

Phase 3 Implementation Report

Personalization Strategies for Non-IID Federated Learning

Muhammad Ibrahim Iqbal (27085)
Muhammad Maaz Siddiqui (27070)
Muhammad Ibrahim Farid (27098)

February 2026

1 Overview

Phase 3 investigates personalization strategies to address the heterogeneity challenges inherent in federated learning across non-IID client distributions. While Phase 1 demonstrated that vanilla FedAvg maintains near-centralized performance (99.19% vs 99.25%), and Phase 2 added blockchain provenance with minimal overhead, this phase explores whether client-specific model adaptation can further improve performance for underrepresented participants.

Two personalization approaches were implemented and evaluated: local fine-tuning and PerFedAvg meta-learning. Experimental results revealed that on the current small-scale dataset, both approaches failed to improve upon the baseline, demonstrating that the existing global model had already reached near-optimal performance. These findings validate the need for larger, more heterogeneous datasets to observe meaningful personalization benefits.

2 Motivation and Problem Context

Federated learning aggregates model updates from multiple clients to produce a single global model. This one-size-fits-all approach performs well when client data distributions are similar, but degrades when clients exhibit significant statistical heterogeneity. In healthcare, a cardiac specialty center observing predominantly arrhythmia cases has fundamentally different data characteristics than a general hospital treating mostly healthy patients.

Personalization addresses this mismatch by allowing each client to maintain a customized model variant tailored to its local data distribution. Rather than forcing all clients to converge toward a single global optimum, personalized federated learning seeks to find client-specific optima while still leveraging collaborative knowledge from the federation.

The motivation for Phase 3 stems from observed performance variance in Phase 1 results. While Clients 2 and 3 achieved 100% test accuracy, Client 1 lagged at 97.57%. This 2.43% gap suggests that the global model may be biased toward the majority distribution, underserving the minority participant. Personalization techniques aim to close this fairness gap by enabling Client 1 to adapt the global model to its unique cardiac specialty distribution.

3 Technical Approaches

3.1 Local Fine-Tuning

The simplest personalization strategy involves post-hoc adaptation of the global model to local data. After each federated aggregation round, clients download the updated global model and perform additional local training epochs exclusively on their private datasets.

This approach requires minimal algorithmic modification. The standard FedAvg training loop proceeds normally, but before evaluation, each client fine-tunes the global model for a small number of epochs (typically 3-5) using a reduced learning rate to avoid catastrophic forgetting. The fine-tuned model is used only for local inference and is not contributed back to the federation.

Mathematically, if θ^t represents the global model at round t and \mathcal{D}_i is client i 's local dataset, the personalized model θ_i^t is obtained via:

$$\theta_i^t = \theta^t - \eta \sum_{(x,y) \in \mathcal{D}_i} \nabla_{\theta} \mathcal{L}(f(x; \theta^t), y)$$

where η is a small learning rate (typically 10% of the original federated learning rate) and the gradient is computed over the client's local data. This process is repeated for several epochs to allow the model to specialize to local patterns.

The key advantage of fine-tuning is simplicity and stability. Since it operates independently after aggregation, it cannot interfere with the federated training dynamics. The disadvantage is lack of coordination: fine-tuning does not influence the global model evolution, meaning future rounds may continue to bias toward majority clients.

3.2 PerFedAvg: Meta-Learning for Fast Adaptation

PerFedAvg (Personalized Federated Averaging) takes a more sophisticated approach by explicitly optimizing the global model for rapid adaptability rather than direct performance. This meta-learning strategy trains a model that can quickly specialize to any client's distribution in just a few gradient steps.

The algorithm operates on two nested optimization loops. The inner loop simulates local adaptation by computing a virtual gradient step on a client's data batch. The outer loop updates the global model parameters based on how well the virtually adapted model performs on a second independent batch. This teaches the model to occupy a region of parameter space from which fast adaptation is possible.

Formally, for client i at round t , PerFedAvg performs:

Inner Loop (Virtual Adaptation):

$$\theta'_i = \theta^t - \alpha \nabla_{\theta} \mathcal{L}_i^{(1)}(\theta^t)$$

Outer Loop (Meta Update):

$$\theta^{t+1} = \theta^t - \beta \nabla_{\theta} \mathcal{L}_i^{(2)}(\theta'_i)$$

where $\mathcal{L}_i^{(1)}$ and $\mathcal{L}_i^{(2)}$ are losses computed on two disjoint data batches, α is the inner learning rate controlling adaptation speed, and β is the outer learning rate controlling meta-update magnitude. The key insight is that the outer gradient $\nabla_{\theta} \mathcal{L}_i^{(2)}(\theta'_i)$ depends on θ^t through the chain rule, creating a meta-gradient that optimizes for post-adaptation performance.

