random seeds (42) were set for NumPy and TensorFlow to aid reproducibility.

V. Results and Discussion

A. Overall Performance Analysis

The final average performance metrics are summarized in Table II. FedProx emerges as the top-performing algorithm with an average test accuracy of 93.51%, a relative improvement of 14.8% over Standalone training. The personalized FL methods failed to deliver a meaningful improvement over the baseline.

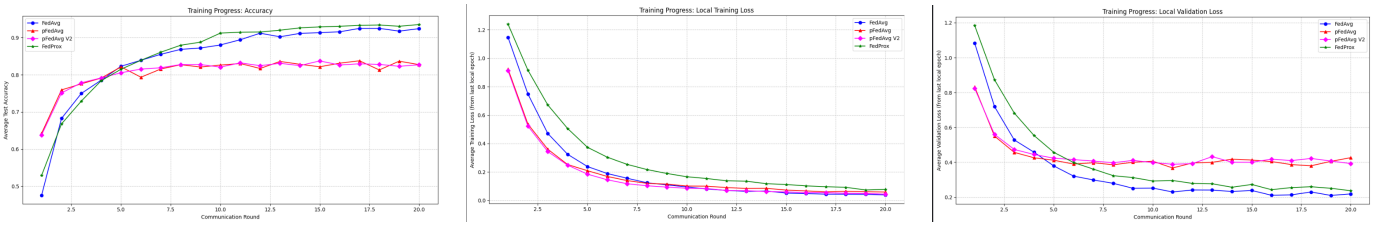
To validate these observations, we performed paired t-tests on the per-client accuracies, summarized in Table III. The tests confirm that the improvements of FedAvg and FedProx over Standalone training are highly significant ($p < 0.0001$). While FedProx outperforms FedAvg in average accuracy, this difference was not statistically significant ($p = 0.0515$), though it consistently shows lower test loss. Crucially, both FedAvg and FedProx are significantly better than the personalized methods. These results provide strong statistical evidence that a robust, global model is the superior approach for this task.

B. Convergence and Overfitting Dynamics

The training dynamics in Fig. 2 show that FedProx and FedAvg converge rapidly, reaching near-peak accuracy within 10-12 rounds. The most critical insight into their differing behaviors comes from analyzing the relationship between training loss (Fig. 2b) and validation loss (Fig. 2c). For FedAvg, the training loss plummets to a very low value (0.0393), indicating that the local models are aggressively overfitting to their specific, non-IID data partitions. In stark contrast, FedProx's training loss remains consistently higher (0.0781). This provides strong empirical evidence that FedProx's proximal term is successfully acting as a regularizer. It prevents the local models from fitting too closely to their skewed data, forcing them to stay nearer to the global objective. While this regularization leads to a slightly higher validation loss at the end of training compared to FedAvg, it is the key mechanism that reduces client drift and ultimately produces a more robust final global model with higher test accuracy.

TABLE I: Detailed Per-Client Final Test Accuracies and Losses After 20 Communication Rounds

	Standalone		FedAvg		pFedAvg		pFedAvg V2		FedProx	
Client (Type)	**Acc**	**Loss**	**Acc**	**Loss**	**Acc**	**Loss**	**Acc**	**Loss**	**Acc**	**Loss**
Client 0 (Bal)	0.9259	0.4164	0.9352	0.3153	0.8889	0.3507	0.9259	0.3202	**0.9630**	0.1306
Client 1 (Bal)	0.7619	0.7295	**0.8889**	0.8052	0.7937	0.6800	0.8571	0.4853	**0.8889**	0.5253
Client 2 (Bal)	0.8767	0.5227	0.9521	0.1618	0.8904	0.3745	0.9041	0.2526	**0.9658**	0.0803
Client 3 (L/R)	0.7662	0.6345	0.9351	0.1664	0.7922	0.6198	0.8312	0.5080	**0.9481**	0.1570
Client 4 (L/R)	0.8772	0.3135	**1.0000**	0.0107	0.8421	0.3710	0.8596	0.4027	**1.0000**	0.0269
Client 5 (G/S)	0.8485	0.6668	0.9091	0.2867	0.8485	0.7942	0.8636	0.5195	**0.9242**	0.3349
Client 6 (G/S)	0.8469	0.5655	0.9286	0.6071	0.8571	0.5432	0.8469	0.5810	**0.9694**	0.3701
Client 7 (G/S)	0.8028	0.6092	**0.9437**	0.2138	0.8310	0.5223	0.7746	0.6656	0.9296	0.2743
Client 8 (Spk)	0.7206	0.8104	0.8676	0.7533	0.8088	0.8021	0.7647	0.6803	**0.8824**	0.4667
Client 9 (Spk)	0.7200	1.1634	**0.8800**	0.6319	0.7200	1.0705	0.6400	0.8679	**0.8800**	0.8792



