DTU Compute
Department of Applied Mathematics and Computer Science

Federated learning for tackling data heterogeneity in healthcare applications

Written by **Ditte Bjerrum Gilsfeldt** and **Lucia Han Lu**

Bachelor Thesis

Federated learning for tackling data heterogeneity in healthcare applications

Bachelor Thesis
07 February, 2025 to 13 June, 2025

By
Ditte Bjerrum Gilsfeldt (s210666)
Lucia Han Lu (s224215)

Supervisor
Sneha Das
Nirupam Gupta

Cover photo: AI-generated illustration by ChatGPT, 2025
Created for: Bachelor thesis on Federated Learning in Healthcare
GitHub Repository: https://github.com/DitteGilsfeldt/Federated_Learning_for_tackling_data_heterogeneity_in_healthcare_applications

Approval

Ditte Bjerrum Gilsfeldt (s210666)

.....

Signature

.....

Date

Lucia Han Lu (s224215)

.....

Signature

.....

Date

Acknowledgements

Sneha Das
Assistant Professor (Tenure track), Department of Applied Mathematics and Computer Science
Cognitive Systems
Statistics and Data Analysis

Nirupam Gupta
Assistant Professor (Tenure Track), Department of Computer Science DIKU
Machine Learning

Contents

Preface

This bachelor thesis, ”Federated Learning for tackling data heterogeneity in healthcare applications” is written by two sixth-semester bachelor students from the Technical University of Denmark (DTU), enrolled in the Artificial Intelligence and Data study line. This thesis of 15-ECTS, builds on our interest in machine learning and the healthcare sector, focusing on privacy-preserving machine learning methods in real-world healthcare data, and it began in February 2025 and culminates in a final hand-in and oral defense in June 2025.

The work in this thesis draws inspiration from our earlier project work, “Organ Failure Following Cardiac Surgery”, in which we explored how machine- and deep-learning models can predict postoperative morbidity and mortality. That work highlighted hospitals’ need to identify high-risk patients, sooner rather than later, to improve survival rates, allocate ICU resources appropriately, and support clinical decision-making. By shifting focus to federated learning, we now aim to preserve patient privacy and prevent data leakage before sensitive information is compromised while still leveraging cross-hospital data to develop robust, personalized prediction models.

Having experienced the challenges of working with centralized clinical datasets in that context, we now pivot to a federated learning paradigm, aiming to achieve comparable predictive power while preserving patient privacy across multiple institutions.

In this thesis, implementing such a framework introduced its own complex set of practical hurdles. These included transforming messy, unstructured records into meaningful inputs, carefully selecting and engineering features, and configuring each site to train locally without ever exposing raw patient information. These hands-on experiences reminded us how crucial it is to design federated learning solutions that are both technically sound and practically viable in clinical settings.

Finally, we extend our thanks to our supervisors, whose names appear in the acknowledgments. Their thoughtful guidance and unwavering support helped us navigate both the theoretical intricacies and practical hurdles of this work, ensuring that our research remained grounded in real clinical needs.

Abbreviations

CFL - Clustered Federated Learning

CNN - Convolutional Neural Network

DL - Deep Learning

DP - Differential Privacy

DTU - Technical University of Denmark

EHR - Electronic Health Record

FedAvg - Federated Averaging

FedProx - Federated Proximal Optimization

FL - Federated Learning

FO - First-Order

HF - Hessian-Free

ICU - Intensive Care Unit

IID - Independent and Identically Distributed

LSTM - Long Short-Term Memory

MAML - Model-Agnostic Meta-Learning

ML - Machine Learning

MLP - Multi-Layer Perceptrons

MTL - Multi-Task Learning

MOCHA - Multi-Objective Communication-efficient Federated Learning

Non-IID - Non-Independent and Identically Distributed

Per-FedAvg - Personalized Federated Averaging

PFL - Personalized Federated Learning

PR-AUC - Precision-Recall Area Under the Curve

ROC-AUC - Receiver Operating Characteristic Area Under the Curve

SDG - Stochastic Gradient Descent

SOTA - State-of-the-art

Abstract

This bachelor thesis investigates the potential of Personalized Federated Learning (PFL) to address data heterogeneity in real-world healthcare applications, focusing on model robustness and stability. While Federated Learning (FL) offers a privacy-preserving framework for decentralized training, its performance often degrades under non-Independent and Identically Distributed (non-IID) conditions. To evaluate personalization under realistic constraints, this study implements and compares two PFL strategies. First-Order Personalized Federated Averaging (FO Per-FedAvg) and a simplified Multi-Task Learning (MTL) approach inspired by Multi-Objective Communication-efficient Federated Learning (MOCHA). These are benchmarked against two standard global FL baselines, Federated Averaging (FedAvg) and Federated Proximal Optimization (FedProx), selected for their widespread use and relevance in evaluating personalization impact.

The experiments simulate a cross-silo FL environment using two real-world intensive care datasets, namely MIMIC-III and eICU. Global models were limited to six harmonized static features to reflect schema constraints across hospitals. In contrast, PFL models had access to hundreds of client-specific static and dynamic features. Each method was evaluated on a binary mortality prediction task using ROC-AUC and PR-AUC, which are appropriate for imbalanced clinical data. Statistical robustness was assessed using the Wilcoxon signed-rank test across three independent random seeds per setup.

The model pipelines were validated by successful convergence on both IID and non-IID splits of MNIST, and a centralized LSTM + MLP model trained on full ICU data achieved near state-of-the-art (SOTA) ROC-AUC. These results confirm that model design and implementation were not limiting factors. However, despite richer feature access, neither the PFL method yielded statistically significant improvements over the global FL baselines. This underperformance was highlighted by a critical issue of prediction collapse in several models. Despite acceptable ROC-AUC values, some models collapsed into predicting only the minority class (mortality), resulting in near-zero accuracy, precision, and recall. This behavior was driven by a combination of extreme class imbalance, aggressive loss weighting, and the models' inability to produce well-calibrated predictions, which are all challenges that personalization alone could not overcome. Specifically, the use of `pos_weight` was too aggressive, leading the models to avoid predicting the majority class entirely.

In conclusion, this thesis demonstrates that federated personalization is not a plug-and-play solution for healthcare FL. Effective cross-institutional collaboration requires upstream solutions such as schema harmonization, curated feature engineering, and compatibility before personalization strategies can offer meaningful gains.

1 Introduction

Federated Learning (FL) is a decentralized approach to collaboratively training Machine Learning (ML) models across multiple institutions without sharing raw data. This preserves privacy and reduces security risks [22]. First formalized by McMahan *et al.* (2016), FL relies on the Federated Averaging (FedAvg) algorithm, which alternates between local model updates on each client's private data and a central aggregation step to refine a globally shared model [24]. FedAvg has been shown to match centralized performance on benchmarks such as MNIST and CIFAR-10, and language classification tasks, despite being trained on fragmented and decentralized datasets [11]. However, while FedAvg is designed to handle non-Independent and Identically Distributed (non-IID) data, its performance can degrade significantly under such conditions, often leading to substantial accuracy drops in real-world domains, like healthcare, when faced with extreme data skew [24]. For example, Zhao *et al.* (2018) have shown that one-class non-IID split can cause accuracy of FedAvg on CIFAR-10 to drop by 51.31% (from 80.83% to 29.52%), and similar drops by 11.31% on MNIST, and a 54.50% drop on a keyword-spotting task, while centralized Stochastic Gradient Descent (SGD) remains stable [24].

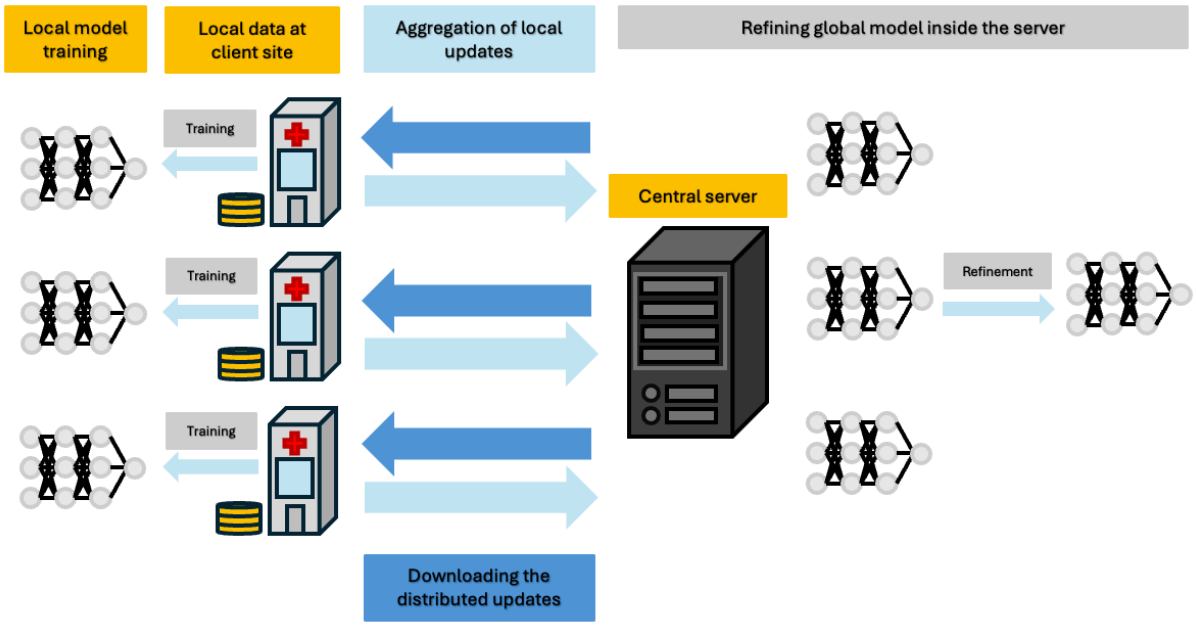


Figure 1.1: Cross-silo federated learning in a healthcare setting. (1) the central server initializes and selects a subset of hospitals. (2) each hospital trains the global model locally on its private healthcare data. (3) hospitals send only updated model weights back. (4) the server aggregates these updates to refine the shared model for the next round.

Figure 1.1 illustrates a typical cross-silo FL process. In this thesis, each client represents a hospital with its own electronically recorded health data. At each round, the central server selects a subset of participating hospitals and distributes the latest global model parameters. Each hospital then trains the model locally using its private data, updating model weights, and subsequently transmits only these updated weights back to the global server. The server aggregates the updated weights to refine the shared model and then redistributes it in the next round. This setup helps preserve patient privacy and reduces data-transfer costs, but its success is also highly dependent on how well the global model can accommodate highly heterogeneous, and often incomplete, healthcare data [8].

The next section reviews key challenges introduced by non-IID data distributions, missingness and class imbalance in healthcare FL, motivating the investigation into Personalized Federated Learning (PFL) techniques.

1.1 Challenges in Federated Learning for healthcare

FL applications in healthcare must effectively manage data heterogeneity. The patient data distribution varies across hospitals, as they have their own patient demographics, measurement protocols, and medical equipment [13]. This results in non-IID data with manifestations including feature shifts (such as different vital sign ranges), label skew (like the prevalence of specific diagnoses), and concept drift (for instance, different diagnostic criteria). One direct consequence of this is model bias. FedAvg uses a weighted-average aggregation to scale the influence of each client by data size, causing larger hospitals to dominate the global model and leaving smaller or specialized centers with suboptimal performance. Furthermore, if the weight updates diverge too much, the global model becomes harder to stabilize, reducing overall generalization performance [23].

Beyond statistical data heterogeneity, differences in computational resources, network connectivity issues, and privacy constraints can all lead to uneven client participation and contribution, further fragmenting the training process, which also degrades the overall model quality in FL settings [8].

Various modifications to FedAvg have been proposed to the issues arising from heterogeneous environments, with Federated Proximal Optimization (FedProx) being among the most prominent ones. FedProx constrains local updates with a proximal penalty, enhancing robustness to data skew and missing clients [11]. However, it still relies on a single global model and cannot adapt well to client-specific distributions. Consequently, more adaptive approaches that account for client-level variability while preserving data privacy have emerged. This, in turn, has led to the development of PFL, which aims to train models tailored to individual clients without compromising the core principles of FL.

1.2 The need for Personalized Federated Learning

Unlike standard FL methods like FedAvg and FedProx, which train a single global model that fails to generalize effectively to clients with skewed data, PFL introduces techniques that allow models to adapt to individual client distributions. This flexibility makes PFL particularly useful in healthcare settings, where patient demographics, diagnostic tools, and treatment protocols vary significantly [19].

PFL approaches aim to improve model adaptation by addressing key limitations of traditional FL. One of the most critical of such limitations is the divergence of local model updates leading to global model instability, and it is formally known as client drift. This occurs due to highly varying data distributions. By incorporating personalized update rules, PFL mitigates the impact of client drift and improves performance in non-IID environments [19].

Furthermore, several key strategies within PFL enable this personalization. Figures 1.2 and 1.3 illustrate two paradigms that this thesis focuses on. The first is meta-learning, which in theory trains a model to rapidly adapt [2] to each client using only a few local updates. While the implementation in this thesis uses a simplified first-order (FO) variant, this framework supports efficient personalization without needing client data exchange [4]. The second is multi-task learning (MTL), which treats each client's learning objective as a separate task and aims to improve local models by leveraging shared structure across clients when such structure is present [5]. Both paradigms offer distinct mechanisms for personalization under non-IID conditions and are explored further in the following chapters.

While other PFL methods exist, such as clustered FL (CFL), which groups clients with similar data distributions to train specific models for each group, most recent advances concentrate on flexible, adaptive frameworks like meta-learning and MTL [19]. By allowing models to adjust to highly skewed and incomplete healthcare datasets, PFL ensures that each institution can receive a version tailored to its needs while maintaining the privacy benefits of FL [10].

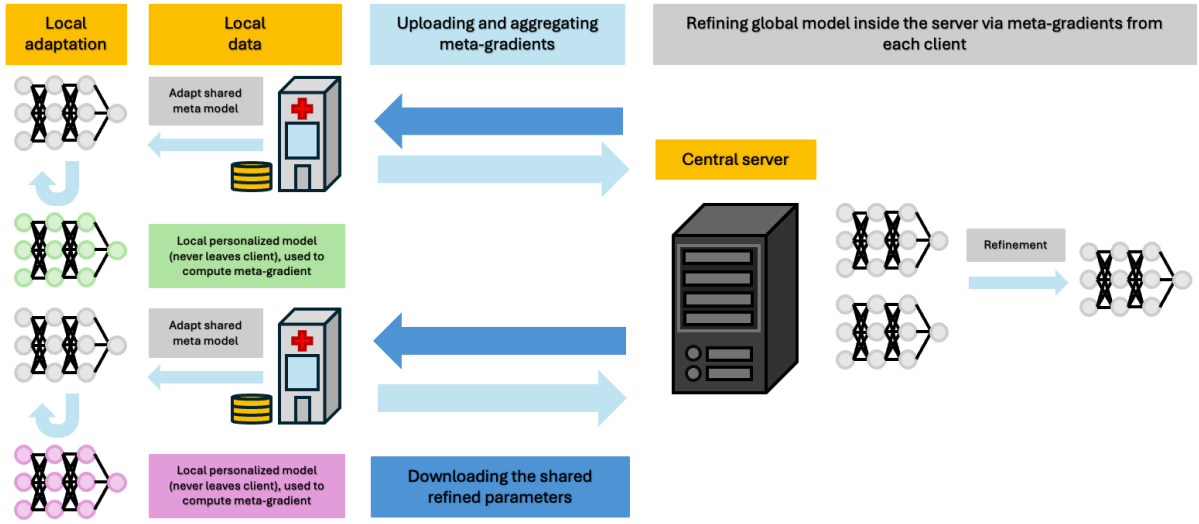


Figure 1.2: Meta-learning in FL. (1) the central server holds a shared initialization of a global meta parameter θ, then broadcasts θ to each client. (2) clients k fine-tunes θ locally to produce a personalized model θ_k. Here, they compute a meta-gradient, which is the difference between θ_k and shared θ. (3) clients upload the meta-gradients to the server. (4) the server aggregates all received gradients to refine the global θ. (5) the updated θ is sent back to clients for next round of adaptation.

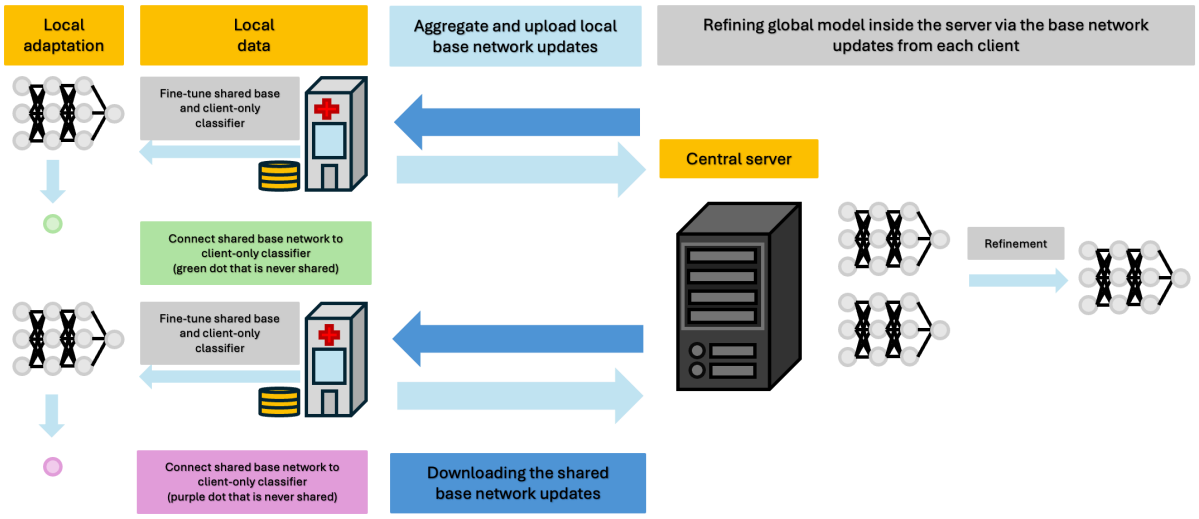


Figure 1.3: Multi-task in FL. (1) the central server has a shared base network and broadcasts it to clients. (2) each client connects the base network to a client-specific classifier (never shared) and trains both the base network and private classifier locally. (3) Each client uploads only the updates of the base network. (4) the server aggregates the updates to refine the globally shared base network. (5) the new updated base network is sent back to all clients for the next training round.

Therefore, the work in this thesis focuses on meta-learning and MTL approaches, while CFL is left to future work.

1.3 Motivation for this thesis

While PFL presents a promising solution to FL challenges, its real-world applicability in decentralized healthcare remains limited. Even though existing methods show improvements in handling heterogeneous benchmark datasets [13], questions persist about their effectiveness, scalability, and consistency across diverse medical institutions [13].

This thesis focuses on bridging these gaps by exploring PFL, a paradigm designed to improve model adaptation for individual clients while preserving the benefits of FL [19].

Specifically, it analyzes a FO implementation of meta-learning and a MTL strategy, represented by a FO Personalized Federated Averaging (Per-FedAvg) and a Multi-Objective Communication-efficient Federated Learning (MOCHA) inspired architecture to evaluate their robustness in highly heterogeneous healthcare environments.

While FO Per-FedAvg draws on the meta-learning framework introduced by Fallah et al. [4], it omits the full inner-loop adaptation step and higher-order gradients for computational simplicity. Similarly, the MOCHA-inspired model does not include the dual optimization or learned task covariance components, but retains the key principle of enabling client-specific models coordinated through a shared representation.

This approach allows the evaluation of core personalization ideas under more practical and constrained setups. It also enables testing coordination between clients in the style of MOCHA, even in cases where its structural assumptions (known task relationships) may not hold. This is a regime underexplored in prior work but common in real-world cross-silo deployments.

By exploring these techniques, the thesis aims to identify strategies that improve model stability, optimize performance across hospitals with varying data, and address key trade-offs between personalization and generalization. This leads to the formulation of the central research question.

1.4 Research questions

The basis for analysis of this thesis is formulated through the following research question: ”How can Personalized Federated Learning (PFL) improve model robustness and stability when handling heterogeneous and missing data in healthcare?”. The analysis of the research question will be answered through the sub-questions listed below:

- To what extent can Personalized Federated Learning (PFL) models, which leverage client-specific dynamic data, improve predictive performance compared to global Federated Learning (FL) models restricted to a common set of static features?
- How effectively can PFL recover client-specific performance on medically heterogeneous datasets (MIMIC-III and eICU), and is this improvement statistically significant when compared to the generalization capability of a single global model?
- What are the empirical trade-offs between PFL and traditional FL in terms of communication efficiency and convergence speed?

1.5 Contributions of this research

This thesis focuses on evaluating FL and PFL strategies for handling heterogeneous and incomplete healthcare data in realistic cross-silo settings. In order to answer the research questions, the follow key contributions have been made:

- A PyTorch-based FL pipeline implementing FedAvg and FedProx, constrained to a small, shared static feature set common to both the MIMIC-III and eICU datasets each as a client.

- A PyTorch-based PFL pipeline that integrates each client's dynamic time-series via a local LSTM encoder with a globally shared Multi-Layer Perceptrons (MLP) head on static features, enabling personalization without raw data exchange.
- A centralized benchmark combining the same LSTM and MLP architecture on the full preprocessed feature set MIMIC-III and eICU data, with both static- and dynamic data, achieving state of the art (SOTA) ROC-AUC for Intensive Care Unit (ICU) mortality prediction.
- A static-only centralized benchmark using an MLP on each of MIMIC-III and eICU static feature sets to quantify the performance gap under strict feature matching.
- A cross-silo FL validation on the MNIST dataset, reusing the architecture of the shared-feature MLP and FL pipeline to achieve SOTA federated results and to demonstrate generalizability.
- An experimental evaluation on MIMIC-III and eICU under missingness, label imbalance, and cross-silo heterogeneity, comparing the robustness and convergence of FL, PFL, and centralized models.
- A comparative analysis of model stability, predictive performance, and the trade-offs between personalization and generalization in healthcare FL.

This thesis focuses on structured ICU data, with all experiments conducted on preprocessed representations of static and dynamic patient records. Finally, privacy-enhancing technologies such as differential privacy are acknowledged but not implemented.

For the remainder of this thesis, Chapter 2 surveys related work, Chapter 3 describes methodology, Chapter 4 explains the experimental setup, Chapter 5 presents results, Chapter 6 discusses implications, and finally, Chapter 7 concludes and outlines future directions.

2 State of the art

Building on the challenges introduced in Chapter 1, this chapter reviews relevant literature in FL and PFL, focusing on methods applicable to cross-silo healthcare environments. Specifically, it surveys baseline global models (FedAvg and FedProx), then reviews two PFL paradigms (Per-FedAvg and MOCHA) that are implemented in this thesis. Finally, it compares the theoretical trade-offs between global and personalized methods and outlines strategies for managing healthcare data heterogeneity and missingness.

2.1 Federated Learning algorithms as baselines

In order to establish a solid reference point for personalization, this section briefly describes the two baseline FL methods evaluated in this thesis. First, FedAvg and then FedProx, as both algorithms aim to train a single global model across decentralized clients while preserving privacy. But these two algorithms differ in their robustness to non-IID data and system heterogeneity.

2.1.1 FedAvg algorithm

The FedAvg algorithm was introduced by McMahan *et al.* in 2016, as mentioned in Chapter 1. It trains a shared model by averaging locally updated weights from each client in each communication round. Clients perform local optimization on their private datasets and send updated models to a central server, which aggregates them proportionally to local data sizes.

While FedAvg performs well on IID data and achieves accuracy near centralized models on benchmark datasets like MNIST and CIFAR-10, its performance degrades under non-IID conditions, exhibiting slower convergence and reduced generalization [11]. These limitations are particularly pronounced in domains like healthcare, where local datasets vary significantly across institutions.