After federated aggregation of these meta-updated parameters, the resulting global model should be positioned such that a few quick gradient steps can efficiently specialize it to any client's distribution. At test time, clients perform rapid adaptation using the inner loop formula before inference.

The advantage of PerFedAvg is principled optimization for personalization during training rather than as an afterthought. The disadvantage is significant complexity: it requires careful hyperparameter tuning (α and β), doubles memory usage (storing both θ and θ'), and assumes sufficient batches per client to compute independent inner and outer gradients.

4 Implementation Details

Both personalization strategies were implemented as standalone training scripts that reuse the data pipeline and model architecture from Phases 1 and 2. The implementations maintain consistency with previous experiments by using the same random seeds, data partitioning, and evaluation protocols.

4.1 Fine-Tuning Configuration

The fine-tuning implementation follows the standard FedAvg training loop for 20 communication rounds with 5 local epochs per round. After each round, before evaluation, each client creates a copy of the global model and fine-tunes it for 3 additional epochs using a learning rate of 0.0001 (10% of the base 0.001 rate). The Adam optimizer is retained for consistency. Fine-tuned models are evaluated on held-out validation and test sets but are not aggregated back into the federation.

4.2 PerFedAvg Configuration

PerFedAvg replaces the standard local training step with the two-loop meta-learning procedure. Hyperparameters were set to $\alpha = 0.01$ (inner learning rate) and $\beta = 0.001$ (outer learning rate) based on values reported in the original PerFedAvg paper. Each local training epoch consists of iterating through data batches and applying the inner-outer update formula. After 5 local epochs, client models are aggregated using weighted averaging identical to standard FedAvg.

At test time, personalized models are created by taking the final global model and performing 3 rapid adaptation steps (inner loop updates only) on each client’s training data before evaluation on test sets.

5 Experimental Results

5.1 Fine-Tuning Performance

Table 1 presents the final test accuracy comparison between global and fine-tuned models across all three clients.

Client	Global Model	Fine-Tuned Model	Improvement
Client 1	97.57%	97.57%	+0.00%
Client 2	100.00%	100.00%	+0.00%
Client 3	100.00%	100.00%	+0.00%
Average	99.19%	99.19%	+0.00%

Table 1: Test accuracy: Global vs Fine-Tuned models

Fine-tuning produced zero measurable improvement across all clients. Client 1, which showed the largest gap from optimal performance in Phase 1, failed to benefit from local adaptation. Clients 2 and 3, already at 100% accuracy, predictably showed no change.

The F1-scores tell a similar story. Client 1’s F1-score remained at 0.662, indicating no improvement in rare-class detection. The overall average F1-score stayed constant at 0.887, confirming that fine-tuning added no value beyond the global model baseline.

5.2 PerFedAvg Performance

PerFedAvg results were significantly worse than both the global baseline and fine-tuning, as shown in Table 2.

Client	Global (Phase 1)	PerFedAvg	Change
Client 1	97.57%	92.31%	-5.26%
Client 2	100.00%	97.92%	-2.08%
Client 3	100.00%	86.67%	-13.33%
Average	99.19%	92.30%	-6.89%

Table 2: Test accuracy: FedAvg baseline vs PerFedAvg

The meta-learning approach degraded average test accuracy by 6.89 percentage points. Client 3 suffered the most severe drop (-13.33%), while even high-performing Client 2 regressed by 2.08%. F1-scores collapsed from an average of 0.887 to 0.426, indicating catastrophic failure in minority class detection.

6 Analysis and Discussion

6.1 Why Fine-Tuning Failed to Improve

The zero-gain fine-tuning results can be attributed to three factors:

1. Near-Optimal Baseline Performance. The global FedAvg model achieved 99.19% average accuracy with two clients at 100%. This ceiling effect leaves minimal room for improvement. Fine-tuning three additional epochs cannot extract performance gains when the model has already converged to near-perfect classification.

2. Insufficient Data Heterogeneity. With only three clients and relatively mild non-IID partitioning (Jensen-Shannon divergence ≈ 0.25), the global model already represents a reasonable compromise across distributions. There exists no strong bias toward any particular client that local adaptation could correct.

3. Small Dataset Scale. Client 1 has 1,148 training samples, Client 2 has 1,340, and Client 3 has 625. For a model with 308,485 parameters, these datasets are small enough that the global model, trained across 3,113 total samples, has already learned all meaningful patterns. Additional local epochs merely overfit to noise rather than discovering new structure.

6.2 Why PerFedAvg Severely Underperformed

The dramatic performance degradation under PerFedAvg stems from hyperparameter sensitivity and data scarcity:

1. Hyperparameter Mismatch. Meta-learning requires careful tuning of inner (α) and outer (β) learning rates. The values $\alpha = 0.01$ and $\beta = 0.001$ were borrowed from the original PerFedAvg paper but may be inappropriate for our ECG classification task and CNN-LSTM architecture. Without extensive hyperparameter search, the meta-update gradients likely pushed parameters into suboptimal regions.