(a) Average Test Accuracy (b) Average Training Loss (c) Average Validation Loss

Fig. 2: Training progress over 20 rounds. (a) FedProx and FedAvg converge to superior accuracy. (b) FedAvg's training loss collapses, indicating aggressive local overfitting. (c) FedProx's stable validation loss demonstrates the regularizing effect of its higher training loss, which improves final generalization.

TABLE II: Final Average Test Accuracy and Loss

Algorithm	**Avg Accuracy**	**Avg Test Loss**
Standalone	0.8147	0.6432
FedAvg	0.9240	0.3952
pFedAvg	0.8273	0.6129
pFedAvg V2	0.8268	0.5283
FedProx	**0.9351**	**0.3245**

TABLE III: Statistical Significance (Paired T-Tests)

Comparison	**p-value (Significance)**
FedProx vs. Standalone	< 0.0001 (Significant)
FedAvg vs. Standalone	< 0.0001 (Significant)
FedProx vs. FedAvg	0.0515 (Not Significant)
FedProx vs. pFedAvg	< 0.0001 (Significant)

C. Robustness and Per-Client Analysis

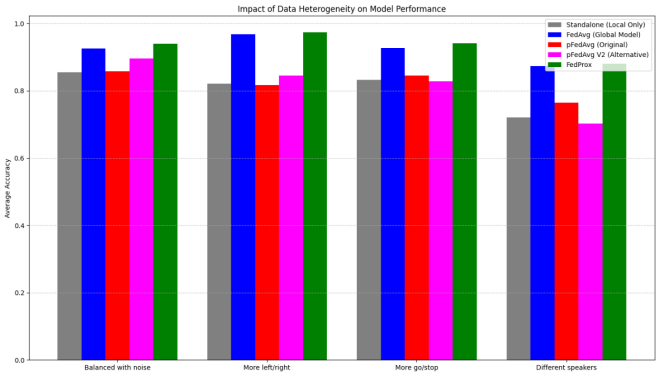
The robustness of the algorithms to specific data distributions is analyzed in Fig. 3. Fig. 3a isolates performance by heterogeneity type, showing that FedProx and FedAvg are highly effective across all groups. Their advantage is starkest on clients with feature skew ("Different speakers"), where FedProx provides a massive 16.1% absolute accuracy uplift over Standalone training. Fig. 3b, along with the data in Table I, shows this benefit at the individual client level. FedProx consistently delivers top-tier accuracy, enabling even clients with the most challenging data (e.g., Client 9) to achieve excellent results, raising their accuracy from 72.0% to 88.0%.