To mitigate these issues, Li *et al.* (2020) proposed FedProx, which modifies FedAvg by introducing a proximal term to the local objective. This regularization discourages client models from drifting too far from the global model, improving stability in the presence of statistical and system heterogeneity.

2.1.2 The FedProx method

Li *et al.* (2020) proposed FedProx to address both heterogeneity in systems, where clients perform varying amounts of local work, as well as heterogeneous data distributions. FedProx modifies the FedAvg algorithm by adding a proximal term to each client's objective to penalize large deviations from the global model [9]. This discourages local updates from drifting too far, making the method more stable under skewed data and partial client participation.

In FedProx, when the proximal term is zero, the algorithm reduces to FedAvg. In heterogeneous settings, FedProx has shown up to a 22% test accuracy improvement over FedAvg. In this thesis, both FedAvg and FedProx are implemented using shared static features, providing a consistent baseline for evaluating the more adaptive PFL approaches.

2.2 Personalized Learning techniques

While global FL methods such as FedAvg and FedProx enable decentralized model training, they often perform poorly in the presence of significant data heterogeneity. This limitation is especially relevant in healthcare, where differences in patient populations, clinical practices, and disease prevalence vary across institutions. To address these challenges, PFL techniques adapt models to individual clients while maintaining the privacy guarantees of FL.

This section examines two PFL methods implemented in this thesis. FO Per-FedAvg, which optimizes for rapid adaptation to new clients, and MTL, which trains personalized models by capturing inter-client relationships.

2.2.1 Meta-learning: FO Per-FedAvg

Meta-learning in FL aims to optimize a model that can quickly adapt to each client's local distribution with minimal training. Fallah *et al.* proposed Per-FedAvg, a meta-learning extension of FedAvg built on the model-agnostic meta-learning (MAML) framework [4]. The objective is to learn a shared initialization that enables efficient personalization through a small number of local gradient steps. This is particularly relevant in cross-silo healthcare settings where data distributions differ significantly across institutions. Unlike global models that generalize poorly under such heterogeneity, Per-FedAvg improves local performance by optimizing for adaptability [4].

The original formulation of Per-FedAvg relies on computing meta-gradients, but computing these involves second-order derivatives (for instance, Hessian-vector products), which are computationally expensive. To address this, the authors propose two approximations:

- First-Order Per-FedAvg (FO) that omits all second-order terms, improving efficiency and scalability.
- Hessian-Free Per-FedAvg (HF) that approximates second-order terms using finite differences, balancing accuracy and cost.

This thesis implements the FO variant due to computational constraints. The implementation approximates the meta-gradient using FO updates without full meta-batch splitting or exhaustive hyperparameter tuning. Preliminary experiments revealed instability on real-world healthcare data, likely caused by the sensitivity of FO Per-FedAvg to local step size and heterogeneity. These challenges are further discussed in Chapter 6.

2.2.2 Multi-task learning: MOCHA

MTL addresses statistical heterogeneity by learning a distinct model for each client while capturing shared structure between tasks. This is highly relevant in healthcare, where institutions differ in feature availability, label space, and population characteristics. MTL approaches can be split into two categories. (1) those assuming known relationships among tasks, and (2) those that learn task relationships from data [18]. Here, (2) is more suitable for FL, where inter-client relationships are unknown and data is highly non-IID.

Smith *et al.* proposed MOCHA, a federated MTL framework that decomposes the overall problem into client-specific subproblems using dual decomposition. This enables local optimization while coordinating updates through a shared task relationship matrix. MOCHA is designed to be robust under system heterogeneity, straggler clients, and partial participation [18].

In this thesis, a simplified version of MOCHA is implemented. The full dual decomposition and learned task relationships via the task relationship matrix are omitted due to complexity and scalability concerns. Instead, the core idea is preserved, which is to have a shared backbone model that is trained across clients, while each client maintains a private output head. This setup enables structured personalization with minimal overhead and aligns with real-world constraints in healthcare settings.

2.3 Literature-based comparison of personalized FL paradigms

The two dominant paradigms in PFL, namely Meta-learning and MTL each have their unique strengths, trade-offs, and use different techniques to accomplish personalization, and both share the common goal of improving performance under non-IID settings.

The comparison here, drawing on results directly reported in the original Per-FedAvg [4] and MOCHA [18] papers, aims to analyze and review the theoretical trade-offs and practical strengths of each approach across different scenarios, rather than replicating empirical benchmarks of the full algorithms (which this thesis explores through simplified and scalable approximations, as clarified earlier).

The comparison will cover the following points:

1. Adaptation speed versus long-term performance.
2. Communication and computation trade-offs.
3. Robustness under extreme non-IID splits.
4. Privacy and security considerations.

2.3.1 Adaptation speed versus long-term performance

One key distinction between PFL approaches lies in how quickly they deliver effective personalization versus the performance they achieve after full convergence.

Per-FedAvg (especially the HF variant) is optimized for rapid adaptation. In the pathological CIFAR-10 splits studied by Fallah *et al.* [4], HF Per-FedAvg reaches roughly 71.25% test accuracy after a single local update, while FedAvg is around 58.59%. Meanwhile, FO is computationally cheaper, but performance drops significantly for larger learning rates [4]. The FO variant used in this thesis, due to computational constraints, is more efficient but exhibits degraded performance under larger learning rates [4].

MOCHA produces fully personalized weights for each client at the end of federated training with no additional fine-tuning required. On highly non-IID benchmarks like sensor-vehicle benchmark in [18], MOCHA achieves per-task accuracy of 5–10% higher than in FedAvg after convergence, but it typically needs more rounds than HF in Per-FedAvg with the one-step fine-tune.

To summarize:

- FO Per-FedAvg (but especially the HF variant) is better for rapid, one-step personalization (for instance, when new clients join the federated process).
- MOCHA is better suited for long-term convergence without post-training adaptation.

2.3.2 Communication and computation trade-offs

Efficient communication and manageable computational load are essential for real-world FL deployment. The two paradigms take fundamentally different approaches to this trade-off.

Per-FedAvg inherits FedAvg's communication scheme where each client downloads a global parameter vector and uploads an updated version after local training. The FO variant used in this thesis introduces minimal computational overhead, as it requires one additional gradient evaluation per local step compared to standard FedAvg, due to the inner adaptation step. The HF variant, however, has a higher computational burden by requiring the approximation of second-order information (Hessian-vector products). Fallah *et al.* describe HF as "computationally costly", whereas FO is preferred for its "computational efficiency" and "scalability" [4].

In contrast, MOCHA involves clients exchanging updates related to dual variables, which are used by the central server to coordinate optimization via a shared task relationship matrix. This matrix may be predefined or learned during training, influencing how local subproblems are structured. Rather than performing a fixed number of SGD steps, each client solves a small convex subproblem, typically a quadratic approximation in each round. This enables MOCHA to reduce communication frequency.

Smith *et al.* report that MOCHA reaches a target accuracy in approximately 100 seconds under network constraints, while minibatch SGD requires closer to 1000 seconds for a comparable problem size [18].

To summarize

- FO Per-FedAvg offers low compute and communication overhead, making it suitable for resource-constrained clients with stable connectivity.
- MOCHA needs more local computation but it is communication-efficient, which benefits clients with limited communication capacity but higher computational capacity.

2.3.3 Robustness under extreme non-IID splits

In practice, FL systems often face extreme heterogeneity where clients hold highly skewed or disjoint data. Robustness under these conditions is a key differentiator between PFL methods.

The meta-objective of Per-FedAvg assumes bounded variation in client-specific losses, as formalized in Assumption 5 of [4]. When this condition is violated, such as under extreme non-IID splits, the FO variant degrades significantly in performance, particularly at larger local learning rates. For instance, Fallah *et al.* report that FO achieves only 35% accuracy on a pathological CIFAR-10 split, while the HF variant maintains 71.25% accuracy after a single local step [4].

While this thesis implements the FO version for efficiency, these results highlight the sensitivity of Per-FedAvg to data skew and hyperparameter choice. Furthermore, if a new client's feature space is highly disjoint from existing clients, even HF may require careful tuning to remain stable.

In contrast, MOCHA explicitly models task relationships through the matrix, enabling inter-client information sharing even when data distributions have little or no overlap. Smith *et al.* demonstrate convergence under mild straggler assumptions and weak convexity, even with non-smooth local loss functions [18].

On highly heterogeneous datasets like Human Activity Recognition, MOCHA consistently outperforms global baselines and maintains stability when clients lack entire label classes. For example, MOCHA achieves 1.77% higher accuracy than a global model under such conditions [18]. While the gain is modest, it reflects consistent per-client improvements and enhanced stability across heterogeneous tasks.

To summarize

- Under moderate non-IID skew, FO Per-FedAvg (but particularly HF) adapts rapidly but is sensitive to hyperparameters and feature shift.
- MOCHA is more robust to extreme label or feature distribution shifts, assuming task relationships are present and can be learned via the task relationship matrix.

2.3.4 Privacy and security considerations

While privacy is a core motivation for FL, PFL methods differ in how much information they expose during training.

Per-FedAvg follows the same communication pattern as FedAvg, which is sharing model parameters and not raw data. This is compatible with standard privacy-preserving mechanisms such as secure aggregation and differential privacy. However, as with other FL methods, Per-FedAvg remains vulnerable to inference attacks that exploit shared model updates, such as model inversion or gradient leakage [8]. Fallah *et al.* do not explore any privacy-specific extensions.

MOCHA, in contrast, involves additional communication of dual variables and a task relationship matrix, which encodes structural similarities between clients. While the authors claim that MOCHA preserves standard FL privacy guarantees that the task relationship matrix could reveal sensitive inter-client relationships (for instance, demographic or disease profile similarities across hospitals) if not adequately protected.

Importantly, the simplified MOCHA implementation in this thesis omits both dual updates and task relationship matrix, avoiding these structural privacy risks. Thus, it is comparable to FedAvg and Per-FedAvg in terms of data exposure, and suitable for deployment in privacy-sensitive healthcare settings without additional protections.

To summarize

- Per-FedAvg does not increase privacy risk over FedAvg and is compatible with existing privacy-enhancing protocols.
- MOCHA may expose inter-client structural information through task relationship matrix, unless mitigated with additional techniques. This risk is avoided in the simplified version implemented here.

2.4 Managing data heterogeneity and missingness in healthcare data for FL

Data heterogeneity and missing values recur throughout this thesis as key obstacles in practical FL deployments in healthcare. Both factors can compromise the stability, convergence and fairness of FL models. This section describes strategies for preparing complex electronic health record (EHR) data for FL, drawing inspiration from the principles and methods presented in the MIMIC-III-Extract pipeline by Wang *et al.* [20]. This paper offers a standardized approach to transform raw healthcare data into usable formats for prediction, addressing many of the challenges that come with heterogeneity and missingness.

2.4.1 Addressing data heterogeneity with MIMIC-III-Extract

Variations in recording practices, equipment and patient populations causes heterogeneity in raw EHR data. Here, MIMIC-III-Extract have implemented several steps to preprocess the raw data to have more robust and consistent features.

Wang *et al.* performs standardization of measurements with, for instance, unit conversions for vital signs laboratory measures so that the units across different features are consistent. They also do semantic grouping of features, where they manually group together raw item IDs (referred to as ”ItemID”) that mean the same thing into a single aggregated feature. For example, ”HeartRate” might be recorded under different itemIDs across different EHR system versions (CareVue versus MetaVision). Hence, by combining all those ItemIDs into one “HeartRate” feature, the data becomes more consistent [20]. The authors also mention that these aggregate representations reduce overall data missingness and the presence of duplicate measures [20].

Finally, there is consistent cohort definition, where defined criteria for cohort selection is applied (like adult patients, Intensive Care Unit (ICU) stays at minimum and maximum length of stay). This is done to establish a foundational dataset for further analyses [20].

2.4.2 Tackling missingness and overall data quality

MIMIC-III-Extract incorporates several techniques in order to manage missingness and improve data quality. For instance, reduction through aggregation is done via semantic grouping of features into clinical aggregates. This also serves to reduce overall data missingness by combining information from related ItemIDs [20].

Outlier detection and handling is also performed, since the authors employ clinically validated ranges to detect and handle outliers in vital signs and other laboratory measurements.

Values falling outside extreme thresholds are treated as missing, while other non-physiological values are replaced with the closest realistic value. Handling outliers in this way helps keep the data accurate and reliable [20].

Furthermore, MIMIC-Extract allows for the exclusion of time-varying variables that exceed a predefined threshold of missing values, ensuring a minimum level of data presence of selected features. The authors, for their own benchmark tasks, apply further preprocessing which includes imputation.

Their strategy involves representing each variable with a missingness mask, the imputed value, and the time since the last observation, using techniques like forward filling followed by patient specific or global means for remaining gaps [20]. The result of all the mentioned steps is a data output that can be prepared for model training.

2.4.3 How data preprocessing of this thesis relates to MIMIC-III-Extract

MIMIC-III-Extract shows how crucial careful preprocessing is when working with complex EHR data in a federated setting. While Wang *et al.* provide a robust solution including imputation strategies for their own benchmark tasks, the work in this thesis will make different design choices, driven by computational constraints and FL compatibility requirements in order to balance data representativeness against practical feasibility.

When adapting comprehensive preprocessing pipelines for FL, it often necessitates careful consideration of trade-offs. For instance, decisions regarding data aggregation (such as the hourly buckets used in MIMIC-III-Extract), the complexity of feature engineering (like using ”clinical aggregates” versus simpler features), and the choice of missing data imputation methods must all account for the distributed nature of FL, client resource limitations, and privacy considerations [20]. The specific preprocessing pipeline and methodological choices adopted in this thesis to navigate these challenges are detailed in the upcoming Chapter 3.

Still, even with careful preprocessing, algorithmic assumptions made by many PFL techniques may not hold, like those presented in this Chapter. For example, both Per-FedAvg and MOCHA assume some degree of underlying task similarity across clients.

So, as this thesis explores, such assumptions are often violated in real-world healthcare settings, where institutions differ significantly in both feature spaces and label distributions. This motivates the empirical evaluation of PFL methods under such conditions.

3 Methodology

This chapter translates the research questions from the Introduction (Section 1.4) and theoretical foundations (Chapter 2) into a reproducible experimental framework.

It begins by justifying the use of Per-FedAvg and a MOCHA-inspired setup in addressing hospital-level heterogeneity. Then, the ICU feature extraction and preprocessing methods are outlined followed by federated simulations including client configuration, sampling, training schedulers and aggregation strategies. The selected model architectures (Long Short Term Memory (LSTM) for time-series vitals and feedforward Multi-Layer Perceptrons (MLP) for static variables) and associated optimization settings are then presented.

Finally, the evaluation protocol featuring AUC, precision, recall, and F1-score are defined. The methodology provides a step-by-step account of how FL and PFL experiments were implemented and measured under realistic healthcare constraints.

3.1 Rationale and theoretical foundations

Two PFL approaches are evaluated. First, FO Per-FedAvg, a FO meta-learning method, and second, a MTL setup inspired by MOCHA. Importantly, the MOCHA implementation used in this thesis does not include the original dual-variable updates or the task relationship matrix, as previously mentioned in Chapter 2. Instead, it adapts the architectural separation between shared and private model components of MOCHA, while simplifying the optimization process for practicality and reproducibility. This reduced version is referred to as MOCHA-inspired throughout.

Below, Table 3.1 summarizes the key characteristics of each FL and PFL method implemented in this thesis, highlighting their handling of heterogeneity.

Table 3.1: Summary of key differences between the FL and PFL methods used in this study

Method	Description
FedAvg	Baseline global FL method, all clients train a shared model on a common static feature set, no personalization.
FedProx	Extension of FedAvg that adds a proximal term to the local loss function, improving stability under data heterogeneity.
FO Per-FedAvg	FO meta-learning approach, clients personalize models using the full local feature set, including static and dynamic features.
MOCHA-inspired	MTL setup, clients share a global static-feature backbone while maintaining private heads for local static and dynamic features.

Standard FL methods like FedAvg and FedProx degrade under extreme data skew. For example, as noted in Chapter 1, FedAvg suffers a 54.5% accuracy drop on a one-class non-IID CIFAR-10 split. In contrast to this, Per-FedAvg and MOCHA produce personalized model weights per client task, which offers better alignment with local data distributions and improved fairness across institutions.

Moreover, healthcare environments also face real-world constraints such as unreliable client participation due to software or hardware failures.

The original MOCHA framework includes a dual optimization formulation that tolerates asynchronous and partial client updates. While this thesis does not implement that full optimization strategy, it adopts MOCHA's architectural principle of coordinating personalized heads via a shared global model (detailed Subsection in 3.7.4).

In addition to this, incomplete clinical records are generally common, particularly in ICU datasets like MIMIC-III (Section 2.4). Hence, in order to address this without introducing significant training overhead, a lightweight imputation strategy is applied locally at each client (detailed in Section 3.3).

Finally, while model updates are exchanged instead of raw data, privacy risks still remain. Techniques such as differential privacy and secure aggregation are acknowledged but not implemented in this work. Instead, this study enforces that raw EHR data remains local to each client, consistent with the cross-silo structure illustrated in Figure 1.1.

3.1.1 Connections to Research Questions and Prior Work

The selected methodology directly addresses the revised sub-questions introduced in Section 1.4.

Each component of the experimental setup is specifically designed to isolate and evaluate the contributions of personalization under conditions of heterogeneity, partial feature availability, and constrained communication budgets.

- **To what extent can Personalized Federated Learning (PFL) models, which leverage client-specific dynamic data, improve predictive performance compared to global Federated Learning (FL) models restricted to a common set of static features?**

  The PFL pipeline (Per-FedAvg, MOCHA-inspired) explicitly allows each client to incorporate local, dynamic features during training, which directly addresses the question of to what extent PFL models, by leveraging client-specific dynamic data, can improve predictive performance compared to global FL models restricted to a common set of static features.

  This is in contrast to the static-only global models (FedAvg, FedProx). This addition enables direct performance comparisons under equivalent communication constraints, while isolating the benefit of personalized adaptation to richer input modalities.

- **How effectively can PFL recover client-specific performance on medically heterogeneous datasets (MIMIC-III and eICU), and is this improvement statistically significant when compared to the generalization capability of a single global model?**

  The experiments are conducted across two distinct real-world hospital datasets with substantial structural and label distribution heterogeneity, specifically to determine how effectively PFL can recover client-specific performance on these medically heterogeneous datasets (MIMIC-III and eICU), and whether this improvement is statistically significant when compared to the generalization capability of a single global model.

  Per-client metrics (for instance, ROC AUC, PR AUC, F1) are logged throughout training to enable both qualitative analysis and formal statistical tests (like Wilcoxon signed-rank test) assessing the significance of the improvements of PFL relative to global baselines.

- **What are the empirical trade-offs between PFL and traditional FL in terms of communication efficiency and convergence speed?**

All methods are evaluated under a fixed budget of communication rounds to determine the empirical trade-offs between PFL and traditional FL in terms of communication efficiency and convergence speed. Round-wise performance metrics are recorded to analyze convergence behavior and communication efficiency. This design enables a practical assessment of the cost-performance dynamics introduced by personalization strategies.

This methodology not only aligns with established findings in PFL literature [18], but also pushes the evaluation onto real-world healthcare scenarios, where clients differ significantly in data structure, schema, and distribution.

3.2 Evaluation of framework design

To evaluate the robustness and personalization capacity of FL methods in heterogeneous healthcare settings, this study implements three independent training pipelines. The first one being centralized baselines, then global FL, and finally PFL. Figure 3.1 illustrates the overall structure of the pipeline.

The process begins with the ingestion of two real-world ICU datasets, MIMIC-III and eICU, which represent electronic health records from distinct clinical institutions, which are then partitioned into two clients, simulating separate hospitals. Each client performs local preprocessing including feature selection, imputation, and standardization. Following this, both centralized and federated training regimes are deployed for comparison, and the same procedures are applied in all training regimes to ensure fair comparison.

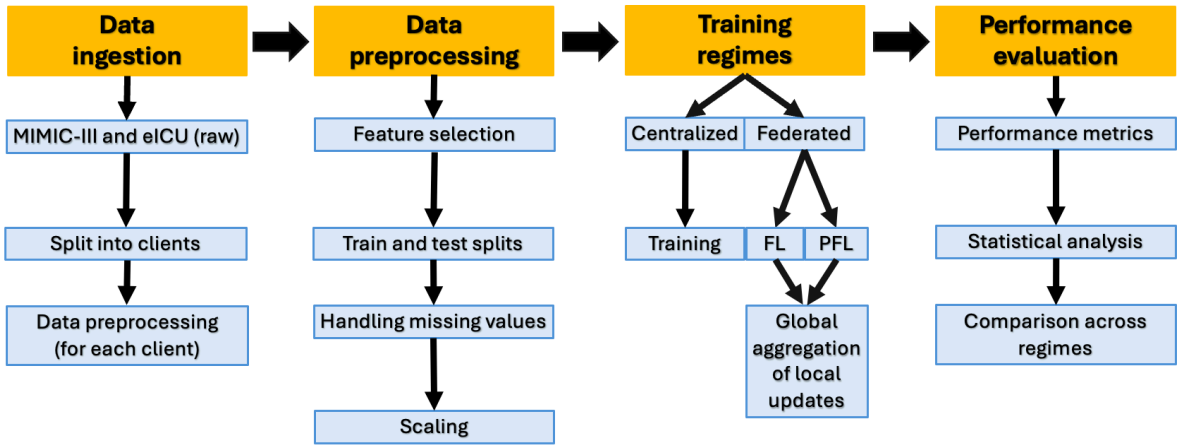


Figure 3.1: Flowchart of the experimental pipeline showing data ingestion of raw MIMIC-III and eICU data, preprocessing steps such as feature selection, handling missing values and scaling, training regimes that are mainly separated into centralized training and federated training. The final step is the performance evaluation.

This specifically includes centralized baselines trained on pooled data, using both the full feature sets and the limited set of common features from the ICU datasets (Section 4.1), along with a reference experiment on MNIST. Global FL methods (FedAvg and FedProx) applied to the simulated hospital splits and to MNIST for benchmarking. PFL methods are also implemented through FO Per-FedAvg and a MOCHA-inspired pipeline, evaluated on both the ICU datasets and MNIST.

Finally, each pipeline is evaluated using shared performance metrics to assess model effectiveness, robustness, and fairness across client-specific data distributions.

3.3 Data preparation

This section outlines the data preprocessing pipeline used to transform raw ICU datasets into structured, privacy-preserving inputs made suitable for federated model training. The procedures deviate from the SOTA methods presented in Section 2.4 in order to better reflect the constraints and objectives of this study, particularly regarding data heterogeneity and local client control.

3.3.1 Datasets and privacy

The MIMIC-III and eICU collaborative research datasets serve as the foundation for this study. Both datasets contain de-identified ICU stay data from multiple hospitals. MIMIC-III, collected from Beth Israel Deacon Medical Center [7], includes patient demographics, vital signs, laboratory results, medications, and clinical notes for a large cohort of patients. Similarly, the eICU Collaborative Research Database provides high-resolution data from various ICUs across the United States [15]. These datasets are representative of real-world clinical data, often exhibiting significant statistical heterogeneity and incompleteness across different care settings.

To simulate a realistic cross-silo FL setting, the full MIMIC-III dataset is assigned to Client 1, and the full eICU dataset to Client 2, representing two distinct institutional silos. Hence, this design reflects real-world federated deployments, where each client maintains their own independent infrastructure and collects data under different clinical protocols and population demographics.

Moreover, all data used in this study is fully de-identified to satisfy privacy constraints and comply with health data regulations [7]. Moreover, no raw data is exchanged between silos or with the central aggregator. Thus, this federated setup enforces local data control throughout the training processes.

3.3.2 Feature aggregation, selection, and preprocessing

This subsection details the preprocessing pipeline applied to transform raw ICU data into structured, privacy-preserving inputs for model training. The procedure is organized into two conceptual stages. (1) event-level hourly aggregation of ICU signals, and (2) feature engineering and imputation. Each stage is formalized below through pseudocode presented as Algorithm 1 and 2 to ensure clarity and reproducibility.

