Phase 1 Implementation Report

Blockchain-Orchestrated Personalized Federated Learning

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

February 2026

1 Overview

This document describes the implementation of Phase 1, which establishes the foundational data pipeline, model architecture, and baseline federated learning system. The primary objective was to demonstrate that federated learning can achieve near-centralized performance on heterogeneous clinical IoT data, setting the stage for blockchain integration and synthetic data augmentation in subsequent phases.

2 System Architecture

The implementation follows a modular design with three core components working in sequence. The data pipeline handles ECG signal acquisition and preprocessing, converting raw PhysioNet records into normalized heartbeat segments. The model layer implements a hybrid CNN-LSTM architecture optimized for time-series classification. The training layer supports both centralized and federated learning paradigms, enabling direct performance comparison.

All configuration parameters are centralized in `config.yaml`, allowing experimentation without code modification. The system uses PyTorch for model development and the Flower framework for federated learning orchestration.

3 Data Pipeline

Medical data was sourced from the MIT-BIH Arrhythmia Database via PhysioNet. We downloaded three representative ECG records totaling approximately 90 minutes of continuous monitoring data. Each record was sampled at 360 Hz and contains annotations marking individual heartbeats and their classifications.

The preprocessing stage applies bandpass filtering to remove noise artifacts, followed by z-score normalization to standardize signal amplitudes. Individual heartbeats are extracted by centering 360-sample windows on annotated R-peaks, yielding 6,318 labeled segments across five classes: Normal, Supraventricular, Ventricular, Fusion, and Unknown beats.

To simulate realistic federated scenarios, data was partitioned into three non-IID clients representing different clinical environments. Client 1 mimics a cardiac specialty center with 40% of data and elevated arrhythmia prevalence. Client 2 represents a general hospital with 35% of data and predominantly normal beats. Client 3 simulates an emergency department with 25% of data and the highest proportion of rare events. This heterogeneous distribution deliberately introduces the statistical challenges that federated learning must overcome.

4 Model Architecture

The classification model employs a hybrid CNN-LSTM architecture totaling 308,485 trainable parameters. Three convolutional blocks extract hierarchical spatial features from the ECG waveform, with channel dimensions progressing from 32 to 128. Each convolution is followed by batch normalization and max pooling, reducing the sequence length while enriching the feature representation.

The extracted features feed into a two-layer LSTM with 128 hidden units, capturing temporal dependencies across the heartbeat sequence. The final layers consist of fully connected networks with dropout regularization, mapping the LSTM output to class probabilities. This architecture balances expressive power with computational efficiency, making it suitable for deployment in resource-constrained federated settings.

5 Training Methodology

Two training paradigms were implemented for comparison. The centralized baseline pools all client data into a single training set, representing the ideal case where data sharing is permissible. Training proceeded for 50 epochs with a batch size of 32 and Adam optimization at a learning rate of 0.001.

The federated implementation uses the FedAvg algorithm across 20 communication rounds. In each round, all three clients train local copies of the global model for 5 epochs on their private data. Model updates are then aggregated using weighted averaging proportional to client dataset sizes, and the updated global model is redistributed. This process continues until convergence, with validation performed after each round.

6 Results

The centralized baseline achieved 99.25% test accuracy with an F1-score of 0.899, establishing the performance upper bound. FedAvg attained 99.19% average test accuracy with an F1-score of 0.887, demonstrating only a 0.06% performance degradation despite the non-IID data distribution. Individual client accuracies ranged from 97.6% to 100%, with Client 1 showing slightly lower performance due to its higher class imbalance.

These results validate that federated learning can maintain near-centralized performance when the number of clients is small and data quality is high. The minimal accuracy gap also suggests that the current non-IID distribution, while heterogeneous, is not severe enough to dramatically hinder model convergence. This establishes a strong baseline for evaluating improvements from blockchain governance and synthetic augmentation in later phases.

7 Reproduction Instructions

The complete implementation can be executed by following this workflow. After cloning the repository and activating the virtual environment, install dependencies using `pip install -r requirements.txt`.

Data acquisition begins with `python src/download_data.py`, which retrieves ECG records from PhysioNet and stores them locally. Next, `python src/preprocess_data.py` segments the signals into individual heartbeats and applies normalization. The command `python src/partition_data.py` then distributes these segments across three non-IID clients according to the configured distributions.

Model training follows a two-stage process. Execute `python src/train_centralized.py` to establish the centralized baseline, which completes in approximately 5-10 minutes on CPU.

Then run `python src/train_fedavg.py` to perform federated training, taking an additional 3-5 minutes. Both scripts save trained models and performance metrics to the experiments directory.

Finally, `python src/compare_results.py` generates comparative visualizations and prints a summary of key findings. The entire pipeline from data download to final results takes roughly 20-30 minutes on standard hardware.

8 Key Findings

This phase successfully demonstrated that federated learning can achieve performance comparable to centralized training on small-scale heterogeneous medical data. The modular codebase provides a solid foundation for integrating blockchain provenance tracking in Phase 2, implementing personalization strategies in Phase 3, and introducing synthetic data augmentation in Phase 4.

The minimal performance gap observed here establishes a challenging baseline for improvement. Future work will investigate whether blockchain-governed synthetic augmentation can enhance fairness for minority classes and whether personalization techniques can further reduce inter-client variance, particularly in scenarios with more severe non-IID conditions.