2. Batch Insufficiency. PerFedAvg requires two independent batches per training step: one for the inner gradient and one for the outer meta-gradient. With batch size 32 and small client datasets, many training iterations exhausted available batches and resorted to reusing the same data for both loops. This violates the independence assumption and degrades meta-gradient quality.

3. Meta-Learning Data Requirements. Learning to learn requires observing many diverse tasks (clients) with substantial data per task. Our three-client setup with 625-1,340 samples each fails to meet this requirement. Meta-learning research typically assumes dozens of clients with thousands of samples, allowing the algorithm to discover transferable adaptation strategies.

6.3 Implications for Larger-Scale Experiments

These negative results are scientifically valuable because they establish boundary conditions for personalization effectiveness. The findings suggest that:

- Personalization techniques require non-trivial heterogeneity to demonstrate value. When the global model already performs well across clients, local adaptation offers no benefit.
- Fine-tuning is more robust than meta-learning for small-scale federated settings. While neither helped here, fine-tuning at least matched baseline performance rather than degrading it.
- PerFedAvg should be reserved for scenarios with at least 10+ clients and 5,000+ samples per client to justify its complexity and hyperparameter sensitivity.

When we scale to the full 48-record MIT-BIH dataset in future work, we expect more severe non-IID distributions across 5-10 clients, creating opportunities for personalization to shine. With 100,000+ total samples and deliberate construction of highly skewed client partitions (e.g., one client sees only rare arrhythmias), we anticipate fine-tuning improvements of 3-5% for minority clients and potentially successful PerFedAvg deployment after proper hyperparameter tuning.

7 Reproduction Instructions

To reproduce Phase 3 experiments, ensure Phases 1 and 2 are complete and the data pipeline is operational.

For fine-tuning experiments, execute `python src/train_personalized_finetuning.py`. The script trains a global FedAvg model for 20 rounds, then fine-tunes it locally for each client before final evaluation. Results are saved to `experiments/personalized_finetuning_results.pkl`.

For PerFedAvg experiments, execute `python src/train_perfedavg.py`. This replaces standard local training with the two-loop meta-learning procedure. Hyperparameters are hardcoded as $\alpha = 0.01$ and $\beta = 0.001$. Results are saved to `experiments/perfedavg_results.pkl`.

Both scripts display per-client accuracies after each round and print final test set evaluations with improvement metrics. Training time is approximately 5-7 minutes per script on CPU.

8 Limitations and Lessons Learned

The primary limitation of this phase is dataset scale. With only 6,318 total samples distributed across three clients, neither personalization technique had sufficient data diversity to demonstrate advantages. Additionally, the near-perfect baseline performance created a ceiling effect that obscured potential benefits.

PerFedAvg’s failure highlights the danger of applying advanced techniques without proper validation. Meta-learning introduces significant complexity (doubled computation, multiple hyperparameters, batch independence requirements) that only pays off in data-rich regimes. For resource-constrained federated learning with small clients, simpler approaches like fine-tuning are more appropriate.

The lesson learned is that negative results are as valuable as positive ones in research. By documenting why personalization failed on small datasets, we provide clear motivation for scaling experiments and set realistic expectations for when these techniques should be deployed.

9 Key Findings and Contributions

Phase 3 contributes a rigorous evaluation of two personalization strategies under realistic federated learning constraints:

1. **Fine-tuning preserves baseline performance** but offers no gains when the global model is already near-optimal. This establishes fine-tuning as a safe default for personalization that will not harm performance.
2. **PerFedAvg requires careful hyperparameter tuning and sufficient data.** Without both, it can severely underperform vanilla FedAvg. This finding serves as a cautionary tale for practitioners considering meta-learning in federated settings.
3. **Personalization benefits depend on heterogeneity severity.** Mild non-IID distributions with high baseline accuracy leave no room for local adaptation to help. Future experiments with larger datasets and deliberate client skew will better demonstrate personalization value.

These findings inform the design of Phase 4, where we will explore synthetic data generation to amplify class imbalance and create more challenging heterogeneity. They also justify the planned scale-up to 48 MIT-BIH records, which will provide the data volume and distribution diversity necessary for personalization techniques to fulfill their potential.

10 Next Steps: Phase 4 Preview

Phase 4 will address the class imbalance problem identified in Phase 1. While rare arrhythmias constitute less than 1% of the dataset, they represent the most clinically critical events. Synthetic data generation using TimeGAN will augment minority classes under blockchain governance, ensuring transparent auditing of artificial samples.

The combination of synthetic augmentation and personalization may yield synergistic benefits: synthetic data can reduce class imbalance, creating more balanced local distributions that personalization techniques can better exploit. This will be evaluated in the final integrated system combining all four phases.