Algorithm 1 Hourly aggregation of ICU events

```
procedure Aggregate-Hourly-ICU-Events(csv_dict)
    Load selected .csv files into csv_dict.
    base_csv ← file containing ICU stay IDs.
    Set observation window to first 24 hours.
    Set bin size to 1 hour.
    for file in csv_dict do
        if file contains matching ICU stay IDs then
            Extract event timestamps.
            Filter events to within first 24h.
            Aggregate numeric features mean, std, min, max, count.
            Aggregate categorical features mode.
            Reshape to wide format one row per patient, hourly columns with _h0 to _h24.
            Merge into base_csv.
    return aggregated hourly features.
```

Algorithm 2 Preprocessing of ICU features

procedure Preprocess-ICU-features(aggregated_data)
    Drop features with $\geq$ 95% missing values.
    Filter cohorts:
    Patients aged $\geq$ 15, first ICU stay only, and with stay length between 12 hours and 10 days.
    Binary encode labels and categorical variables.
    Cyclically encode time features (like admission hour, weekday).
    Split into static and dynamic features: Dynamic features must appear at every hour
    Reshape dynamic features to $(N, H = 24, M)$.
    Impute missing values (mean) and apply standard scaling based on training data only.
    return processed static and dynamic arrays.

The preprocessing steps outlined above were applied independently per client, preserving local data control in alignment with FL constraints. While the pseudocode provides a formal overview, a few implementation notes should still be emphasized:

- All features were standardized using parameters fit on the training split only to prevent information leakage.
- Dynamic features were retained only if present at all 24 hourly bins, ensuring temporal consistency across the dataset.
- The final tensors used in training followed a shape of $(N, H{=}24, M)$ for dynamic inputs, and were saved separately for each client.

These steps ensured a consistent data representation across MIMIC-III and eICU clients while accommodating to their inherent statistical heterogeneity. Figure 3.2 summarizes the overall process visually.

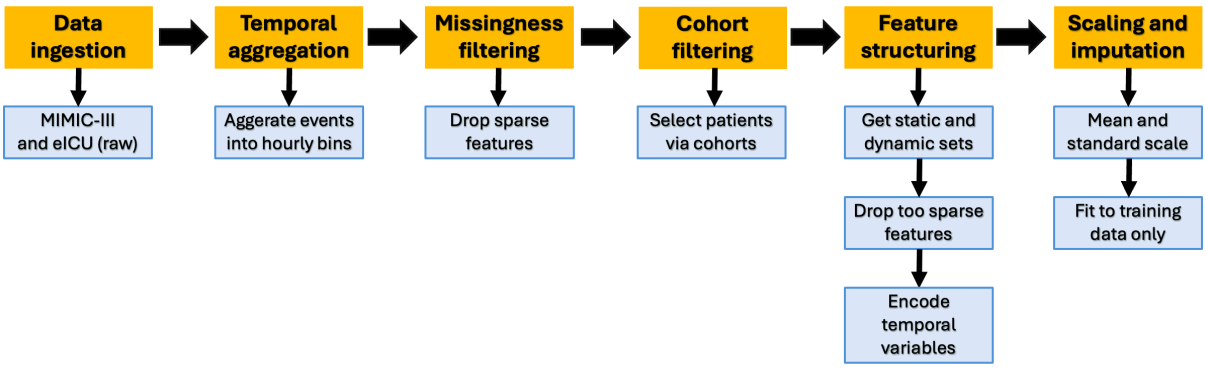


Figure 3.2: Overview of the preprocessing steps applied independently to the MIMIC-III and eICU datasets. The flowchart outlines each major step, including data ingestion, filtering of sparse features, cohort definition, temporal aggregation, static and dynamic feature structuring, and final scaling and imputation based on training data only.

3.3.3 Justification of preprocessing modifications that deviate from MIMIC-III-Extract

While the original MIMIC-III-Extract provides a valuable set of pre-defined clinical aggregates with good reasoning, the preprocessing in this thesis is performed at a more fundamental level.

Specifically, raw time-stamped data from ICU-related tables is aggregated directly, without semantic grouping, contrasting the MIMIC-III-Extract [20].

This approach increases flexibility for client-specific data handling and allows the same pipeline to be scaled across both MIMIC-III and eICU datasets in a federated context. Therefore, several deviations from MIMIC-III-Extract were made to suit this setup. First, tables that could not be joined using ID-identifiers were excluded in order to avoid alignment issues. Examples of tables that were excluded were admission records or clinical notes for MIMIC-III and nurse charting notes for eICU with additional details described in Section 4.1.

So, only tables that had consistent and timestamped ICU-level data were retained, and all retained table records were filtered to include only events from the first 24 hours of each ICU stay, consistent with the time-window strategy that MIMIC-III-Extract used.

Furthermore, the initial raw aggregation resulted in a high-dimensional and sparse feature space. To ensure computational feasibility for subsequent processing and model training, and to mitigate the risk of extremely sparse features slipping through and introducing noise rather than signal, features with more than 95% missingness were discarded. While MIMIC-III-Extract allows for configurable missingness threshold [20], the application in this thesis with a 95% cut-off reflects the active decision to focus on features with stronger presence, hence, optimizing the balance between potential information gain and the requirements of further data handling.

Finally, semantic feature grouping, as used in MIMIC-III-Extract to define interpretable aggregates such as heart rate or creatinine, was not applied here. Instead, features were aggregated directly from source tables without manually categorizing them into higher-level clinical concepts. This choice aimed to retain raw temporal and feature-level variation, allowing personalization models to potentially uncover data-driven patterns unique to each client. While this may reduce interpretability, it aligns with the objective of evaluating whether personalized FL models can adapt to unstructured and heterogeneous data settings without relying on expert-defined semantic groups.

3.4 Federated simulation setup

Before initiating federated training, the experimental environment must be enabled to simulate decentralized learning across distinct healthcare institutions.

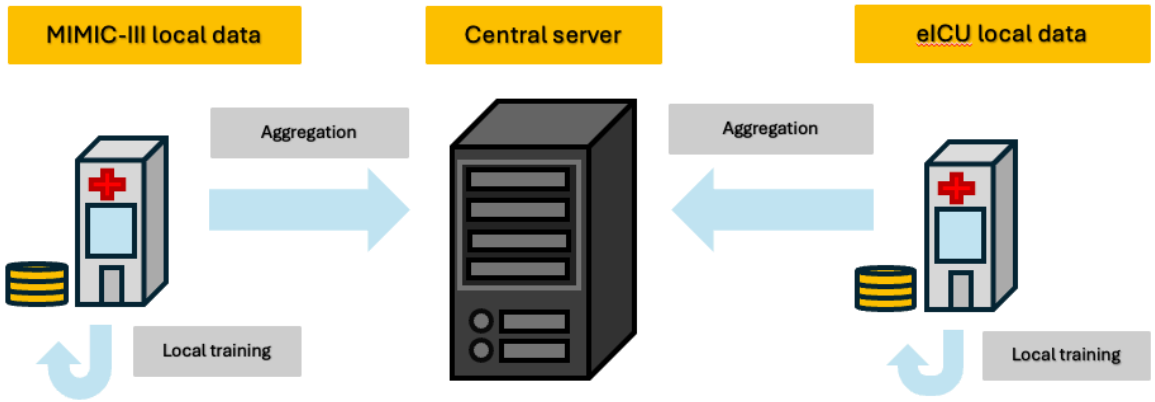


Figure 3.3: Cross-silo federated setup used in all FL and PFL experiments. The MIMIC-III and eICU datasets each represent an independent client. Local models are trained separately and only model updates are shared with the central server.

In this study, a cross-silo FL setup is adopted, where each client corresponds to a separate hospital with locally stored data to reflect real-world institutional heterogeneity (MIMIC-III as Client 1 and eICU as Client 2). This configuration is illustrated in Figure 3.3, which shows how these datasets act as independent clients communicating only model updates with a central server.

3.5 Federated Learning pipeline

To establish baseline performance under distributional shift, this study implements a standard cross-silo FL pipeline. As outlined in Section 2.1, this pipeline builds on the foundational algorithms FedAvg and FedProx, adapting them for deployment in realistic healthcare settings. Two hospital clients (MIMIC-III and eICU) participate in collaborative model training without exchanging raw data. The pipeline follows to the classical FL cycle:

- **Client initialization:** each client loads its local dataset independently, ensuring alignment on shared features and target labels.
- **Model broadcast:** the server sends the current global model to all clients at the start of each round.
- **Local training:** clients train a copy of the model locally on their datasets using Stochastic Gradient Descent (SDG).
- **Upload:** clients return updated model weights to the central server.
- **Federated aggregation:** the server aggregates updates using weighted averaging like FedAvg.
- **Evaluation:** the global model is validated on held-out client-specific test sets [8].

In this pipeline, only features that are aligned across both clients are used for model training, reducing the pipeline to only seeing a subset of static features. This is to ensure comparability and compliance with FL constraints, as it reflects the assumption that federated models must operate on features present at all institutions, as no raw data is exchanged and no global schema harmonization is enforced.

Furthermore, the global optimization objective follows the typical structure of FL where each client's contributions to the global model is weighted by the size of its local dataset. To address client-specific class imbalance, each client computes a local weighting term (`pos_weight`) applied during loss computation. This preserves privacy while avoiding explicit data resampling (more on `pos_weight` is explained in Section 4.3). The FL pipeline in this study includes exactly the following two algorithms:

- **FedAvg:** a baseline algorithm that aggregates client updates without any local constraints.
- **FedProx:** an extension of FedAvg that introduces a proximal term, penalizing large deviations from the global model to improve stability under non-IID conditions [9].

These methods were selected due to their simplicity, strong literature support, and relevance for evaluating baseline performance under data heterogeneity.

3.6 Personalized Federated Learning pipeline

To overcome the limitations of global FL models in heterogeneous clinical settings, this study implements a PFL pipeline. Unlike classic FL, where a single model is shared across all clients, PFL techniques introduce mechanisms for local adaptation, allowing each client to train a model better suited to its specific data distribution.

This setup aims to improve predictive performance under highly non-IID conditions with heterogeneous feature spaces and label skews. Unlike the global FL pipeline which is restricted to the six static features shared across both clients, the PFL setup allows each client to utilize its full local feature set, including both additional static variables and all dynamic time-series features. This reflects realistic healthcare scenarios where institutions differ in both data availability and features.

Therefore, in this pipeline, each client has access to both the shared static features and its own unique local features, and these include all dynamic and the remaining static features that are not among the shared ones. This design enables client-specific model construction while maintaining collaborative training over shared components. The PFL pipeline includes the following two personalization strategies:

- **FO-Per-FedAvg:** a meta-learning algorithm where each client performs adaptation on a support set, followed by evaluation on a query set to compute meta-gradients. These are aggregated server-side to update the shared initialization.

  This setup implements FO approximation of MAML, meaning it avoids second-order derivatives. Moreover, due to dataset heterogeneity, a full architectural alignment with SOTA paper [4] was not possible, meaning that only clients with similar input spaces could share meta-gradients. Moreover, due to complexity and computational constraints, meta-gradient validation and full hyperparameter optimization were limited. These limitations and constraint effects are all discussed in 6.3.

- **MTL:** a simplified variant of the MOCHA framework where each client trains a model composed of a private head and a shared static-feature backbone. Only the backbone is federated, while client-specific heads are retained locally.

  The full MOCHA optimization, including dual decomposition and the task relationship matrix, is not implemented. Instead, this setup captures MOCHA's core idea of decoupled yet coordinated personalization in a scalable form appropriate for real-world healthcare constraints.

Model architectures are adapted per client to fully leverage all of their local features. This reflects realistic healthcare deployment scenarios, where institutions often differ in data availability and recording practices. Unlike the baseline FL pipeline, which is restricted to globally shared features, the PFL pipeline supports heterogeneous input spaces.

The training and evaluation routines follow the same modular structure as the FL pipeline, with added logic for local adaptation and client-specific components. Here, aggregation is applied only to shared model parts when applicable (for instance, the static backbone in MOCHA-inspired MTL), while private components are excluded from synchronization. Further details on evaluation routines are explained in Section 3.8.

3.7 Model architectures

This section outlines the neural network architectures used across centralized, global federated, and personalized learning experiments. All models share a common architectural foundation to ensure comparability, while task-specific adaptations are introduced where it was necessary. Depending on the input type, models incorporate either recurrent or feedforward components.

Temporal EHR features are processed using LSTM layers, whereas static demographic variables and image inputs are handled through an MLP. These components are combined into modular architectures tailored to each experimental setup.

Each architecture is described in detail with rationale for its structure and the specific experimental conditions under which it was deployed and used.

3.7.1 LSTM + MLP for static and dynamic data.

This model architecture, visualized in Figure 3.4 is used for centralized learning on the full feature set of both MIMIC-III and eICU datasets in order to model both the dynamic time-series data alongside the static demographic features.

This model provides an upper-bound for performance, as it operates on the full feature set without any federated constraints such as data silos or shared feature limitations.

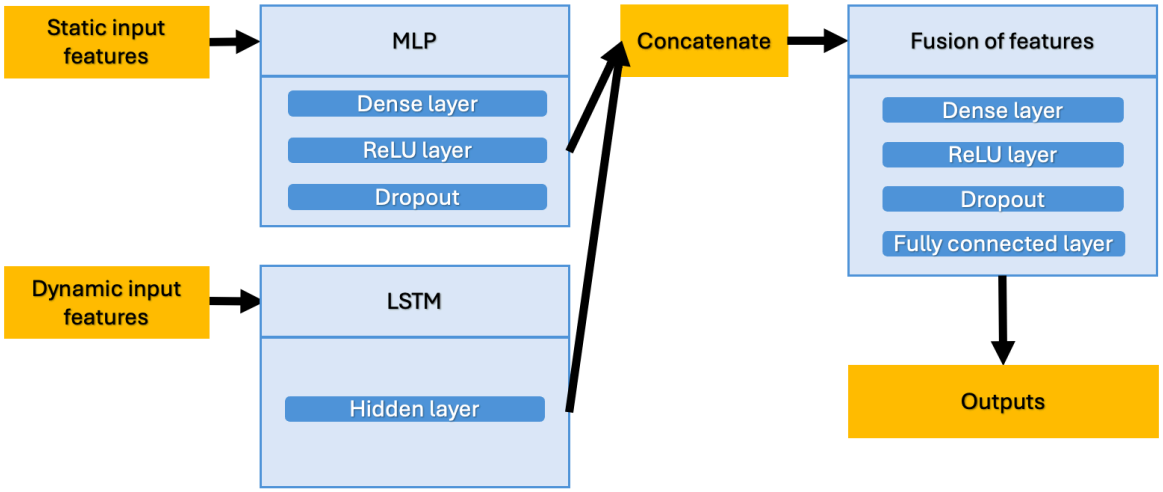


Figure 3.4: Illustration of the LSTM + MLP architecture used for modeling both dynamic and static input features. Static features are processed through an MLP block, while dynamic time-series features are handled by an LSTM block. Their outputs are concatenated and passed through a fusion block to produce the final prediction.

The model combines a recurrent layer with a feedforward network. First, dynamic features, structured as hourly time-series data across 24 hours, are encoded using a single-layer LSTM with a hidden size of 32. The static features are processed through a separate MLP consisting of a linear transformation to 32 units, followed by ReLU activation and a dropout rate of 0.6. The outputs from the LSTM (last hidden state) and the static MLP are concatenated.

This combined representation is then passed through a classification head, which includes an additional linear layer of 32 units with ReLU activation and a dropout rate of 0.5. The final layer is a linear transformation to a single logit for binary classification. Moreover, all layers are fully connected where applicable.

The LSTM + MLP architecture is inspired by prior work on temporal EHR model training with LSTM from [6], and it also follows common design patterns in clinical ML, where static and dynamic inputs are processed separately before they are fused together for training [21].

Though, this architecture is not always explicitly labeled as LSTM + MLP in other literature, its underlying structure is widely adopted for multi-modal healthcare modeling tasks involving both demographic and time-series inputs.

3.7.2 MLP for static data.

This model architecture, visualized in Figure 3.5, is used in both centralized and FL where only static features are aligned across clients.

The same architecture is used for both centralized and federated learning, enabling a direct comparison of predictive power across setups. It is used in the FL pipeline with FedAvg and FedProx experiments due to its architectural simplicity, which reduces communication overhead and improves generalization in feature-misaligned federated contexts, and ability to operate under feature-limited scenarios common in FL with heterogeneous datasets.

The model consists of a two-layered feedforward network. The first layer maps input features to a 256-dimensional hidden representation, followed by ReLU activation and a dropout of rate 0.4. This is then followed by another linear layer of 128 units, then ReLU and a dropout of rate 0.3. The final output layer is a linear transformation producing a single logit for binary classification. All layers are fully connected.

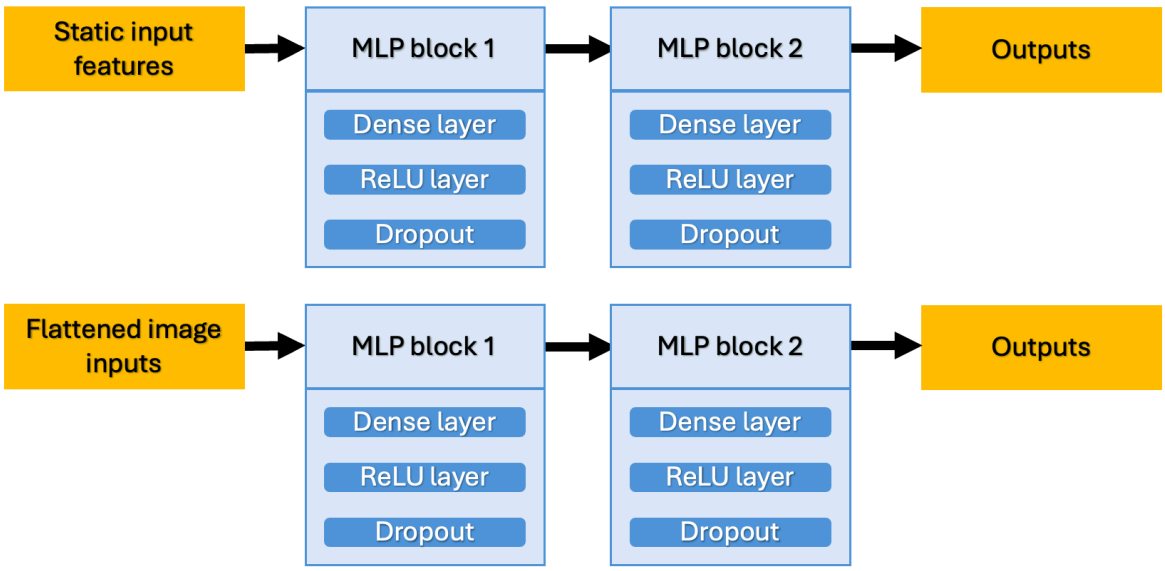


Figure 3.5: Visualization of the MLP architecture used for both static and image-based input data. The top row illustrates the model applied to static features, while the bottom row shows the same structure but applied to flattened image inputs. In both cases, the input passes through two MLP blocks with each consisting of the same structure of layers. Following these two blocks, the final output is then produced.

3.7.3 MLP for image data.

This model architecture, visualized in Figure 3.5, is used in both centralized and FL as a control model. Since it is trained on the MNIST data, known to be uniform. This setup provides a controlled environment for validating model convergence, isolating architectural or training issues from the effects of clinical data complexity

The architecture is identical to the MLP for static data, but adapted for image inputs. It consists of a two-layered feedforward network.

The input features is not flattened images and are mapped to a 256-dimensional hidden representation, followed by ReLU activation and a dropout of rate 0.4. The next layer is a linear transformation to 128 units, followed by ReLU and a dropout of rate 0.3. The final output layer is a linear transformation producing 10 logits for multi-class classification (corresponding to the 10 digit classes). Here, all layers are also fully connected.

3.7.4 LSTM + MLP model for static and dynamic data in personalized learning.

This model architecture, visualized in Figure 3.6, extends the centralized LSTM + MLP structure to support heterogeneous input spaces in the PFL setting. It consists of the modular components with LSTM to process dynamic, an MLP for static features, and a fusion head.

This architecture introduces both shared and client-specific modules. Specifically, a global MLP processes the common static features between clients, while local MLP and LSTM modules handle the private static and dynamic features, respectively. The private data refers to features that are not common across clients.

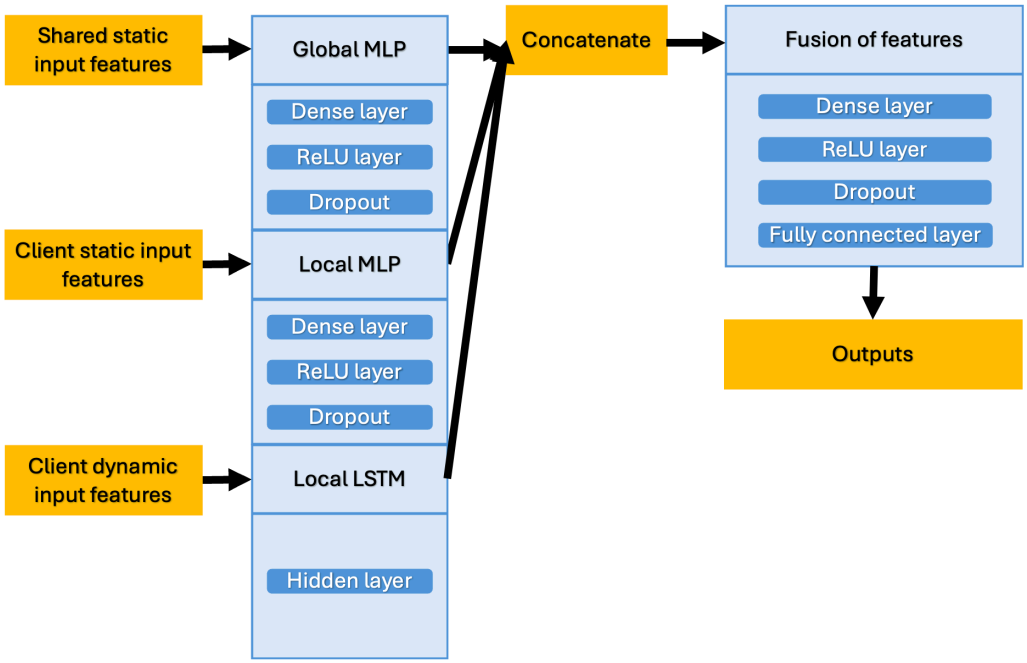


Figure 3.6: Visualization of the personalized LSTM + MLP architecture. Common static features are processed through a global MLP, while client-specific static and dynamic inputs are handled by a local MLP and LSTM, respectively. The resulting representations are concatenated and passed through a fusion block to produce the final prediction.

The LSTM used for dynamic features has a hidden size of 64. The local MLP for client-specific static features transforms to 32 units. The global MLP for common static features also transforms to 32 units. These outputs, along with the final hidden state of the LSTM, are concatenated. This combined representation is then passed to a fusion head, consisting of two linear layers with 64 and 32 units respectively, each followed by ReLU and dropout of rate 0.1, and a final linear layer to a single logit.

Furthermore, this modular design supports two distinct PFL strategies. In FO Per-FedAvg, each client receives a copy of a shared initialization encompassing all model components and performs local adaptation using meta-learning. However, because the local feature spaces differ, the way the model architectures are initialized vary with input dimensionality, hence, limiting structural alignment across clients.

In MOCHA-inspired setup, only the global MLP backbone is federated and aggregated across rounds, while client-specific LSTM and local static MLP remain private and unshared. Therefore, only the shared backbone is updated collaboratively, while private heads are trained independently.

This approach captures MOCHA's core idea of decoupled but coordinated personalization, however, as mentioned before, it does not use dual decomposition and task relationship matrix from MOCHA. So, this architecture implementation adopts a simplified version of MOCHA that maintains its core personalization principle while ensuring feasibility for deployment in real-world EHR settings, where strict model alignment is not practically sound. This architecture supports client-specific modeling under heterogeneous input conditions.