VI. Conclusion and Future Work

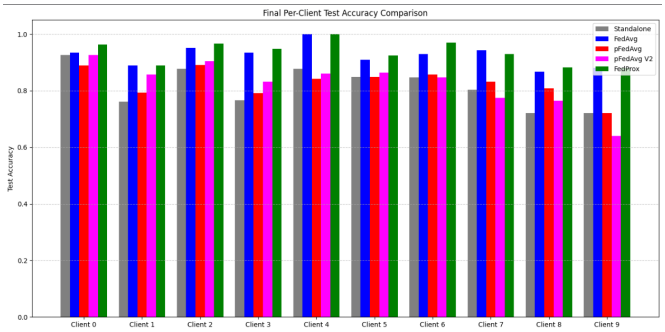
This project provided a comprehensive analysis of federated learning for speech command recognition under realistic conditions of data heterogeneity. We successfully demonstrated that collaborative training via FL offers significant performance benefits over isolated, on-device training.

Our key finding is the clear superiority of FedProx. It not only achieved the highest final test accuracy but also converged to the lowest average test loss, demonstrating superior overall performance. Crucially, by analyzing the loss curves, we showed that FedProx performs a dual role: it directly combats the client drift problem caused by non-IID data, and it simultaneously acts as a regularizer during local training—as evidenced by its higher training loss compared to FedAvg—reducing overfitting and improving the global model's generalization. While standard FedAvg also performed well, FedProx's consistent edge underscores the value of using algorithms explicitly designed for the challenges of heterogeneous data.

Future work could first address the limitations of the simple personalized methods explored here. A particularly promising direction is Personalized Federated Learning (pFL) through the lens of meta-learning. Instead of training a single global model that performs best on the average data distribution, the goal of federated meta-learning is to train a global model initialization that can be rapidly adapted to each client's unique data distribution with only a few steps of local fine-tuning. This approach, exemplified by algorithms like Model-Agnostic



(a) Impact of Heterogeneity by Client Type



(b) Final Per-Client Accuracy Comparison

Fig. 3: Final accuracy analysis. (a) FedProx and FedAvg show the most robust performance across all heterogeneity groups. (b) This superior performance is visualized across nearly all individual clients.

Meta-Learning (MAML) [5], directly optimizes for a model that is easy to personalize, a more principled approach than the simple interpolation used in this study.

Furthermore, this study assumed a synchronous, idealized environment. Real-world deployments must contend with system-level heterogeneity, where clients have varied computational resources and network connectivity, leading to a ”straggler” problem. Future work should therefore explore asynchronous FL protocols, such as FedAsync [6], which allow the server to perform updates without waiting for all clients, making the system more robust.

Finally, while FL provides a strong privacy baseline, model updates can still leak information. To provide mathematically provable privacy guarantees, future work could integrate technologies like Differential Privacy (DP) [7]. This involves adding calibrated noise to client updates before aggregation, creating a crucial trade-off between the strength of the privacy guarantee and the final model’s accuracy.

REFERENCES

[1] A. Jung, *Federated Learning: From Theory to Practice*, Aalto, 2025. Available: https://github.com/alexjungaalto/FederatedLearning/blob/main/material/FLBook.pdf.

[2] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proc. Int. Conf. Artif. Intell. Stat. (AISTATS)*, PMLR vol. 54, pp. 1273–1282, 2017.

[3] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated Optimization in Heterogeneous Networks," in *Proc. Mach. Learn. Syst. (MLSys)*, vol. 2, pp. 429–450, 2020.

[4] P. Warden, "Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition," *arXiv preprint arXiv:1804.03209*, 2018.

[5] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach," in *Advances in Neural Information Processing Systems*, 2020, vol. 33, pp. 3557–3568.

[6] X. Xie, S. Lu, H. Wang, and S. Pu, "Asynchronous federated optimization," *arXiv preprint arXiv:1903.03934*, 2019.

[7] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning differentially private recurrent language models," *arXiv preprint arXiv:1710.06963*, 2017.

[8] M. D. Shakil, M. A. Rahman, M. M. Soliman, and M. A. Islam, "Automatic Isolated Speech Recognition System Using MFCC Analysis and Artificial Neural Network Classifier: Feasible For Diversity of Speech Applications," in *2020 IEEE Student Conference on Research and Development (SCOReD)*, Batu Pahat, Malaysia, 2020, pp. 300-305.