3.7.5 Connecting architecture consistency and theoretical assumptions

While all clients follow the same architectural layout in personalized learning, their specific models differ due to variations in input dimensions, particularly for dynamic time-series and local static features.

Consequently, client-specific modules such as the LSTM and local MLP layers have different parameter shapes and cannot be directly shared or aggregated. This violates an implicit but critical assumption in many PFL methods, such as FO Per-FedAvg [4], which require all clients to share an identical model architecture with fully aligned parameter shapes in order to perform meta-gradient aggregation.

In contrast, only the static MLP backbone is structurally consistent across clients, allowing limited parameter sharing in the MOCHA-inspired setup. The implications of this design limitation are further discussed in Subsections 6.3.3 and Section 6.4, especially how they relate to real world healthcare problems.

3.8 Evaluation strategy and metrics

Model evaluation depends heavily on the learning paradigm and the nature of the data used. In centralized training, evaluation is performed on a held-out test set drawn from the same distribution as the training set. This section covers how binary classification tasks involve ICU mortality prediction using EHR data, while multiclass classifications are evaluated on MNIST.

3.8.1 Centralized evaluation

In centralized training, models are evaluated on a hold-out test set drawn from the same data distribution as the training set. Table 3.2 presents the performance metrics used to evaluate the different models depending on what type of task they were trained on. For binary classification tasks, these will be performed on ICU mortality prediction using EHR data, and the multiclass classification tasks are performed on the image classification dataset MNIST.

3.8.2 Evaluation of Federated Learning

For FL, evaluation is performed locally on the test set of each client after every communication round. The global model is evaluated independently on the test data of each client, and results are then aggregated using a macro-average across clients. This approach follows the cross-silo FL assumption that central access to a shared or pooled evaluation dataset is not possible. In more detail, the global performance at round t is computed as:

$$\text{Global ROC-AUC}_t = \frac{1}{k} \sum_{i=1}^{k} \text{ROC-AUC}_{t,i} \tag{3.1}$$

Where K is the number of clients, and $\text{ROC-AUC}_{t,i}$ is the ROC-AUC of the global model evaluated on client i at round t. The same client-wise evaluation and macro-averaging is applied for PR-AUC, precision, recall, and F1-score.

Table 3.2: Evaluation metrics based on classification task type

Binary classification (ICU mortality)	**Multiclass classification (MNIST)**
Binary cross-entropy loss	Categorical cross-entropy loss
ROC-AUC	Macro-averaged ROC-AUC
Precision, recall	Macro-averaged precision, recall
F1-score	Macro-averaged F1-score
–	Classification accuracy

3.8.3 Evaluation of Personalized Federated Learning

For PFL, evaluation largely follows the same decentralized approach as in FL, where the model for each client is evaluated locally on their own test data after each communication round.

However, because the objective in PFL is not to optimize a single global model, but also to improve per-client performance under data heterogeneity, then additional considerations were also introduced.

Firstly, for FO Per-FedAvg, evaluation captures two key stages:

- **Pre-personalization performance:** measured directly on the shared initialization before local adaptation (the zero-shot).
- **Post-personalization performance:** measured after the client has done fine-tuning of the initialization locally for a small number of gradient steps.

These two evaluation stages were chosen, as they reflect the benefit of local adaptation. Both losses and classification metrics (exactly those presented in Table 3.2, for instance, ROC-AUC, PR-AUC, F1-score) are recorded client-wise to track personalization effects over training rounds. In addition to this, averaged scores across clients are also reported for consistency with standard FL evaluation, but the emphasis remains on individual performance.

Secondly, for the MOCHA-inspired approach, each client trains its own private head while sharing a common static-feature backbone. Since model parameters are not fully aligned across clients, there is no global model in the conventional sense. Evaluation is, thus, fully local and consists of computing each client's performance using its own personalized model.

Overall, for PFL, the evaluation strategy focuses on client-level improvements under non-IID conditions, and aims to capture how well the PFL methods adapt to local data variability.

3.8.4 Justification of metric choices

The metrics used for evaluation reflect the data type and the underlying class imbalance. The choice of ROC-AUC is emphasized for binary clinical outcomes due to its threshold-independent interpretability and relevance under class imbalance [17]. PR-AUC is used to complement ROC-AUC by focusing on precision and recall trade-offs.

Macro-averaged metrics are used for multiclass tasks like MNIST to treat all classes equally. Then, accuracy and cross-entropy loss provide a general indication of the prediction quality of a model. While accuracy and cross-entropy loss are reported in multiclass setups, they are avoided in imbalanced binary classification tasks due to their limited reliability under class imbalance [1].

More importantly, all binary classification evaluations use a fixed standard threshold of 0.5. This standard choice simplifies comparison across models, but it may lead to degraded precision, recall, and F1-score values, especially under class imbalance. In some cases, it is expected that models may produce precision or recall values of zero if the model consistently predicts only one class, even though ranking metrics like ROC-AUC or PR-AUC remain high. This effect is further discussed in Subsection 6.3.2, which examines how threshold choice and class imbalance affect classification metrics. As a result, the analysis primarily relies on ROC-AUC and PR-AUC, which are threshold-independent and more informative under imbalance.

Finally, in PFL, these metrics are computed per client to evaluate local adaptation performance. This client-focused evaluation aligns with the core objective of improving model performance under heterogeneous, institution-specific data distributions.

3.8.5 Statistical analysis of experimental results

To then evaluate the robustness and significance of model performance across the different learning experiments, statistical analysis of the evaluation metrics recorded during experimentation is performed. The main focus will be on the per-client comparisons where, for each client, the performance metrics ROC-AUC and PR-AUC under FL and PFL pipelines are recorded. ROC-AUC and PR-AUC are robust to class imbalance, as previously mentioned as well, which is why the statistical analyses are focuses on only these two metrics. This is especially relevant, since no threshold optimization was applied, and other threshold-dependent metrics like precision, recall and F1-score may be unreliable here.

In order to assess if results in PFL methods are a significant improvement over global FL methods, a paired test will be applied to the per-client metrics (ROC-AUC and PR-AUC values). Given the likely non-normal distribution of performance metrics due to the heterogeneousness of clinical datasets, a Wilcoxon signed-rank test is chosen and will be used for the comparisons of results [12].

Furthermore, all experiments are conducted with three random seeds (7, 42 and 1337), and for each seed, the models are re-initialized and re-trained independently. For each client and method, mean and standard deviations are calculated across these seed runs to summarize performance and variability.

However, it should be noted that a small number of runs on 3 seeds limits the statistical power of the Wilcoxon test. Hence, because of this setup, the resulting p-values found in this thesis are mostly exploratory indications for performance differences. But despite this, the test still serves as a possibility to find possible trends in per-client improvements under PFL and to provide statistical views on model behavior across clients (though limited).

3.9 Implementation and Reproducibility

The FL system is implemented in Python using PyTorch, with all training pipelines structured in modular Jupyter notebooks. These notebooks are organized to separate model definition, local training logic, aggregation mechanisms, and evaluation routines. This modularity supports flexible experimentation across centralized, FL, and PFL setups, with shared components reused across configurations to ensure consistent benchmarking.

Federated learning functionality, including FedAvg and FedProx, is implemented through well-documented function blocks within the FL notebook. Similarly, the personalized learning pipelines (FO Per-FedAvg and the MOCHA-inspired model) rely on a larger set of helper functions for local model initialization, adaptation, and client-specific evaluation.

But despite the increased complexity of the PFL setup, key utilities are reused across both FL and PFL pipelines to promote code consistency and minimize duplication. Random seeds are fixed for data splits and client sampling, reducing stochastic variation and improving reproducibility. While hardware differences like GPU performance or Mac OS environment may affect training time slightly, all software dependencies were locked to specific versions of libraries used in order to maintain consistent behavior across experiments.

Finally, the complete codebase, including notebooks, configuration settings, and documentation, is available via a private GitHub repository[1] shared with supervisors.

[1] GitHub Repository

4 Experimental setup

This chapter outlines the experimental setup used to evaluate both FL and PFL models. It begins with a brief description of the datasets used in this study. Following data overview, the chapter details the experimental design used to assess model performance under conditions of non-IID data distributions and missing values, which are both common in cross-institutional healthcare settings. The experiments utilize two publicly available healthcare datasets, namely MIMIC-III and eICU, chosen to reflect the complexity and heterogeneity of real-world clinical data collected from different hospitals.

4.1 Data overview

This section covers the description of both clinical datasets utilized the experiments conducted in this thesis.

4.1.1 Overview of the MIMIC-III Dataset

The MIMIC-III database is a large, publicly available collection of de-identified health records from ICU patients admitted to the Beth Israel Deaconess Medical Center [7]. The dataset contains time-stamped information such as demographics, vital signs, laboratory results, medication administrations, procedures and discharge outcomes. It is organized across 26 CSV tables, but in this thesis, only a subset were used.

Unlike MIMIC-III-Extract that relies on the CSV tables `CHARTEVENTS.csv` and `ADMISSIONS.csv`, this thesis employed a more selective subset of tables that included direct references to ICU stay identifiers, specifically the label (`ICUSTAY_ID`), along with consistent timestamp formats. In other words, tables that did not include (`ICUSTAY_ID`) were excluded to simplify patient-level alignment and to avoid complex joins. This selective strategy was also mainly motivated by the aim of ensuring that the experimental design and setup could be readily reused for analyses on other datasets, such as eICU. Table 4.1 below shows the tables that were included in the aggregated step of CSV files.

Table 4.1: Overview of CSV files that were aggregated in data preprocessing along with their descriptions

Name of CSV file	Brief description
`ICUSTAYS.csv`	Documents ICU stays and associated units, identified by `ICUSTAY_ID`.
`DATETIMEEVENTS.csv`	Logs the timing of events like catheter insertions or dialysis, without associated measurements.
`INPUTEVENTS_CV.csv`	Records of fluid and medication input using the CareVue system.
`INPUTEVENTS_MV.csv`	Records of fluid and medication input using MetaVision system.
`OUTPUTEVENTS.csv`	Contains information about outputs such as urine or blood loss during ICU stays.
`PROCEDUREEVENTS_MV.csv`	Structured procedure records.

The tables that were explicitly excluded are CHARTEVENTS.csv, which was omitted due to its large size, which presented data mangement and processing challenges that were beyond the scope and resources of this study. Other tables, ADMISSIONS.csv, PATIENTS.csv, LABEVENTS.csv, MICROBIOLOGYEVENTS.csv, NOTEEVENTS.csv were all excluded due to the lack of direct ICU stay identifiers, ICUSTAY_ID, or insufficient temporal resolution within the first 24 hours.

Furthermore, all retained tables were filtered to events occurring within the first 24 hours of each ICU stay. Rows outside this time window were discarded. Table 4.2 summarizes the proportion of rows retained after this filtering step.

Table 4.2: Rows retained from each MIMIC-III table within the first 24 hours

CSV file	Retained Rows	Total Rows
DATETIMEEVENTS.csv	797,085	4,485,937
ICUSTAYS.csv	61,532	61,532
INPUTEVENTS_CV.csv	3,158,746	17,527,935
INPUTEVENTS_MV.csv	1,010,925	3,618,991
OUTPUTEVENTS.csv	1,045,350	4,349,218
PROCEDUREEVENTS_MV.csv	143,908	258,066

When all selected tables were merged by ICU stay and hour, the final MIMIC-III dataset contained $61,532$ patient-stay rows and $7,304$ hourly-aggregated feature columns. Then three additional demographic variables (subject ID, gender, date of birth) were also appended, bringing the intermediate total to $7,307$ columns.

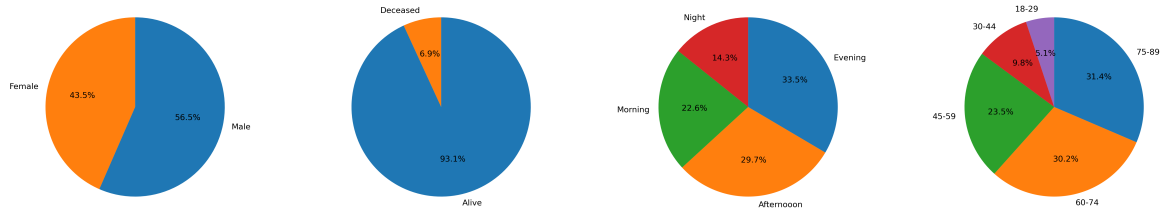
Following the aggregation step, all features were filtered and preprocessed as described in Subsection 3.3.2. This included dropping features with more than 95% missingness, imputing remaining missing values using mean imputation and applying standard scaling on the split training data only. Categorical variables were one-hot encoded, time-based features were cyclically encoded, and dynamic features were retained only if they were present at all 24 hourly bins.

The preprocessed data would then be saved into numpy arrays for easy data storage. In order to support experiments for both FL and PFL, several set of numpy arrays were saved, and all of the saved arrays are listed below:

- **Static (all) train (**$27585,\ 10$**) and static (all) test (**$6897,\ 10$**):** all static features.
- **Static (shared) train (**$27585,\ 6$**) and static (shared) test (**$6897,\ 6$**):** static features shared between MIMIC-III and eICU.
- **Static (private-only) train (**$27585,\ 4$**) and static (private-only) test (**$6897,\ 4$**):** client-specific static features (MIMIC-III-only).
- **Dynamic train (**$27585,\ 24,\ 205$**) and dynamic test (**$6897,\ 24,\ 205$**):** Hourly dynamic features over 24 hours.
- **Label train (**27585**,) and label test (**6897**,):** binary outcome label.

For all prediction tasks in this thesis, the target label used is the binary variable is_alive, where 1 indicates survival and 0 indicates death. This definition reverses the typical convention in mortality prediction literature, where death is labeled as 1. Consequently, survival is treated as the positive class in training and evaluation. This is crucial for interpreting performance metrics, for example, high recall reflects correctly identifying a large proportion of actual survivors, not deaths. Further details on evaluation metrics were provided in Subsection 3.8.4.

To provide a visual overview of the feature composition, Figure 4.1 shows the distribution of static feature categories for MIMIC-III, separated into shared and private subsets.



Gender split | Is alive split | Admission hour | Age group distribution

Figure 4.1: Pie plots of a subset of demographic and temporal distributions in the MIMIC-III dataset

To summarize, the saved arrays represent the finalized and structured inputs used throughout the experiments. By separating shared and local static features, and preserving temporal structure in the dynamic features, this design enables direct comparison between global and personalized FL models under realistic, client-specific data conditions.

4.1.2 Overview of the eICU Dataset

Table 4.3: Overview of CSV files that were aggregated in data preprocessing along with their descriptions

Name of CSV file	Brief description
`admissionDx.csv`	Records of primary admission diagnoses per APACHE scoring, used for severity estimation.
`diagnosis.csv`	Logs all diagnoses assigned during the ICU stay, ordered by severity.
`lab.csv`	Main source for structured laboratory test results, with time-stamped values.
`medication.csv`	Lists active medication orders issued during ICU stay, but does not guarantee administration.
`infusionDrug.csv`	Contains records of intravenous drug infusions, with dosage and timing.
`intakeOutput.csv`	Documents fluid balance through intake and outputs (like urine, drains).
`microLab.csv`	Results of microbiology tests including cultures and pathogen detection.
`nurseAssessment.csv`	Structured nursing observations including mental status, mobility, and skin integrity.
`respiratoryCharting.csv`	Respiratory assessments including oxygen therapy, airway status, and ventilation data.
`customLab.csv`	Hospital-specific lab results not covered by the standard `lab.csv`.
`carePlanGeneral.csv`	Care plan notes including goals, discharge readiness, and consults.

The eICU Collaborative Research Database is a multi-center critical care dataset containing de-identified health records for ICU patients across more than 20 hospitals. It includes 23 CSV files, each linked by the key `patientUnitStayID`. The data is centered on ICU admissions, and all time offsets are relative to ICU admission. Similarly with the MIMIC-III data, only a subset of tables was selected for this study, based on data completeness, timestamp precision, and the presence of ICU-aligned identifiers.

Like with MIMIC-III, the preprocessing focused on retaining events from the first 24 hours. Tables that did not have the ICU admission identifier (`patientUnitStayID`) or did not have consistent timestamps were excluded. Table 4.3 summarizes the CSV files used in the aggregation process.

All selected tables were filtered to include only events within the first 24 hours of ICU admission. This restriction ensures temporal consistency across patients and aligns with the design used for the MIMIC-III dataset. Table 4.4 shows the number of rows retained from each CSV file after this filtering step.

Table 4.4: Rows retained from each eICU table within the first 24 hours

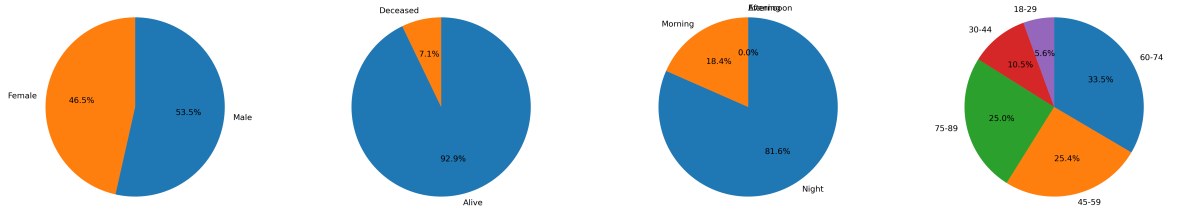
CSV file	Retained Rows	Total Rows
admissionDx.csv	572,006	626,858
diagnosis.csv	1,098,773	2,710,672
lab.csv	8,733,243	39,132,531
medication.csv	4,152,212	7,301,853
infusionDrug.csv	1,349,234	4,803,719
intakeOutput.csv	2,903,401	12,030,289
microLab.csv	4,240	16,996
nurseAssessment.csv	4,232,705	15,602,498
respiratoryCharting.csv	4,710,117	20,168,176
customLab.csv	390	1,082
carePlanGeneral.csv	2,048,636	3,115,018

Once all of the selected tables were merged on `patientUnitStayID` and hour, the final dataset contained $200,859$ ICU stays and $6,231$ hourly-aggregated feature columns. Feature preprocessing was then applied in the same manner as for MIMIC-III, as detailed in Subsection 3.3.2. The preprocessed data was saved into numpy arrays just like how it was done for MIMIC. The list of saved arrays is shown below:

- **Static (all) train (**$34137,\ 9$**) and static (all) test (**$8535,\ 9$**):** all static features.
- **Static (shared) train (**$34137,\ 6$**) and static (shared) test (**$8535,\ 6$**):** static features shared between MIMIC-III and eICU.
- **Static (private-only) train (**$34137,\ 3$**) and static (private-only) test (**$8535,\ 3$**):** client-specific static features (MIMIC-III-only).
- **Dynamic train (**$34137,\ 24,\ 123$**) and dynamic test (**$8535,\ 24,\ 123$**):** Hourly dynamic features over 24 hours.
- **Label train (**34137**,) and label test (**8535**,):** binary outcome label.

These saved arrays define the feature visibility under different learning setups. In FL, only the six static features that are common across both clients are visible and used for model training.

In PFL, each client instead accesses their full static feature set, which includes the six shared features but plus three additional static features for eICU and four for MIMIC-III, respectively. In addition to this, both clients utilize their full set of dynamic features. Furthermore, visual overview of the composition of static features for eICU is shown Figure 4.2.



Gender split | Is alive split | Admission hour | Age group distribution

Figure 4.2: Pie plots of a subset of demographic and temporal distributions in the eICU dataset

To conclude, both MIMIC-III and eICU were preprocessed the same way to ensure consistency in temporal structure, feature formatting and outcome labeling. Furthermore, by isolating shared versus local static features and retaining all dynamic sequences, the resulting datasets were ready to be used for evaluating the centralized, FL and PFL setups under realistic conditions for healthcare institutions.

The next section outlines the experimental design and model configurations used to assess performance across these scenarios.

4.2 Centralized experiments

This section establishes how the centralized models are used to define performance baselines. These experiments were, therefore, used to provide upper bounds for model performance and are used to validate architectural choices but without the challenges of FL and PFL being influencing factors.

4.2.1 MLP on static shared data

In this experiment, a two-layer MLP (described in Subsection 3.7.2) was trained using only the six static features shared between the MIMIC-III and eICU datasets. This simulates the input limitations faced in FL settings, where feature alignment across institutions is required.

Each dataset was treated as an individual training task. The model was trained separately on MIMIC-III and eICU for binary prediction, using the `is_alive` label as the target, and as described in Section 4.1, 1 indicates survival and 0 indicates death. The goal was to evaluate baseline performance when training on reduced, aligned features under ideal centralized conditions.

4.2.2 LSTM + MLP on full data

The second centralized experiment used the LSTM + MLP model from Section 3.4, and it was trained on the full feature sets (both static and dynamic) available in MIMIC-III and eICU. This setup represents the most information-rich scenario, simulating an ideal case without any constraints.

The model was trained on each dataset independently and serves as the upper bound for the FL and PFL experiments. It also tests the capacity of hybrid model architectures to capture temporal clinical patterns and how they can connect with static patterns.

4.2.3 MLP on MNIST data

In order to verify the correctness of the pipelines implemented in this thesis and to establish a reference point under clean and ideal conditions, the static MLP was also trained on the MNIST dataset, serving as a check to isolate issues potentially arising from noisy or imbalanced clinical data

MNIST consists of 60000 training and 10000 test images of handwritten digits with dimensions 28×28. The model input was flattened to 784 dimensions, and classification was performed over 10 digit classes. The inputs of the static MLP was also adjusted to match the flattened dimensions, as described in Subsection 3.7.3.

4.2.4 Training parameters for centralized experiments

To ensure comparability and stable training, all centralized models were trained using the same optimizer and general hyperparameters listed in Table 4.5. Model-specific loss functions and input dimensions were adjusted to fit the dataset and task.

Table 4.5: Training Parameters for Centralized Models

Parameter	Value
Epochs	Early stopping
Batch size	64
Optimizer	`Adam`
Learning rate	$1 \cdot 10^{-4}$
Weight decay	$1 \cdot 10^{-5}$
Evaluation frequency	Each epoch

For the dataset-specific details, any training with MIMIC-III and eICU used `BCEWithLogitsLoss` for binary classification, and input dimensionality also varied based on the static-only and the full static and dynamic experiments. Furthermore, any training with MNIST used `CrossEntropyLoss` for its multi-class classification task, and input was, as previously described, the flattened 784-dimensional vector.

4.3 FL experiments

One of the main experiments conducted in this thesis is the exploration of FL with the MIMIC-III and eICU datasets, aiming to evaluate how FL methods perform under data heterogeneity when training is distributed across clients with institution-specific data with aligned input features. For the experiments in this study, the clients are structured as follows (consistent through all experimental setups):

- **Client 1:** MIMIC-III dataset.
- **Client 2:** eICU dataset.

Each client retains their data locally using the exact static MLP architecture explained in Subsection 3.7.2. Model updates are coordinated via a central sever (using FedAvg and FedProx), and only model parameters are shared during training, further ensuring that the raw data remains private.

For the entire FL pipeline, only the shared static features between MIMIC-III and eICU were used to ensure FL comparability. The prediction target is binary mortality outcome, represented by the `is_alive` variable. The shared static features were already preprocessed and split into train and test split with the target label extracted as well.

Figure 4.3 provides the general structure of the entire FL pipeline that incorporates both the FedAvg and FedProx algorithm.

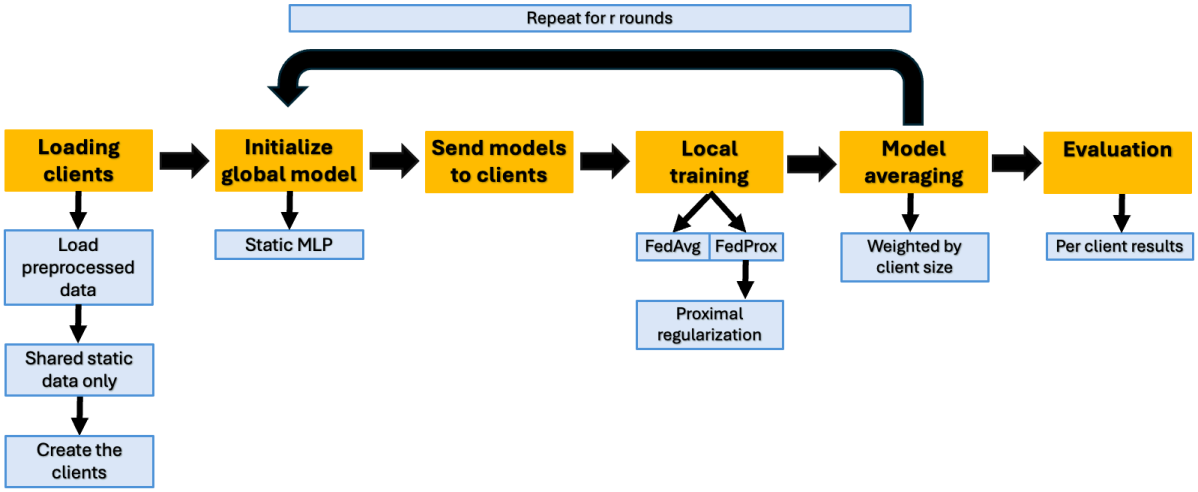


Figure 4.3: Overview of the FL pipeline, where each round r includes sending the global model to clients, local training (FedAvg or FedProx), aggregation, and evaluation on each client's test set.

In the pipeline, FedAvg is executed first and it then undergoes evaluation of its performance. Then FedProx is executed and evaluated as well. Training used binary cross-entropy loss via the `BCEWithLogitsLoss` formulation, which integrates sigmoid activation and is more numerically stable than applying standalone sigmoid followed by `BCELoss` [14].

To address severe class imbalance in the `is_alive` prediction task, where deaths (`is_alive` $= 0$) are the minority class and survivals (`is_alive` $= 1$) are the majority. `pos_weight` was computed for each client. This `pos_weight` was defined as the ratio of the minority class count to the majority class count, giving a number less than 1. Hence, this is used to mitigate imbalance by down-weighting the loss contribution from the majority class. Optimization was performed using the `Adam` optimizer.

Next, to maintain consistency across experiments and ensure comparability between methods, the FL setup used the fixed hyperparameters presented in Table 4.6.

Table 4.6: Training parameters for the FL pipeline

Parameter	**Value**
Number of clients	2 (MIMIC-III and eICU)
Input features	6 shared static features
Target label	`is_alive` (1 = survival and 0 = death)
Rounds (r)	50
Local epochs per round	5
Batch size	64
Optimizer	`Adam`
Learning rate	$1 \cdot 10^{-4}$
Weight decay	$1 \cdot 10^{-5}$
Loss function	`BCEWithLogitsLoss` (with `pos_weight`)
Evaluation frequency	Each round
μ (Proximal term strength)	$1 \cdot 10^{-2}$

The parameters in Table 4.6 were selected based on early stability training. These values provide a consistent and replicable FL setup across both FedAvg and FedProx. The learning rate ($1 \cdot 10^{-4}$) and weight decay for Adam ($1 \cdot 10^{-5}$) follow standard values commonly used in binary classification tasks on tabular data, which were known to yield stable convergence during initial tests. The batch size of 64 was chosen as a trade-off between stable gradients and memory usage.

Five local epochs were selected to strengthen the local training signal before aggregation at the cost of slightly higher per-round computation. The `Adam` optimizer was used due to its built in regularization, and lastly, the `pos_weight` term helped mitigate label imbalance across clients.

Finally, evaluation was performed after each round to monitor convergence and assess global model performance when evaluated on the held-out test set of each client. These metrics included ROC-AUC, PR-AUC and loss, both computed per client. These metrics were chosen to reflect model performance under class imbalance, as discussed in Subsection 3.8.4. Next, to ensure robustness and reduce the influence of randomness in model initialization and data shuffling, each FL experiment (FedAvg and FedProx) was repeated across three different random seeds.

For each method, the final results report the mean and standard deviation of ROC-AUC, PR-AUC, and loss across these runs. This approach provides a more reliable estimate of performance under realistic training conditions.

In addition to the main FL experiments being run on MIMIC-III and eICU, a secondary FL experiment was run on the MNIST dataset, using the same MLP architecture but adjusted for 784-dimensional inputs (Subsection 3.7.3). This served as a controlled test to ensure that the FL pipeline functioned as expected under IID and balanced conditions. Here, the same number of clients, rounds, optimizer, and training parameters from Table 4.6 were used. MNIST was split randomly into two IID partitions to isolate training behavior from clinical data noise or complexity.

4.4 PFL experiments

PFL methods are particularly relevant when data distributions vary significantly between institutions. In such settings, a single global model (as used in FL experiments) tends underperform, as it cannot capture the client-specific patterns well. So, PFL addresses this by allowing local model adaption, improving personalized performance but at the cost of global consistency. Two PFL approaches were implemented in this experimental setup, first, FO Per-FedAvg based on meta learning and then multi-task learning through a MOCHA-inspired architecture.

Each client uses the same global static MLP as in FL but as a backbone in this setup, and then with a personalized extension via a private LSTM + MLP architecture (Section 3.4).

The entire PFL pipeline used the six shared static features between MIMIC-III and eICU along with the remaining static features and all dynamic features for both datasets (Section 4.1 details the dimensions of the static (not shared) and dynamic features for both datasets). Similarly to the FL pipeline, all data has already been preprocessed and split into test and train sets with the target label `is_alive` extracted.

Moreover, the integration and update strategies differ between the two approaches, Figure 4.4 illustrates the FO Per-FedAvg implementation, while Figure 4.5 visualizes the MOCHA-inspired setup.

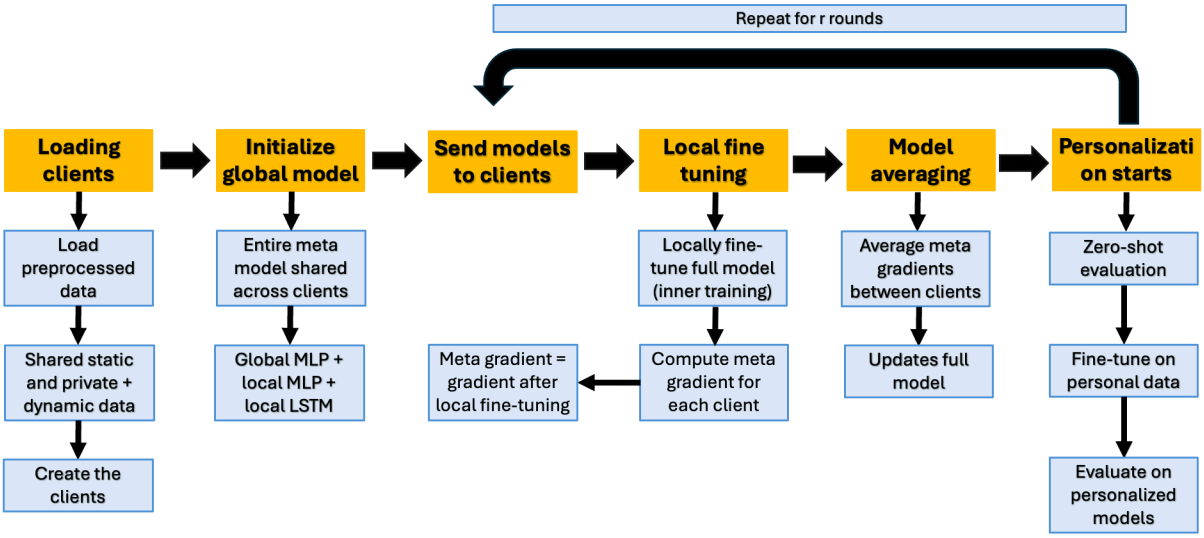


Figure 4.4: Overview of the FO Per-FedAvg (meta-learning) pipeline where, in each round, the full meta model is sent to clients, locally fine-tuned, and used to compute meta-gradients. These are aggregated on the server to update the global model, and after meta-training, clients personalize models through fine-tuning on local data.

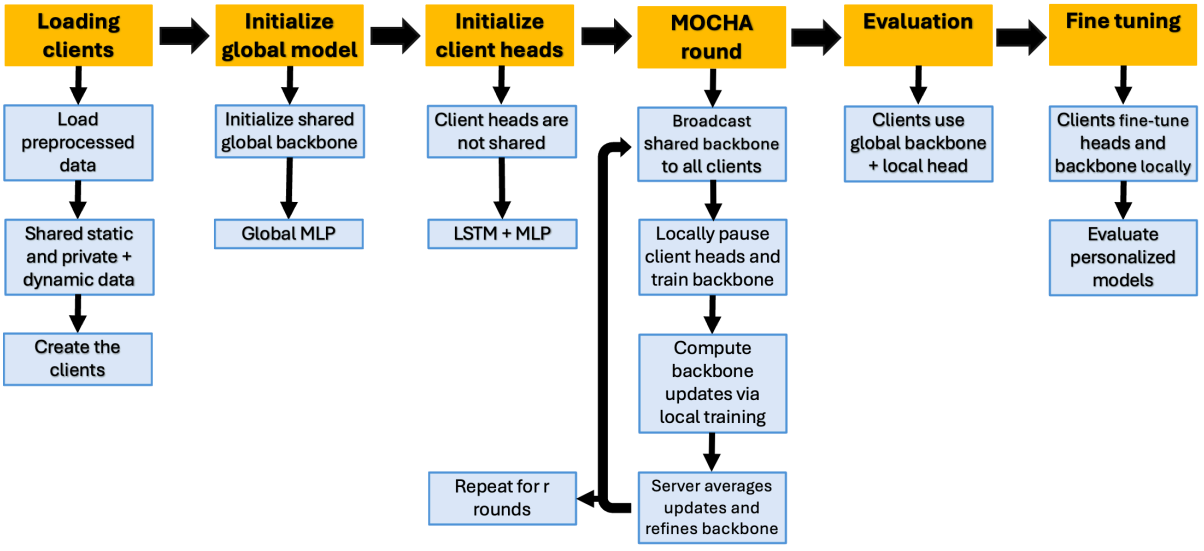


Figure 4.5: Overview of the MOCHA-inspired MTL, where a global backbone is shared across clients, while client-specific heads remain private. In each round, the backbone is trained locally with frozen heads, and updates are averaged, and after training, clients fine-tune both backbone and head locally for personalized evaluation.

The pipeline would first execute FO Per-FedAvg and then evaluate its performance, and following this, the MOCHA-inspired MTL was executed and evaluated. Just like the FL pipeline, training here used binary cross-entropy loss via `BCEWithLogitsLoss`.

To address severe class imbalance, the `pos_weight` term was again computed for each client. Optimization was performed using the `Adam` optimizer.

Table 4.7 presents the fixed hyperparameters used across the entire PFL pipeline.

These were also the same parameters used in the other experiments (Table 4.6) that were selected based on early stability training with same reasoning explained in 4.3, but with additional hyperparameters to account for the PFL techniques.

Table 4.7: Training parameters for the PFL pipeline

Parameter	**Value**
Number of clients	2 (MIMIC-III and eICU)
Input features (MIMIC-III)	6 shared static, 4 private static, 205 dynamic
Input features (eICU)	6 shared static, 3 private static, 123 dynamic
Target label	`is_alive` ($1 =$ survival and $0 =$ death)
Meta rounds (r)	50
Local epochs per round	5
Batch size	64
Optimizer	`Adam`
Inner learning rate	$1 \cdot 10^{-4}$
Meta learning rate	$1 \cdot 10^{-4}$
MOCHA learning rate	$1 \cdot 10^{-4}$
Weight decay	$1 \cdot 10^{-5}$
Loss function	`BCEWithLogitsLoss` (with `pos_weight`)
Evaluation frequency	Each round

In the table, the inner learning rate refers to inner training in FO Per-FedAvg while meta learning rate is the outer one. MOCHA learning rate simply refers to the learning rate used in the MOCHA-inspired setup.

Evaluation was performed after each round, evaluating both global and the personalized model for each client using hold-out test sets. This dual evaluation was used to assess how well the shared components generalized across clients along with how effectively personalization improved performance on local data. The metrics include ROC-AUC and PR-AUC along with loss.

Furthermore, just like the FL experiments, each PFL experiment was repeated across three different random seeds to account for variability due to initialization and data shuffling. Final performance metrics (ROC-AUC, PR-AUC, and loss) are reported as the mean and standard deviation across these runs, providing a more robust and reproducible comparison between FO Per-FedAvg and the MOCHA-inspired architecture.

Similarly to the testing of FL experiments, a secondary PFL experiment was also conducted using MNIST again to validate the correctness of the FO Per-FedAvg and MOCHA-inspired implementation under controlled conditions with explicit label heterogeneity.

The same MLP architecture was reused from Subsection 3.7.3. The PFL setup (meta rounds, local steps, optimizer, learning rates) followed the same configuration as the healthcare PFL experiments shown in Table 4.7, again to ensure consistency and isolate the impact of data heterogeneity. But this time, to simulate non-IID conditions, the MNIST dataset was split between two clients with disjoint label sets where Client 1 received digits $\{0, 1, 2, 3, 4\}$, and Client 2 received digits $\{5, 6, 7, 8, 9\}$.

This setup enforced label distribution skew, allowing PFL to be evaluated in a controlled but heterogeneous multi-class setting similar to SOTA research papers.

However, it is also important to note that this split violates the assumptions of meta-learning, as Per-FedAvg assume that client tasks are drawn from the shared distribution and that some relation between tasks exists, which can allow meta models to generalize across tasks. In this MNIST experiment, clients receive different label spaces (one only sees digits 0 to 4, and the other sees 5 to 9), so there is no class overlap.

From [4] and Chapter 2 it is shown that such conditions will result in poor zero-shot performance, since the meta-initialization cannot generalize meaningfully across unrelated tasks. This experiment is, thus, not intended to evaluate generalization, but rather to test personalization capabilities of FO Per-FedAvg under extreme task heterogeneity. This is particularly relevant for drawing comparisons to the MIMIC-III and eICU experiments, where the client data distributions are known to be very disjoint as well.

To assess the statistical significance of observed performance differences between methods, statistical tests, specifically Wilcoxon signed-rank test, were applied to the evaluation metrics reported per client over rounds comparing FO Per-FedAvg against FedAvg and FedProx as well as comparing MOCHA-inspired against FedAvg and FedProx. Further details of the test methodology, including the chosen metrics and significance thresholds, are described in Subsection 3.8.5. This ensures that observed improvements are not due to random variation, especially in the presence of class imbalance and limited sample sizes per client.

4.5 Summary of experimental setups

This section provides a complete overview of all experimental setup configurations across centralized, FL and PFL training pipelines, as shown in Table 4.8.

Table 4.8: Summary of all experimental setups across different pipelines

Experiment type	Model	Dataset	Feature setup	Goal
Centralized	MLP	MIMIC-III and eICU	Shared static features	Upper bound under FL constraints
Centralized	LSTM + MLP	MIMIC-III and eICU	Full static and dynamic features	Ideal upper bound (no FL constraints)
FL	MLP	MIMIC-III and eICU	Shared static features	Baseline performance in global FL
PFL	FO Per-FedAvg	MIMIC-III and eICU	Shared static + private features	Personalized adaptation per client
PFL	MOCHA-inspired	MIMIC-III and eICU	Shared static + private features	Federated personalization via partial sharing
Centralized	MLP	MNIST	Flattened pixel values	Centralized baseline validation
FL	MLP	MNIST	Flattened pixel values	Federated pipeline validation
PFL	MLP	MNIST	Flattened pixel values	Personalized pipeline validation

5 Results

This chapter presents the empirical findings of the study with model performance evaluated primarily through the ROC-AUC score. This metric was selected for its demonstrated robustness against the class imbalance inherent to the clinical datasets under investigation, as detailed in Subsection 3.8.4.

Table 5.1 serves as a synoptic overview of the principal results, presenting the final ROC-AUC scores from all experiments.

Table 5.1: ROC-AUC values across different models on a 20% held-out set where bold text highlights the main experimental results. Note that for MNIST, the ROC-AUC is macro-averaged and PFL models do not carry results from a global model

Model	Learning paradigm	Dataset	Data type	ROC-AUC
MLP	Centralized	MNIST	Images	0.9995
MLP	FedAvg global	MNIST	Images	0.9994
MLP	FedProx global	MNIST	Images	0.9995
MLP	FO Per-FedAvg personalized	MNIST (0-4)	Images	0.9917
MLP	FO Per-FedAvg personalized	MNIST (5-9)	Images	0.9710
MLP	MOCHA-inspired personalized	MNIST (0-4)	Images	0.9999
MLP	MOCHA-inspired personalized	MNIST (5-9)	Images	0.9995
MLP	Centralized	MIMIC-III	Static	0.6612
MLP	Centralized	eICU	Static	0.6056
MLP	FedAvg global	MIMIC-III and eICU	Static	0.6266 ± 0.0045
MLP	FedProx global	MIMIC-III and eICU	Static	0.6285 ± 0.0031
MLP	**FedAvg per-client**	**MIMIC-III**	**Static**	**0.6478 ± 0.0111**
MLP	**FedAvg per-client**	**eICU**	**Static**	**0.6094 ± 0.0009**
MLP	**FedProx per-client**	**MIMIC-III**	**Static**	**0.6575 ± 0.0057**
MLP	**FedProx per-client**	**eICU**	**Static**	**0.6050 ± 0.0010**
LSTM + MLP	Centralized	MIMIC-III	Static + dynamic	0.8458
LSTM + MLP	Centralized	eICU	Static + dynamic	0.7738
LSTM + MLP	**FO Per-FedAvg personalized**	**MIMIC-III**	**Static + dynamic**	**0.7194 ± 0.00**
LSTM + MLP	**FO Per-FedAvg personalized**	**eICU**	**Static + dynamic**	**0.6448 ± 0.00**
LSTM + MLP	**MOCHA-inspired personalized**	**MIMIC-III**	**Static + dynamic**	**0.6193 ± 0.1094**
LSTM + MLP	**MOCHA-inspired personalized**	**eICU**	**Static + dynamic**	**0.6386 ± 0.0162**

The following sections provide a detailed presentation of these results, examining the performance outcomes from each experimental paradigm individually.

5.1 Results of centralized models

This section outlines the results from the centralized learning paradigm. The subsections within will present the performance of the baseline MLP on the MNIST, MIMIC-III, and eICU datasets, followed by an analysis of the more complex LSTM + MLP model on MIMIC-III and eICU, which serves as an upper-bound performance benchmark.

5.1.1 MLP centralized on MNIST results

The detailed, per-class metrics are presented in the classification report in Table 5.2. It shows that the centralized model trained of MNIST achieved an overall accuracy of 97.44%. The macro-averaged ROC-AUC score was 0.9995 and the PR-AUC was 0.9961, showing high classification performance across all digit classes.

Table 5.2: Summary table of evaluation metrics for static MLP on MNIST, where all metrics apart from loss and accuracy are macro-averaged

Dataset	Accuracy	Loss	Precision	Recall	F1 Score	ROC-AUC	PR-AUC
MNIST	0.9744	0.0985	0.9744	0.9742	0.9742	**0.9995**	**0.9961**

The learning progression of the MLP model on MNIST is visualized in the accuracy and loss curves presented in Figures 5.1 and 5.2, respectively.

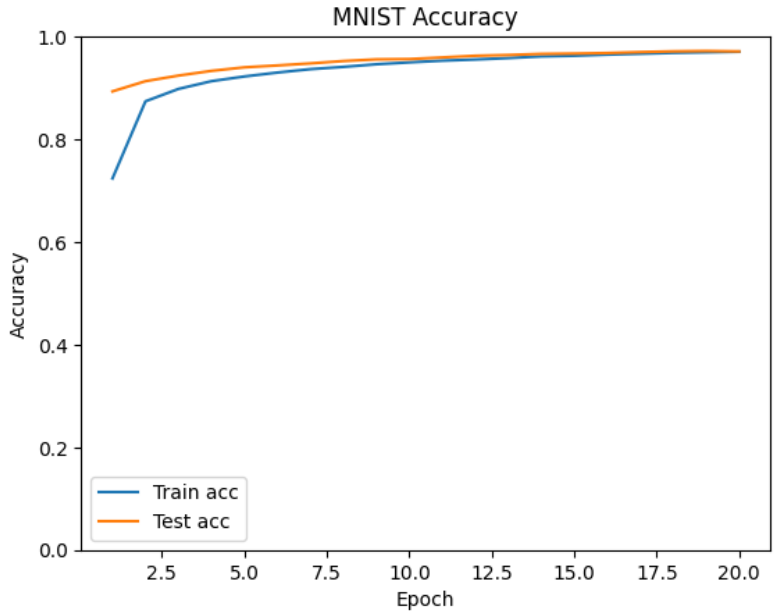


Figure 5.1: Centralized MLP trained on MNIST with train accuracy (blue) and test accuracy (orange) plotted across epochs

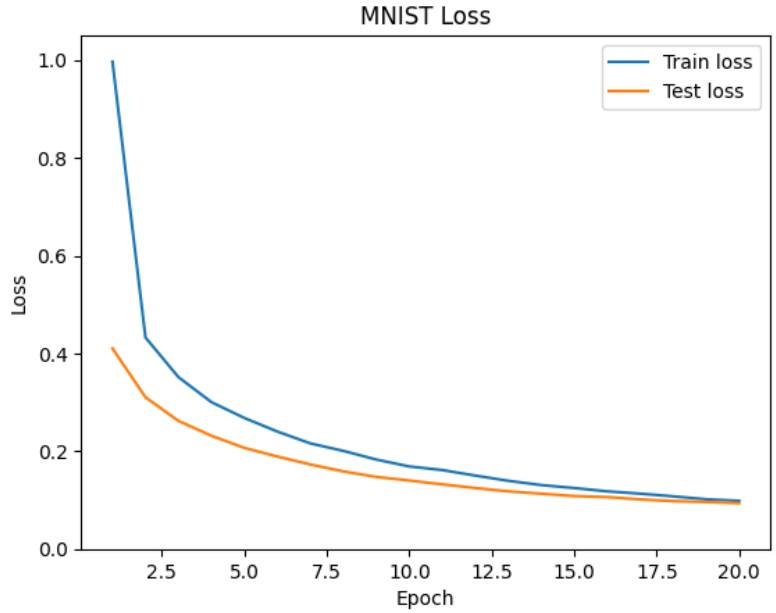


Figure 5.2: Centralized MLP trained on MNIST with train loss (blue) and test loss (orange) plotted across epochs

The accuracy plot (Figure 5.1) illustrates the model's performance on both the training (blue line) and test (orange line) sets across the training epochs.

Both curves exhibit a steep increase during the initial epochs, with accuracy rapidly climbing above 90% within the first ten epochs before gradually converging towards a plateau near 97%. The minimal gap between training and test accuracy suggests stable training behavior, with no clear sign of divergence between the two curves.

The loss plot (Figure 5.2) provides a complementary view of the training dynamics. Both the training and test loss decrease sharply from a high initial value and then continue to decline at a much slower rate as the training progresses. Both loss curves converge to low final values, showing consistent reduction in training and test error over time.

5.1.2 MLP centralized on MIMIC-III and eICU results

The MLP model, as detailed in Subsection 3.7.2, was trained separately on the MIMIC-III and eICU datasets using only six static features common across both datasets. The overall performance of these models was limited, as reported in Table 5.3, where it is seen that the model achieved a ROC-AUC of 0.6612 on MIMIC-III and 0.6056 on eICU. These values are consistent with the scores previously summarized in the overview Table 5.1. Note that in this setup, $\texttt{is\_alive} = 1$ denotes survival (as mentioned in Chapter 4), so ROC-AUC reflects the model's ability to identify survivors, reversing the typical convention in mortality prediction tasks

Table 5.3: Summary table of evaluation metrics for static MLP on MIMIC-III and eICU

Client	Loss	Precision	Recall	F1 Score	ROC-AUC	PR-AUC
MIMIC-III	2.6925	0.00	0.00	0.00	**0.6612**	**0.9599**
eICU	1.9678	0.00	0.00	0.00	**0.6056**	**0.9497**

Precision, recall, and F1-score were zero for both datasets at the default classification threshold, showing poor sensitivity and specificity for $\texttt{is\_alive} = 1$ prediction from static features alone. In contrast to this, PR-AUC values remain high (0.9599 for MIMIC-III and 0.9497 for eICU), showing the model's ability to achieve better trade-offs at varying thresholds.

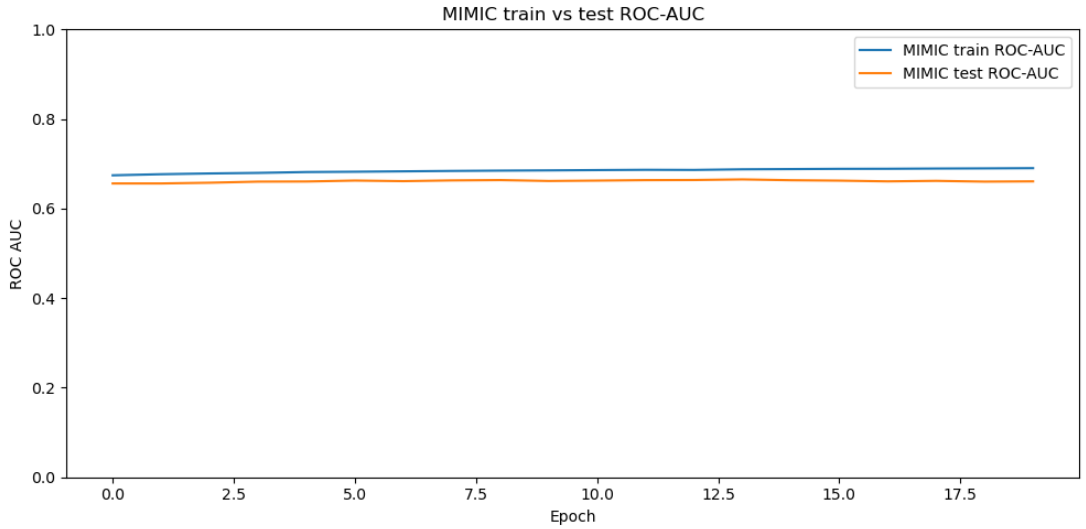


Figure 5.3: Centralized MLP trained on MIMIC-III with train ROC-AUC (blue) and test ROC-AUC (orange) plotted across epochs

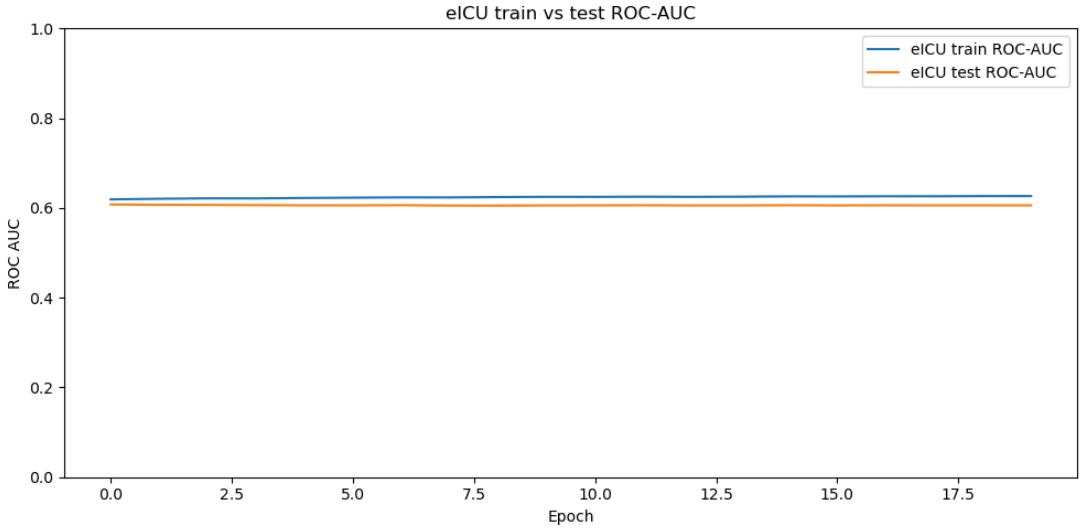


Figure 5.4: Centralized MLP trained on eICU with train ROC-AUC (blue) and test ROC-AUC (orange) plotted across epochs

The training and test progress in Figure 5.3 and 5.4 for both MIMIC-III and eICU, show that training ROC-AUC for the MIMIC-III curves converge to a value of approximately 0.68 and the test for the MIMIC-III curve converge to a value of approximately 0.66. Similarly, for the eICU ROC-AUC the training stabilized to a value of approximately 0.63 and the test curves stabilize around 0.61.

Figures 5.5 and 5.6 display the ROC curves at key training epochs for the centralized MLP model on MIMIC and eICU, respectively.

In both plots, the ROC curves for different epochs are tightly grouped together, showing minimal variation as training progresses. All curves consistently remain above the random-chance diagonal line, signifying that the model's predictions are better than random guessing.

The ROC-AUC values shown in the legends for these ROC curves are generally around 0.606 to 0.608, displaying a limited but consistent discriminative ability.

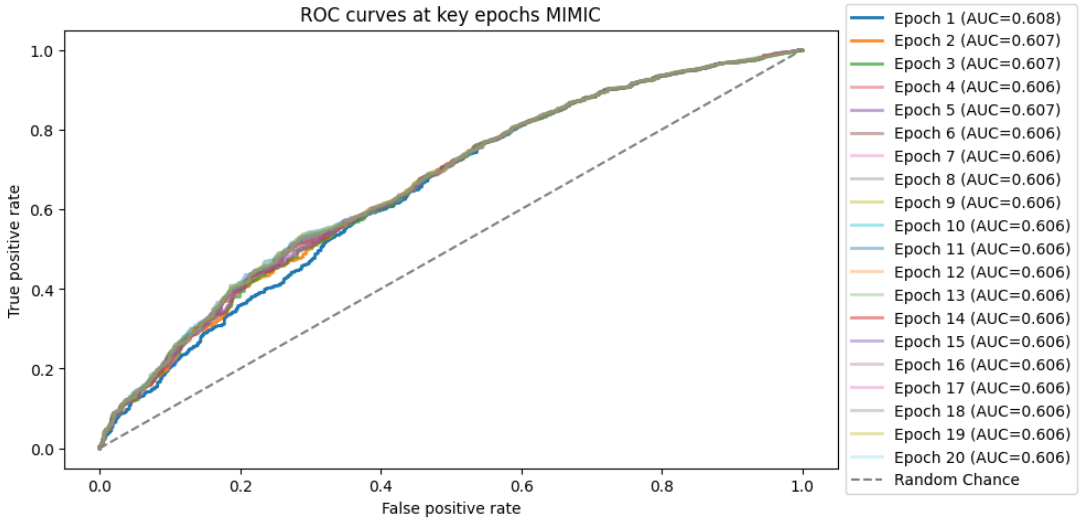


Figure 5.5: ROC curves at key epochs for centralized MLP trained on MIMIC

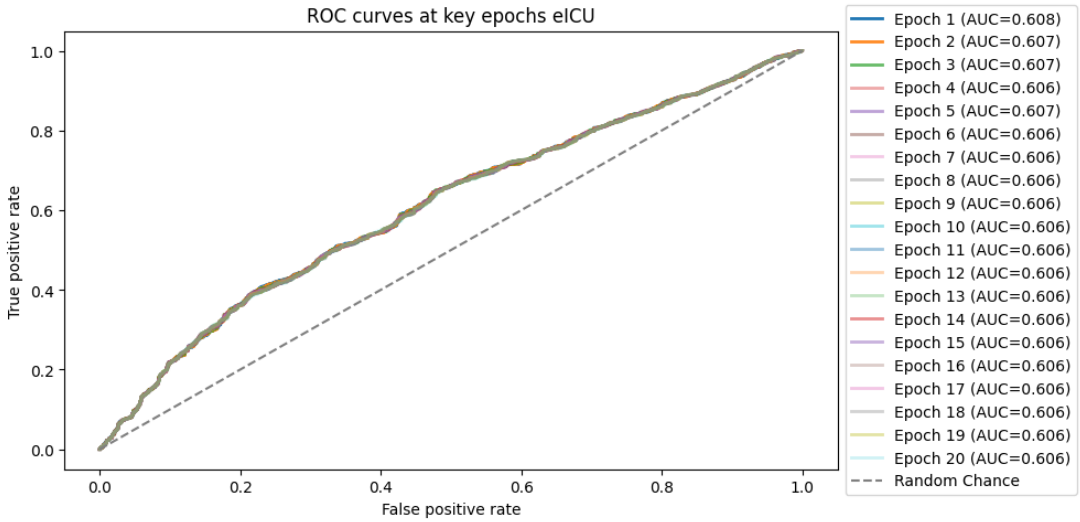


Figure 5.6: ROC curves at key epochs for centralized MLP trained on eICU

To summarize, the static MLP achieved stable but limited performance when trained on static features alone. These results form a baseline for evaluating improvements from dynamic data integration in later experiments.

5.1.3 LSTM centralized on MIMIC-III and eICU results

To establish an upper-bound performance benchmark, a more complex model incorporating both static and dynamic features was trained. An LSTM + MLP, as detailed in Subsection 3.7.1, was trained separately on the full MIMIC-III and eICU datasets.

The evaluation metrics for these centralized trained models are presented in Table 5.4. For the MIMIC-III dataset, the model achieved a ROC-AUC of 0.8458 and a PR-AUC of 0.9856. The performance on the eICU dataset resulted in a ROC-AUC of 0.7738 and a PR-AUC of 0.9740.

Table 5.4: Summary table of evaluation results for LSTM + MLP on MIMIC-III and eICU

Client	Loss	Precision	Recall	F1-Score	ROC-AUC	PR-AUC
MIMIC-III	0.0724	0.9768	0.8001	0.8797	**0.8458**	**0.9856**
eICU	0.0829	0.9652	0.7944	0.8715	**0.7738**	**0.9740**

For MIMIC-III, which is shown in Figure 5.7, the training ROC-AUC begins at 0.70 while the test ROC-AUC begins at 0.68.

Both curves show a steep increase over the initial epochs, with the training ROC-AUC reaching 0.837 by Epoch 5 and the test ROC-AUC reaching 0.818 by Epoch 5.

The test ROC-AUC curve then stabilizes around 0.84 to 0.85 from approximately Epoch 10. The training ROC-AUC curve continues its ascent throughout the training process, reaching 0.928 by Epoch 27.

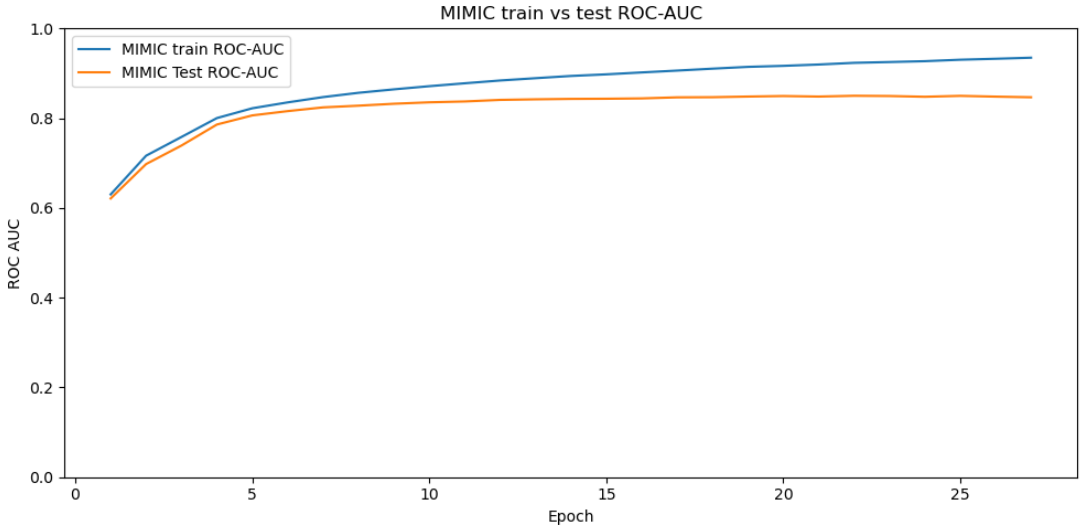


Figure 5.7: Centralized LSTM + MLP trained on MIMIC-III with train ROC-AUC (blue) and test ROC-AUC (orange) plotted across epochs

Similarly, for the eICU dataset in Figure 5.8 the training ROC-AUC starts at 0.605 and the test ROC-AUC at 0.599. Both curves show a significant rise in the initial epochs, with the training ROC-AUC reaching 0.757 by Epoch 5 and the test ROC-AUC reaching 0.745 by Epoch 5. The test ROC-AUC curve then stabilizes around 0.77 to 0.78 from approximately Epoch 10. The training ROC-AUC curve continues to improve, reaching 0.843 by Epoch 23.

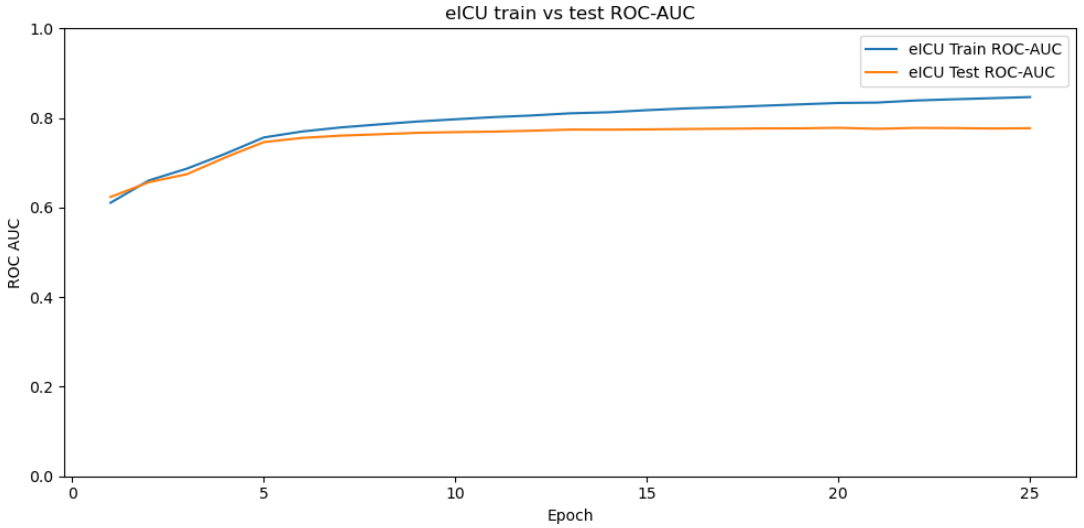


Figure 5.8: Centralized LSTM + MLP trained on eICU with train ROC-AUC (blue) and test ROC-AUC (orange) plotted across epochs

In addition to ROC-AUC and PR-AUC, the other performance metrics in Table 5.4 also show strong model performance. Both models achieve high precision of 0.9768 for MIMIC-III and 0.9652 for eICU, and recall values close to 0.80 for both, resulting in F1-scores of 0.8797 and 0.8715, respectively. The loss values are low across both datasets, showing effective convergence during training.

These results show that the model reached higher ROC-AUC when dynamic features were included, increasing from 0.6612 to 0.8458.

Figures 5.9 and 5.10 present the ROC curves at key training epochs for the centralized LSTM + MLP model. For MIMIC-III (Figure 5.9), the ROC-AUC values in the legend show an improvement from 0.685 to a final stable value of 0.851, with stabilization occurring after approximately 11 epochs.

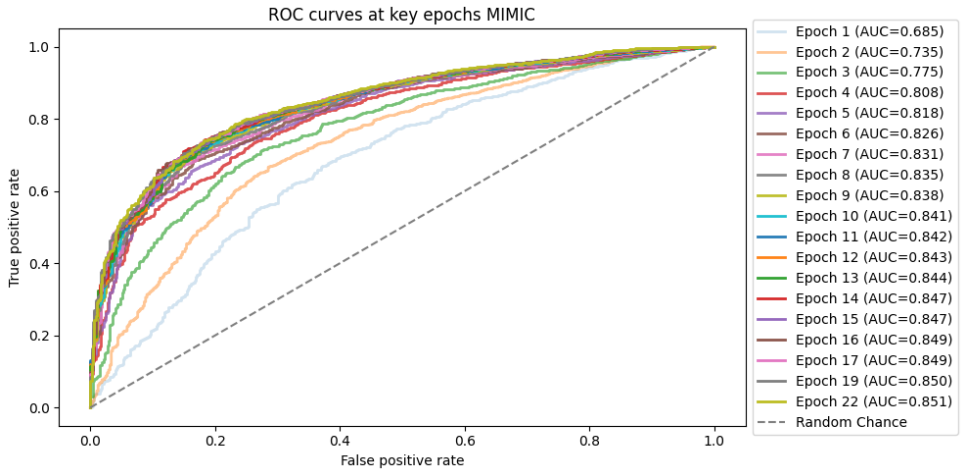


Figure 5.9: ROC curves at key epochs for centralized LSTM + MLP trained on MIMIC-III

For eICU (Figure 5.10), the ROC-AUC values in the legend improved from 0.599 to a final stable value of 0.777, with stabilization occurring after 10 epochs.

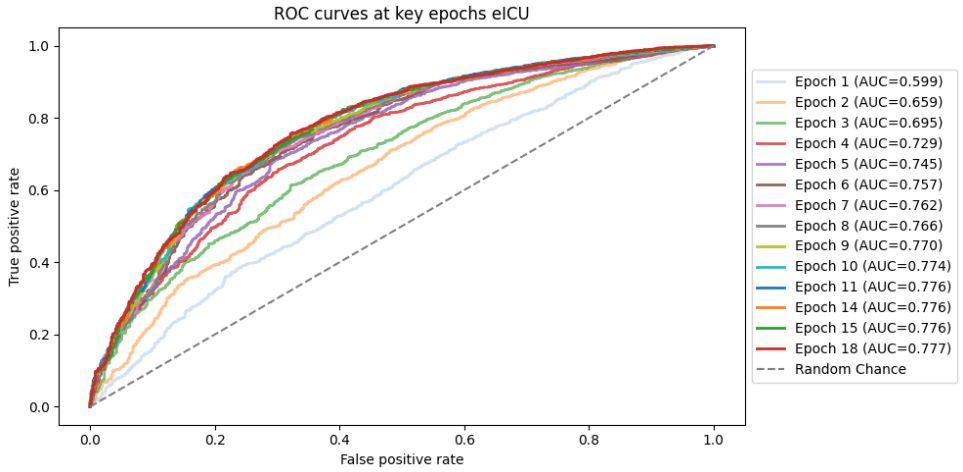


Figure 5.10: ROC curves at key epochs for centralized LSTM + MLP trained on eICU

In both plots, the ROC curves visibly progress towards the top-left corner, indicating improved performance over initial epochs. After this initial improvement, the curves become tightly grouped, and they consistently remain above the random-chance diagonal line. These results define the centralized baseline for subsequent comparisons with models trained under FL and PFL setups.

5.2 Results of federated models

This section covers the results from the FL- (FedAvg and FedProx) and PFL (FO Per-FedAvg and MOCHA-inspired) pipelines with subsections first presenting FL and PFL run on MNIST, serving as pipeline validity checks.

These are then followed by results from FL and PFL run on MIMIC-III and eICU as clients.

5.2.1 FedAvg and FedProx on MNIST results

This experiment serves to validate the implementation and functionality of the FL algorithms (FedAvg and FedProx) under balanced and IID conditions before applying them on clinical data. FedAvg and FedProx were evaluated on the MNIST dataset to establish their effectiveness in a controlled, multi-client environment. Both FedAvg and FedProx were trained with 5 local epochs per round over 50 communication rounds.

The global model trained with FedAvg achieved an overall performance of 98.09% in accuracy, a macro-averaged ROC-AUC of 0.9994, and a PR-AUC of 0.9969 as shown in Table 5.5.

Table 5.5: Summary table of FedAvg global model performance on MNIST split into two clients, where all metrics apart from loss and accuracy are macro-averaged

Dataset	Accuracy	Loss	Precision	Recall	F1 Score	ROC-AUC	PR-AUC
MNIST	0.9809	0.1018	0.9808	0.9807	0.9807	**0.9994**	**0.9969**

The global model trained on the FedProx achieved an overall accuracy of 97.98%, close to FedAvg as shown in Table 5.6. However, it obtained a marginally higher macro-averaged ROC-AUC of 0.9995 and PR-AUC of 0.9972.

Table 5.6: Summary table of FedProx global model performance on MNIST split into two clients, where all metrics apart from loss and accuracy are macro-averaged

Dataset	Accuracy	Loss	Precision	Recall	F1 Score	ROC-AUC	PR-AUC
MNIST	0.9798	0.0939	0.9796	0.9797	0.9796	**0.9995**	**0.9971**

Figure 5.11 shows that both FedAvg and FedProx models converged quickly, with test accuracy surpassing 90% in the early rounds and stabilizing near 98%. FedAvg consistently maintained slightly higher accuracy. Figure 5.12 shows a corresponding decline in test loss, where FedProx achieved a marginally lower and more stable final loss.

Both FedAvg and FedProx achieved rapid accuracy gains, surpassing 0.90 within a few rounds and converging near 0.98. FedAvg maintained slightly higher test accuracy throughout training compared to FedProx.

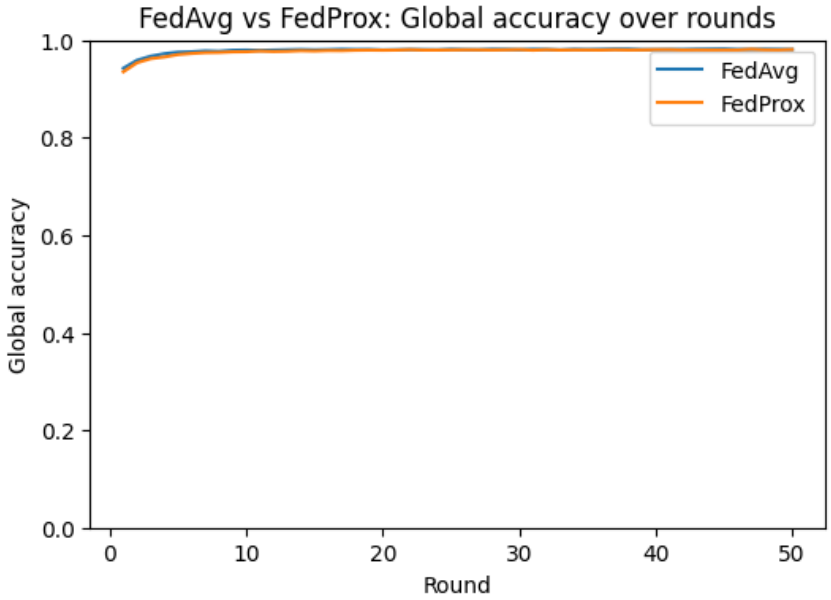


Figure 5.11: FedAvg (blue) vs FedProx (orange) global test accuracy plotted across 50 rounds of MLP trained on MNIST split into two IID clients

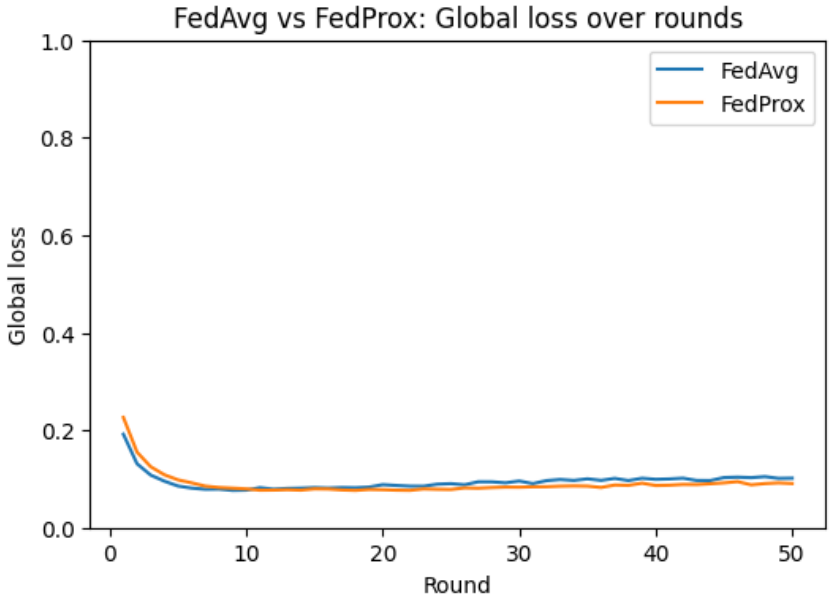


Figure 5.12: FedAvg (blue) vs FedProx (orange) global test loss plotted across 50 rounds of MLP trained on MNIST split into two IID clients

Figure 5.11 shows that both FedAvg and FedProx models converged quickly, with test accuracy surpassing 90% in the early rounds and stabilizing near 98%. FedAvg maintained slightly higher accuracy throughout. Meanwhile, Figure 5.12 shows a corresponding decline in test loss, with both models stabilizing after approximately 8 rounds. FedProx achieved a marginally lower and more stable final loss.

These results confirm that the FL implementations behave as expected under IID conditions and are valid for further application to clinical data.

5.2.2 PFL on MNIST results

Tables 5.7 and 5.8 show the performance of FO Per-FedAvg on the MNIST dataset split across two clients. Client 1 (seeing digits 0–4) achieved an accuracy of 90.62%, macro-averaged ROC-AUC of 0.9917, and PR-AUC of 0.9746. Client 2 (seeing digits 5–9) performed notably worse, with accuracy at 76.24%, ROC-AUC at 0.9710, and PR-AUC at 0.9163.

Table 5.7: Summary table of FO Per-FedAvg performance on MNIST client seeing 0,1,2,3,4 digits, where all metrics apart from loss and accuracy are macro-averaged

Dataset	Accuracy	Precision	Recall	F1 Score	ROC-AUC	PR-AUC
MNIST	0.9062	0.9136	0.9055	0.9042	**0.9917**	**0.9746**

Table 5.8: Summary table of FO Per-FedAvg performance on MNIST client seeing 5,6,7,8,9 digits, where all metrics apart from loss and accuracy are macro-averaged

Dataset	Accuracy	Precision	Recall	F1 Score	ROC-AUC	PR-AUC
MNIST	0.7624	0.8157	0.7501	0.7058	**0.9710**	**0.9163**

The MOCHA-inspired model outperformed FO Per-FedAvg for both clients. As shown in Tables 5.9 and 5.10, Client 1 achieved near-perfect metrics, with 99.94% accuracy and ROC-AUC of 0.9999. Client 2 also reached high performance with 99.59% accuracy and ROC-AUC of 0.9951.

Table 5.9: Summary table of MOCHA-inspired performance on MNIST client seeing 0,1,2,3,4 digits, where all metrics apart from loss and accuracy are macro-averaged

Dataset	Accuracy	Precision	Recall	F1 Score	ROC-AUC	PR-AUC
MNIST	0.9994	0.9994	0.9994	0.9994	**0.9999**	**0.9998**

Table 5.10: Summary table of MOCHA-inspired performance on MNIST client seeing 5,6,7,8,9 digits, where all metrics apart from loss and accuracy are macro-averaged

Dataset	Accuracy	Precision	Recall	F1 Score	ROC-AUC	PR-AUC
MNIST	0.9959	0.9958	0.9959	0.9959	**0.9995**	**0.9745**

Figures 5.13 and 5.14 show the personalized test accuracy and loss curves over 50 communication rounds for both clients. Despite the non-IID split, both clients converged rapidly. Client 1 maintained near-zero loss throughout training, while Client 2 exhibited slightly more fluctuation after round 20 but still achieved stable convergence.

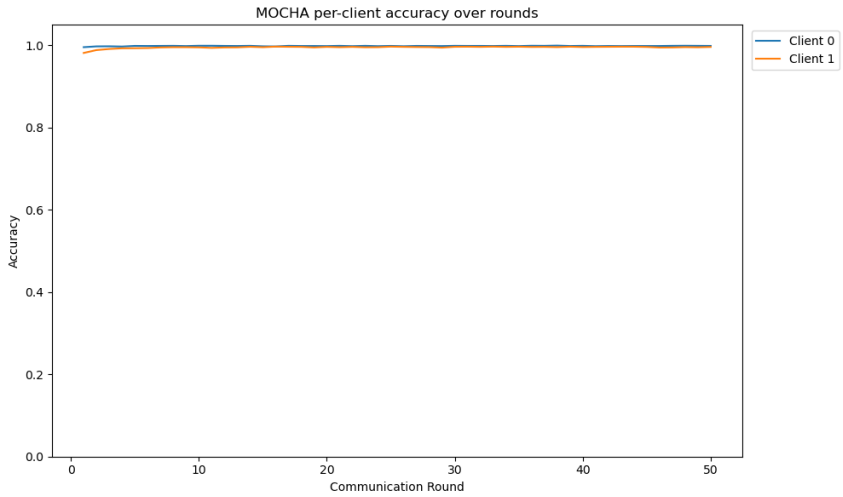


Figure 5.13: Average personalized test accuracy plotted over rounds of MOCHA-inspired MLP trained on non-IID MNIST split

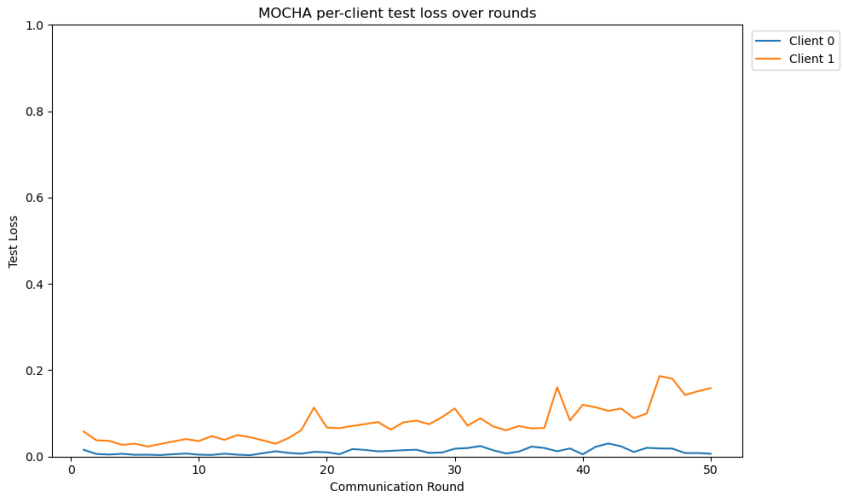


Figure 5.14: Average personalized test loss plotted over rounds of MOCHA-inspired MLP trained on non-IID MNIST split

5.2.3 FedAvg and FedProx on MIMIC-III and eICU results

The FL experiments using FedAvg and FedProx were conducted over 50 communication rounds on the MIMIC-III and eICU datasets. Results were averaged across three random seeds (7, 42, 1337), with standard deviations reported (see Subsection 3.8.5 for details). Tables 5.11 through 5.14 report performance metrics, while Figures 5.15 through 5.18 visualize the corresponding ROC-AUC trends and ROC curve shapes.

The FedAvg global model achieved a moderate ROC-AUC of 0.6266 ± 0.0045 and high PR-AUC of 0.9532 ± 0.0008, as shown in Table 5.11. However, performance on threshold-based metrics was null (precision, recall, and F1-score all zero), indicating a failure to detect any positive cases.

Table 5.11: Summary table of global model performance metrics for FedAvg on MIMIC-III and eICU

Model	Loss	Precision	Recall	F1-Score	ROC-AUC	PR-AUC
Global	1.3140 ± 0.1108	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.6266 \pm 0.0045}$	$\mathbf{0.9532 \pm 0.0008}$

At the client level, performance was slightly higher for MIMIC-III than eICU, with ROC-AUCs of 0.6478 ± 0.0111 and 0.6094 ± 0.0009, respectively, as seen in Table 5.12.

Table 5.12: Summary table of per-client performance metrics for FedAvg

Client	Loss	Precision	Recall	F1-Score	ROC-AUC	PR-AUC
MIMIC-III	1.5439 ± 0.1252	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.6478 \pm 0.0111}$	$\mathbf{0.9573 \pm 0.0018}$
eICU	1.1282 ± 0.1110	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.6094 \pm 0.0009}$	$\mathbf{0.9499 \pm 0.0003}$

Figures 5.15 and 5.16 further illustrate the ROC-AUC progression and corresponding ROC curve shapes. ROC-AUC for MIMIC-III declined slightly over the rounds, while eICU stayed near chance level. But despite poor threshold-based performance, all curves remained above the random diagonal, showing some ability to distinguish between alive and deceased patients.

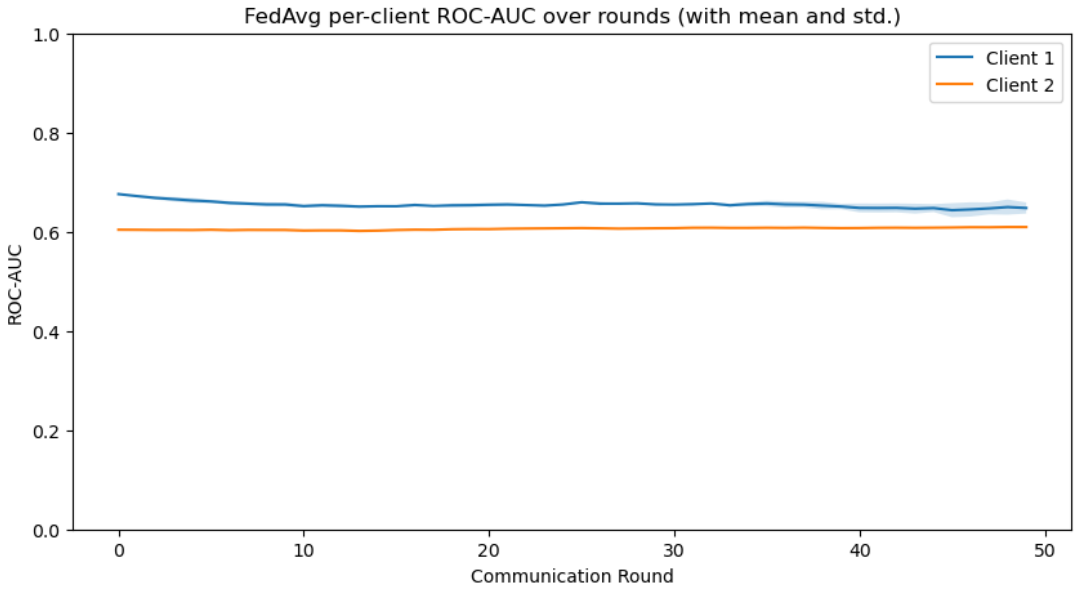


Figure 5.15: Client-wise test ROC-AUC plotted across 50 rounds for FedAvg MLP trained on MIMIC-III and eICU as clients

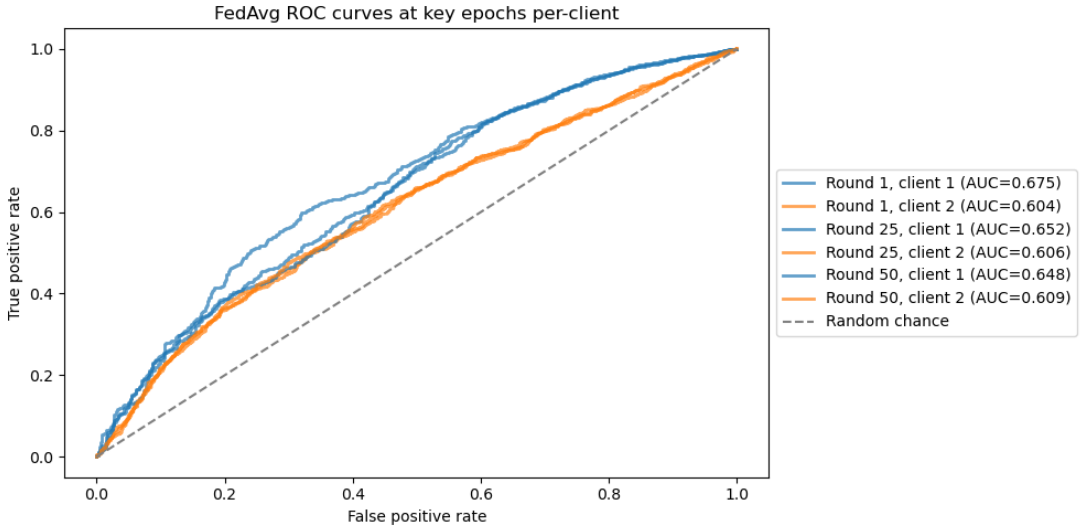


Figure 5.16: ROC curves at key epochs for FedAvg MLP trained on MIMIC-III and eICU as clients

Having detailed the performance of the FedAvg algorithm, the results for the FedProx algorithm are subsequently presented.

Global results for FedProx were similar to FedAvg (ROC-AUC 0.63, PR-AUC 0.95), but client-level performance improved slightly for MIMIC-III (ROC-AUC 0.66). eICU remained around 0.61. Precision and recall stayed at zero across models due to extreme class imbalance.

Table 5.13: Summary table of global model performance metrics for FedProx on MIMIC-III and eICU

Model	Loss	Precision	Recall	F1-Score	ROC-AUC	PR-AUC
Global	1.7409 ± 0.1067	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.6285 \pm 0.0031}$	$\mathbf{0.9539 \pm 0.0006}$

Table 5.14: Summary table of Per-client performance metrics for FedProx

Client	Loss	Precision	Recall	F1-Score	ROC-AUC	PR-AUC
MIMIC-III	2.0878 ± 0.1704	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.6575 \pm 0.0057}$	$\mathbf{0.9592 \pm 0.0012}$
eICU	1.4606 ± 0.0757	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$\mathbf{0.6050 \pm 0.0010}$	$\mathbf{0.9496 \pm 0.0002}$

Figures 5.17 and 5.18 show that FedProx produced slightly more stable ROC-AUC values over rounds, particularly for MIMIC-III. Again, all ROC curves for both datasets stayed above the diagonal, and MIMIC-III consistently outperformed eICU.

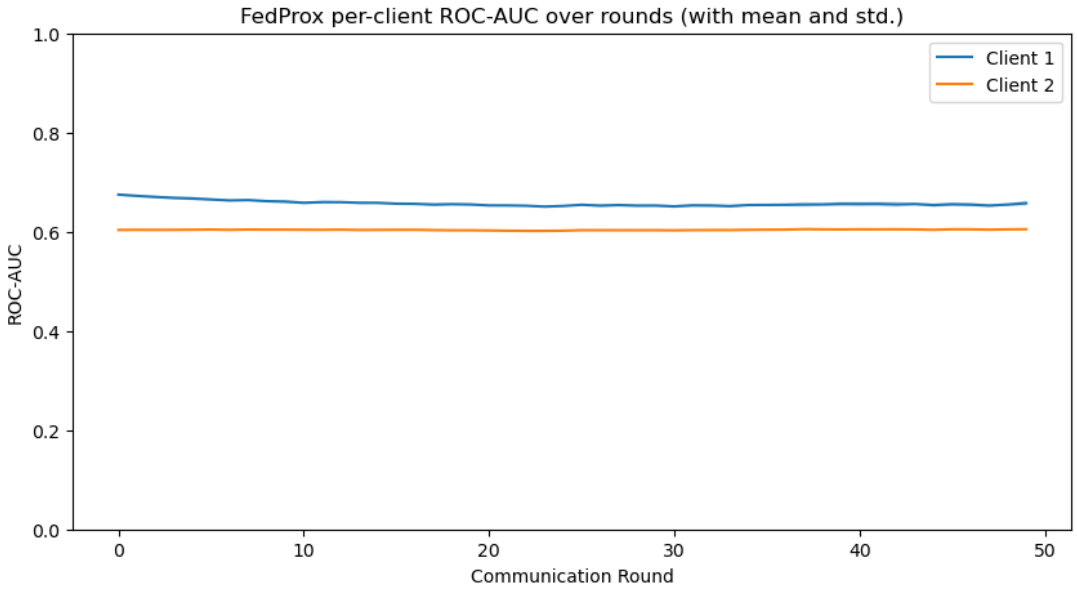


Figure 5.17: Client-wise test ROC-AUC plotted across 50 rounds for FedProx MLP trained on MIMIC-III and eICU as clients

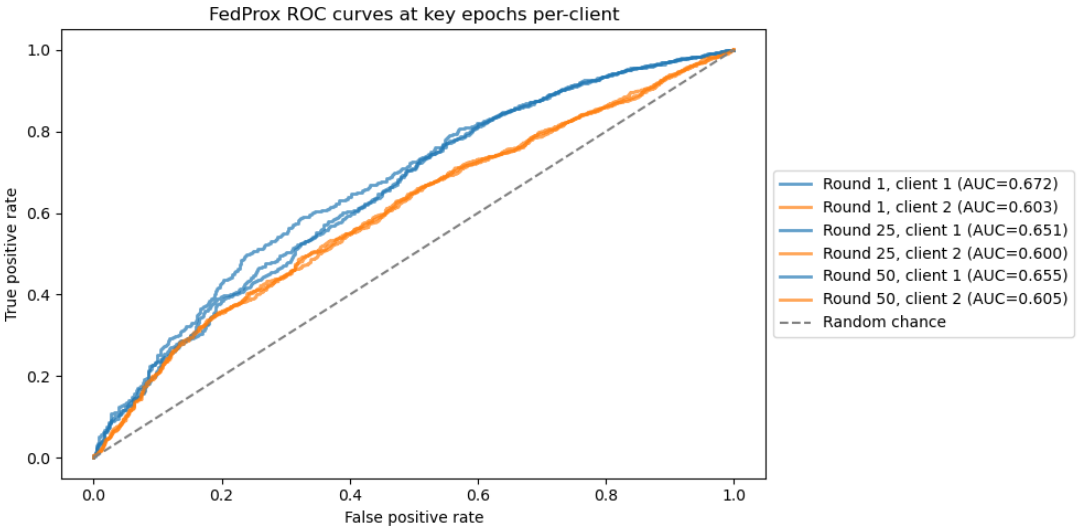


Figure 5.18: ROC curves at key epochs for FedProx MLP trained on MIMIC-III and eICU as clients

In general, both FedAvg and FedProx models showed modest ROC-AUC values and strong PR-AUC values, but poor performance at the classification threshold due to extreme label imbalance. FedProx yielded marginal improvements over FedAvg, especially for MIMIC-III. Standard deviations were higher for MIMIC-III, indicating more variable learning dynamics compared to eICU.learning algorithms on the clinical datasets, showing consistent but modest predictive power based on the limited static features available.

5.2.4 PFL on MIMIC-III and eICU results

This section details the results from the PFL experiments, which aim to train models that are tailored to the specific data distribution of each client.

The performance of two PFL algorithms, FO Per-FedAvg and a MOCHA-inspired approach, is evaluated. All metrics are averaged over three random seeds (7, 42, 1337), as in the FL experiments.

The personalized performance metrics for FO Per-FedAvg are shown in Table 5.15. The model for MIMIC-III achieved a ROC-AUC of 0.7194, while eICU reached 0.6448. However, despite the high ROC-AUC and PR-AUC values, both clients had zero precision, recall, and F1-scores. This suggests the model could rank predictions (i.e., produce well-calibrated scores), but failed to identify any positive samples at the default classification threshold.

Training dynamics for FO Per-FedAvg are shown in Figure 5.19. ROC-AUC scores for both clients fluctuated throughout the 50 communication rounds without converging. MIMIC-III ranged between 0.65 and 0.79, while eICU varied between 0.60 and 0.74. MIMIC-III consistently exhibited higher ROC-AUC values compared to eICU.

Table 5.15: FO Per-FedAvg personalized model performance metrics (mean ± standard deviation across seeds)

Client	Loss	Balanced Accuracy	Precision	Recall	F1-Score	ROC-AUC	PR-AUC
MIMIC-III	0.0773 ± 0.00	0.50 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	**0.7194 ± 0.00**	**0.9708 ± 0.00**
eICU	0.0888 ± 0.00	0.50 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	**0.6448 ± 0.00**	**0.9558 ± 0.00**

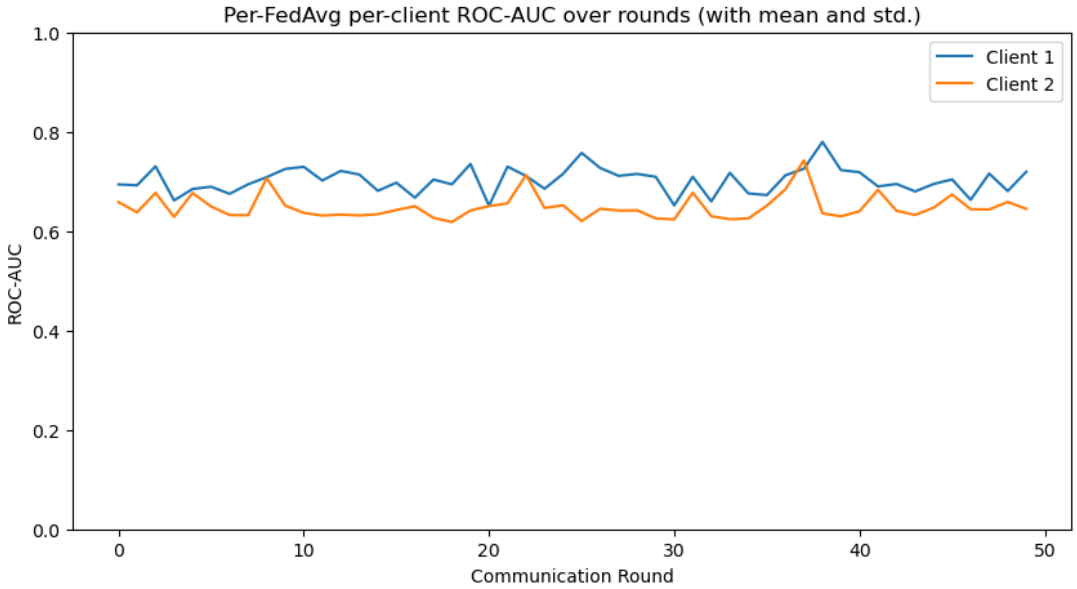


Figure 5.19: Client-wise test ROC-AUC plotted across 50 rounds for FO Per-FedAvg LSTM + MLP on MIMIC-III and eICU as clients

The performance of the MOCHA-inspired PFL model is shown in Table 5.16. The ROC-AUC was 0.6193 for MIMIC-III and 0.6386 for eICU. In contrast to FO Per-FedAvg, the personalized model for MIMIC-III demonstrated strong classification performance, with a high F1-score of 0.9371. However, the model for eICU achieved a much lower F1-score due to a very low recall, indicating that it failed to capture many of the true positive cases.

Training dynamics for MOCHA-inspired are shown in Figure 5.20. These learning curves are smoother and more stable compared to the fluctuations seen in Per-FedAvg.

MIMIC-III performance improved over the first 25 rounds, peaking near a ROC-AUC of 0.80, and then declined slightly. This shows that the final ROC-AUC reported in Table 5.16 is slightly lower than the peak performance during training. eICU maintained relatively stable ROC-AUC around 0.65 throughout.

Table 5.16: MOCHA-inspired personalized model performance metrics (mean ± standard deviation across seeds)

Client	Loss	Precision	Recall	F1-Score	ROC-AUC	PR-AUC
MIMIC-III	0.0015 ± 0.0007	0.9443 ± 0.0139	0.9358 ± 0.0891	0.9371 ± 0.0403	$\mathbf{0.6193 \pm 0.1094}$	$\mathbf{0.9491 \pm 0.0173}$
eICU	0.0066 ± 0.0009	0.9548 ± 0.0102	0.2712 ± 0.3704	0.2995 ± 0.3976	$\mathbf{0.6386 \pm 0.0162}$	$\mathbf{0.9529 \pm 0.0042}$

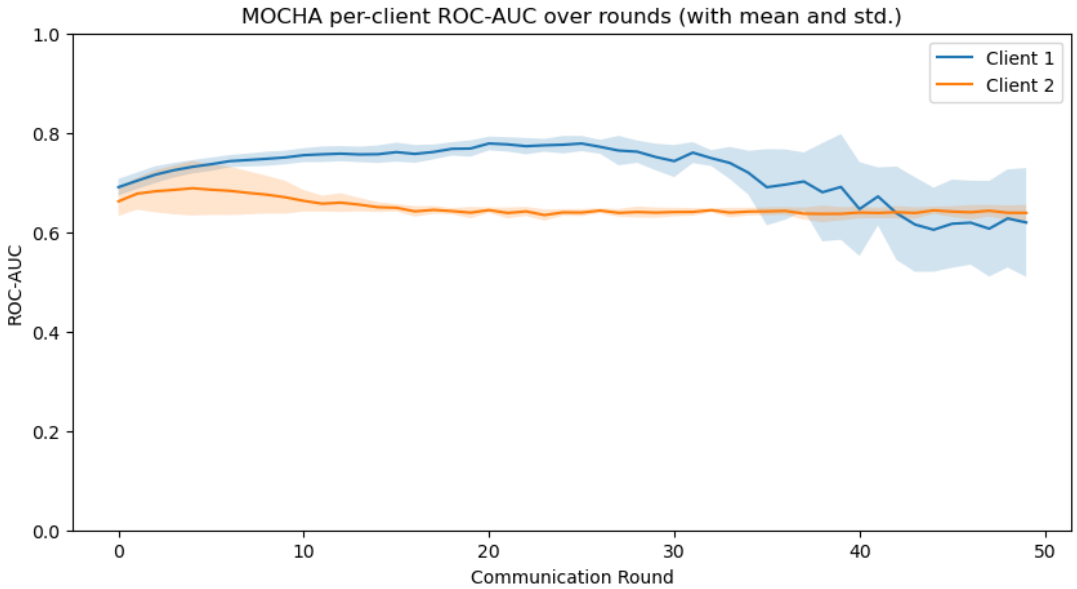


Figure 5.20: Client-wise test ROC-AUC plotted across 50 rounds for MOCHA-inspired LSTM + MLP on MIMIC-III and eICU as clients

The MOCHA-inspired model also showed higher standard deviation values across several metrics—particularly recall and ROC-AUC for MIMIC-III, and recall for eICU. This indicates greater sensitivity to initialization or client heterogeneity compared to FO Per-FedAvg, which showed no variation across seeds.

5.3 Statistical analysis FL versus PFL pipeline results

This section presents Wilcoxon signed-rank test comparisons between the personalized learning pipelines (FO Per-FedAvg and MOCHA-inspired) and the FL baselines (FedAvg and FedProx), across three random seeds (7, 42, 1337). Metrics under comparison include ROC-AUC and PR-AUC for both MIMIC-III and eICU clients.

No statistically significant differences were found between either FO Per-FedAvg or MOCHA-inspired and the federated baselines as seen in Table 5.17. In all comparisons, p-values were above 0.25, exceeding the 0.05 threshold. This indicates that neither personalized learning approach consistently outperformed standard FL baselines in a statistically meaningful way.

Table 5.17: Wilcoxon signed-rank test results: FO Per-FedAvg and MOCHA-inspired against FL baselines (ROC-AUC and PR-AUC, 3 seeds)

Client	Metric	Comparison	Statistic	p-value	Significant ($\alpha = 0.05$)	Interpretation
MIMIC-III	ROC-AUC	FO Per-FedAvg vs FedAvg	0.00	0.25	No	No significant difference
	ROC-AUC	MOCHA vs FedAvg	3.00	1.00	No	No significant difference
	PR-AUC	FO Per-FedAvg vs FedAvg	0.00	0.25	No	No significant difference
	PR-AUC	MOCHA vs FedAvg	1.00	0.50	No	No significant difference
eICU	ROC-AUC	FO Per-FedAvg vs FedAvg	0.00	0.25	No	No significant difference
	ROC-AUC	MOCHA vs FedAvg	0.00	0.25	No	No significant difference
	PR-AUC	FO Per-FedAvg vs FedAvg	0.00	0.25	No	No significant difference
	PR-AUC	MOCHA vs FedAvg	1.00	0.50	No	No significant difference

Although both PFL methods achieved numerically higher ROC-AUC and PR-AUC scores in some experiments (like MOCHA on MIMIC-III), these differences lacked statistical support. FO Per-FedAvg showed flatter performance trends with near-zero variance across seeds, whereas MOCHA showed higher variability, particularly on recall and ROC-AUC for MIMIC-III, indicating some sensitivity to initialization or client heterogeneity.

6 Discussion

This study investigated whether PFL could improve model robustness under conditions of extreme data heterogeneity in healthcare. By simulating a realistic cross-silo federation using MIMIC-III and eICU as institutional clients, the experiments tested whether personalization could outperform a simpler, shared global model trained on limited common features.

Despite PFL's theoretical strengths under non-IID conditions, the results showed no statistically significant improvement in predictive performance over global FL baselines. These findings challenge the assumption that personalization alone can address the full complexity of real-world clinical heterogeneity.

This chapter critically analyzes the results in relation to the original research questions, explores limitations in both data and methodology, compares the findings to prior literature, and reflects on the practical implications of applying FL in real-world healthcare environments.

6.1 Summary of key findings

The implementation of FL and PFL pipelines was validated on a controlled dataset, where both global and personalized methods performed as expected. On MNIST, global FL methods (FedAvg and FedProx) achieved near-perfect accuracy of 98.09%. More notably, FO Per-FedAvg for MNIST, despite operating under a disjoint label split across clients, validated the PFL pipeline under disjoint label splits. The MOCHA-inspired model also converged rapidly under this split, further validating that the PFL pipelines and model architectures behave as expected in idealized settings. Moreover, a centralized LSTM + MLP model trained on the full clinical feature sets achieved strong ROC-AUC scores of 0.8558 on MIMIC-III and 0.7738 on eICU, demonstrating that the full datasets contain meaningful signals when used outside a federated context.

However, under extreme heterogeneity and skewed data distributions, neither PFL approach (FO Per-FedAvg nor the MOCHA-inspired model) achieved a significant performance advantage over simpler global FL models. Despite access to hundreds of local features, PFL performance was statistically indistinguishable from global models limited to six static features.

For instance, global FedAvg models achieved ROC-AUC scores of 0.6478 ± 0.0111 for MIMIC-III and 0.6094 ± 0.0009 for eICU, while the FO Per-FedAvg model reached 0.7194 ± 0.00 and 0.6448 ± 0.00, respectively. Wilcoxon signed-rank tests confirmed that these differences were not statistically significant ($p > 0.05$). This result suggests that simply providing more detailed, client-specific data does not guarantee improved predictive performance in federated settings.

These findings contradict the common expectation that personalization naturally yields better performance under non-IID conditions. In this study, the severity of domain divergence, stemming from differences in patient populations, clinical protocols, and data schemas appeared to limit the models' ability to leverage richer local information. The additional complexity of PFL methods did not compensate for the fragmentation across data distributions. Despite decent ROC-AUC scores, model outputs were poorly calibrated, and classification performance at standard thresholds remained unreliable. This highlights a misalignment between ranking performance and prediction quality at the standard threshold, which is particularly problematic in imbalanced settings.

Together, these results reinforce concerns in the FL literature that statistical heterogeneity, particularly in real-world domains, can severely degrade generalization and limit the effectiveness of personalization strategies.

The experimental setup was intentionally designed to highlight the contrast between minimal global feature access and enriched local personalization. However, the inability of PFL to outperform global baselines, even when given broader feature access, raises important questions about its practical value under real-world constraints.

These findings point to the need for more fundamental strategies such as robust data harmonization, model calibration, or improved architectures before personalization can deliver meaningful benefits in heterogeneous healthcare environments.

6.2 Interpretation in relation to research questions

In this section, a detailed interpretation of the empirical findings is presented, directly addressing the core research question and its sub questions outlined in Section 1.4. The analysis synthesizes observations from the experimental results (Chapter 5), methodological design (Chapter 3), experimental setup in Chapter 4, and existing SOTA (Chapter 2) to provide a comprehensive understanding of PFL's performance, stability, and efficiency when applied to heterogeneous healthcare datasets.

6.2.1 Sub question 1: PFL's predictive performance with richer local data

Despite having access to a much richer feature set, PFL models did not show statistically significant performance gains over global FL baselines using only six static features. As detailed in Chapter 4, this restriction reflected a realistic constraint on shared schemas due to high missingness. Further validating the foundational capabilities of the implemented pipelines, experiments were conducted on the MNIST dataset (Subsections 5.1.2 and 5.1.1).

In a controlled and IID setting using MNIST, both FedAvg and FedProx achieved very high macro-averaged ROC-AUC scores of 0.9994 and 0.9995 respectively. This confirms that the FL pipeline and MLP architecture are capable of achieving SOTA results and converging efficiently under ideal conditions. The significant drop in performance when applying the same MLP and FL pipeline to the healthcare datasets strongly suggest that the primary challenge stemmed from data heterogeneity and incompleteness, not from implementation flaws or algorithmic weaknesses.

The aggressive feature filtering of limited global models to six static inputs was designed to maximize the information gap that PFL models were expected to bridge. By granting PFL access to a more comprehensive, client-specific feature space, the goal was to test whether this richer local context could compensate for the severe lack of shared global information and lead to superior predictive performance.

Both FO Per-FedAvg and the MOCHA-inspired PFL model performed well on MNIST under disjoint label splits. FO Per-FedAvg achieved strong per-class F1-scores and macro ROC-AUC above 0.97, validating that the adaptation and label remapping logic were correctly implemented. Similarly, the MOCHA-inspired model converged to near 1.0 accuracy with minimal loss (see Figure 5.13 and 5.14).

These results rule out implementation issues and confirm that the PFL pipelines perform as expected under controlled heterogeneous conditions. However, this success did not translate to the clinical datasets, reinforcing that real-world domain shifts, driven by schema divergence, missingness, and clinical variability are far greater challenges than synthetic label splits on controlled datasets.

In contrast, on MIMIC-III and eICU, PFL models were configured to leverage the full, more extensive local feature sets available to each client, including hundreds of additional client-specific static variables and dynamic time-series features unique to each hospital's domain.

Despite access to this wealth of additional, personalized information, PFL models did not demonstrate a statistically significant improvement in predictive performance over the heavily restricted global baselines. For instance, while a centralized LSTM + MLP model (which leveraged all static and dynamic features) achieved high ROC-AUC of 0.8458 for MIMIC-III and 0.7738 for eICU, showing the potential of the full feature set in a centralized setting, personalized FL models struggled to translate this local information advantage into federated improvement.

The ROC-AUC for PFL models, such as 0.7194 ± 0.00 for FO Per-FedAvg and 0.6193 ± 0.1094 for MOCHA-inspired both on MIMIC-III, while sometimes numerically higher than global FL models (0.6478 ± 0.0111 for FedAvg on MIMIC-III), were not statistically different. Thus, any observed numerical gains from PFL were not robust across seeds.

Beyond the ROC-AUC, a deeper analysis reveals a critical limitation, as both global and PFL models consistently yielded near-zero precision, recall, and F1-scores when evaluated at a 0.5 threshold. This reflects a systemic failure to produce well-calibrated binary predictions. In several cases, models collapsed into predicting only the minority class (mortality), likely due to the overly aggressive use of `pos_weight`, which skewed the loss surface, as previously mentioned.

These findings suggest that the extreme reduction of the common feature space, combined with the heterogeneity of real-world clinical data, imposed constraints that personalization could not overcome. The inability of PFL to significantly outperform highly limited global models underscores a central limitation when domain shifts are too severe, client specific models may fail to generalize or transfer meaningful representations. This challenges the widely held assumption that more personalized, client-rich information will necessarily lead to better model performance in heterogeneous settings.

6.2.2 Sub question 2: Client-specific recovery on heterogeneous datasets

This analysis evaluated the effectiveness of PFL methods in recovering client-specific performance on heterogeneous healthcare datasets, and whether any observed improvements where statistically significant compared to a single global model. This investigation revealed contrasting outcomes in terms of model stability and performance translation.

FO Per-FedAvg, while effective and stable under synthetic conditions like MNIST, exhibited significant instability during training on the clinical datasets, limiting its effectiveness in achieving robust client-specific performance recovery. As illustrated in Subsection 5.2.4, the ROC-AUC scores for both MIMIC-III and eICU fluctuated across the 50 communication rounds, with no clear sign of convergence. This behavior aligns with known limitations of first-order approximations in meta-learning, as discussed in the SOTA (Subsection 2.2.1), which often struggle in highly heterogeneous environments. Instability may have been made worse by architectural mismatches between clients, as FO Per-FedAvg implicitly assumes aligned model structures, which is an assumption violated here due to differing feature spaces.

In contrast, the MOCHA-inspired model demonstrated smoother convergence and more stable learning curves for both clients. Subsection 5.2.4 showed how the learning curves for both clients exhibited clearer trends, with MIMIC-III performance steadily improving before a slight decline, and eICU maintaining a consistent ROC-AUC. This was due to its architecture, which separates private heads from a shared backbone, described as an MTL-inspired design that supports coordinated yet decoupled personalization. This structure likely mitigated client drift more effectively than FO Per-FedAvg.

However, this stability did not translate into statistically significant improvements. This is a critical observation for the effectiveness of client-specific recovery. As reported in Section 5.3, no performance differences were statistically significant. This outcome suggests that while the MOCHA-inspired architectural approach was valuable in mitigating client drift and promoting stable learning in the face of heterogeneity, the absence of the full MOCHA optimization strategy, which enables learning of inter-client relationships, likely limited its ability to truly leverage inter-client similarities for performance enhancement.

The SOTA (Subsection 2.3.3) notes that the full MOCHA framework explicitly learns task relationships to enable information transfer even when distributions share no overlap. Without this, the simplified setup could not fully exploit shared structure or guide personalized models toward better generalization.

Finally, as validated through the MNIST experiments, the MOCHA-inspired pipeline is functionally sound under controlled heterogeneous conditions. Its lack of statistically significant gains in the clinical datasets, therefore, underscores the real-world challenges of domain shifts, rather than any implementation flaws.

6.2.3 Sub question 3: Trade-offs in communication efficiency and convergence

This sub question examines the empirical trade-offs between PFL methods and traditional FL in terms of communication efficiency and convergence speed. The experimental evaluation, conducted over 50 communication rounds, provided clear contrasts between FO Per-FedAvg and the MOCHA-inspired model.

While FO Per-FedAvg converged efficiently on MNIST under disjoint label splits, it exhibited significant communication inefficiency on the clinical datasets. As shown in Subsection 5.2.4, the model failed to converge, with ROC-AUC scores for both MIMIC-III and eICU fluctuating widely across rounds.

Despite being designed for computational efficiency through simplified local gradient steps, the method consumed substantial resources without yielding a stable model within the set round budget (Table 5.17). In practice, this would require far more communication rounds or extensive hyperparameter tuning to reach convergence. Indeed, the original authors of Per-FedAvg ran their experiments for 100–200 rounds [4], underscoring its inefficiency in real-world deployment scenarios.

This non-convergence is consistent with known limitations of FO Per-FedAvg methods (Subsection 2.2.1), which often struggle in highly heterogeneous, non-convex loss landscapes. The training curves show unstable trajectories, confirming that FO Per-FedAvg trades convergence stability for algorithmic simplicity. More robust variants such as HF Per-FedAvg attempt to address this by incorporating second-order information, improving gradient direction and potentially accelerating convergence in non-IID settings.

In contrast, the MOCHA-inspired model demonstrated a more favorable trade-off. As shown in Subsection 5.2.4, it exhibited comparatively smoother training dynamics and converged more stably within the same 50 rounds. However, performance on MIMIC-III began to decline after peaking mid-training, suggesting that the convergence was not sustained. This instability may indicate overfitting to sparse local patterns or poor generalization due to extreme inter-client divergence.

FedAvg and FedProx showed stable convergence on MNIST, but on the clinical datasets, their learning curves remained flat, suggesting convergence without meaningful optimization which shows a sign of weak signal extraction. The MOCHA-inspired model's relative stability likely stems from its shared-backbone and private-heads architecture, which mitigates client drift.

This highlights how architecture can support stable training under heterogeneity, though not necessarily improve performance when the task is inherently hard.

6.3 Methodological limitations

This section covers the methodological limitations that this study was met with, which have significantly influenced the experimental outcomes. First, it begins with fundamental constraints that affected all results, followed by data-specific challenges such as class imbalance. It then addresses structural issues in the experimental setup itself, and finally, it discussed the most critical limitation, which is feature space constraints and the resulting failure of personalization to bridge the information gap.

6.3.1 Fundamental constraints

Several fundamental limitations affected all experimental results presented in this thesis. Most critically, no extensive hyperparameter tuning (like grid search or random search) was performed due to computational constraints. Thus, all models were trained using fixed configurations (Chapter 4), which likely resulted in suboptimal performance. This limitation was particularly impactful for FO Per-FedAvg since it is known to be highly sensitive to learning rates, step sizes, and the specific data splits used during meta-training [4].

Moreover, each federated setup (both FL and PFL) was only repeated across three random seeds. While this allowed for basic variance estimation, it is insufficient to capture the full range of outcome variability under high heterogeneity, especially in complex datasets like MIMIC-III and eICU. So, as a result, the statistical power of the comparisons is limited, and observed performance differences should be interpreted cautiously.

Finally, all experiments were constrained to a maximum of 50 communication rounds and subject to early stopping. Personalized methods, in particular, often require longer meta-training or adaptation phases, typically ranging from 100 to 200 rounds in the literature [4] to reach stable convergence. The imposed round limit may have prevented full convergence, further weakening the reliability and generalizability of the results.

Taken together, these constraints limit the robustness of the findings and likely contributed to the absence of statistically significant improvements from the personalization strategies tested.

6.3.2 Data related challenges

The quality and structure of the input data introduced several challenges that significantly influenced model performance across training setups. The most critical issue here was severe class imbalance in both MIMIC-III and eICU, where positive cases (death) represented a small minority. To mitigate this, a `pos_weight` term was applied in the loss function (Chapter 4), computed as the ratio of negative to positive samples resulting in values of approximately 0.07 for MIMIC-III and 0.08 for eICU.

While this is a standard and mathematically correct approach in imbalanced settings, it led to unintended side effects in specific configurations, particularly when models were limited to static features. In several cases, this contributed to prediction collapse, where models consistently predicted only one class, resulting in zero precision, recall, and F1-score, despite still achieving high ROC-AUC and PR-AUC values (Chapter 5).

This behavior occurred across centralized, FL, and PFL models, suggesting that class imbalance was a significant factor, but not the sole cause.

The effect was likely amplified by model design choices and miscalibrated prediction probabilities. Notably, both the LSTM + MLP baseline and the MOCHA-inspired architecture were more resilient and achieved non-zero classification metrics (Tables 5.4 and 5.16).

One contributing factor may have been the use of a fixed classification threshold of 0.5 across all models (Chapter 3), which is particularly problematic under imbalance.

Poorly calibrated outputs, when evaluated against this static threshold, likely led to systematic underprediction of the minority class.

More sophisticated calibration techniques or adaptive thresholding might have mitigated these issues, but were out of scope due to resource constraints. Nonetheless, the results highlight how model design and evaluation choices can significantly impact performance under class imbalance, and sometimes even more than the learning paradigm itself.

6.3.3 Structural limitations and feature space constraints

Beyond data and algorithmic factors, structural design choices in the experimental setup introduced key limitations that shaped the interpretation and generalizability of results.

First, the FL and PFL experiments were restricted to two clients (MIMIC-III and eICU), selected to reflect realistic institutional heterogeneity. While this aligns with typical cross-silo settings (as noted in [8]), it limits the ability to generalize findings to broader federated environments with more diverse client populations and possible clustering effects.

Preprocessing was performed locally on each dataset, using identical procedures but without enforcing semantic feature alignment. This was intended to preserve autonomy and scalability, drawing inspiration from the MIMIC-Extract framework [20] (Section 3.3). However, since variable selection was based solely on local missingness rates, no semantic harmonization was performed aside from the inclusion of a few prominent features mentioned in MIMIC-Extract. Hence, only six static features overlapped between the two datasets, effectively constraining global FL models to a minimal input space.

In contrast, personalized models had access to rich, client-specific feature spaces, including static and dynamic variables. While this allowed personalization methods to model more detailed patterns locally, it introduced a structural asymmetry that makes direct comparison with global models difficult. Some apparent gains in PFL may reflect input richness, not necessarily improved personalization logic.

This imbalance was evident in Chapter 5, where global MLPs trained on static inputs often failed to produce meaningful predictions, while LSTM + MLP and MOCHA-inspired PFL models, both operating on richer features achieved more stable outputs. However, differences in architecture and input space make it difficult to attribute improvements to personalization in isolation. Moreover, statistical testing confirmed that richer local data alone was insufficient to improve outcomes.

Overall, the unequal access to feature spaces represents a major confounding factor. While it reflects real-world deployment constraints, it also limits interpretability. Future work may benefit from either harmonizing a richer global feature set or explicitly isolating the effect of feature depth from personalization capacity.

6.4 Comparison to related work

Most personalization methods in the literature are evaluated on controlled synthetic non-IID settings (commonly using CIFAR-10 or MNIST with label or feature skew). These setups assume that clients have shared feature spaces and label meanings, which differs significantly from real-world healthcare data. Studies like Fallah *et al.* (2020) report large gains using HF Per-FedAvg, for instance, 71.25% accuracy against a 58.59% accuracy for FedAvg under these conditions [4].

In contrast, this thesis evaluates PFL under a real-world cross-silo setup with two structurally distinct healthcare datasets (MIMIC-III and eICU), exhibiting severe differences in data structure, schema, and population.

Specifically, only six features were aligned across over 200 variables, creating extreme heterogeneity beyond what is typically modeled in benchmark studies. This divergence in data structure partly explains why methods like FO Per-FedAvg and the MOCHA-inspired model did not significantly outperform global baselines.

While the implementations in this thesis were validated under synthetic conditions (using MNIST), the assumptions that the PFL techniques FO Per-FedAvg and MOCHA relies on, such as shared representations and transferable patterns, broke down under real-world heterogeneity.

Additionally, methods like FedProx showed no meaningful improvement over FedAvg, consistent with literature [9], indicating its limited benefits under structural feature misalignment. FO Per-FedAvg also exposed its well-known sensitivity to hyperparameters and convergence instability when tuning is limited, as noted in [4].

Moreover, despite the MOCHA-inspired architecture offering more stable learning curves, it lacked the full components, such as adaptive task weighting and second-order dual optimization. These simplifications, though common in practice, likely limited its ability to fully exploit inter-client relationships in this highly divergent setup.

Overall, the findings highlight a gap between the synthetic evaluations commonly used in PFL research and the complex realities of healthcare data. They suggest that robustness to schema divergence, calibration, and architectural flexibility may be more critical than personalization alone in real-world federated settings.

6.5 Implications for real-world practice

The findings of this thesis demonstrate that federated personalization is not a plug-and-play solution, particularly in cross-silo healthcare environments characterized by structural heterogeneity. In this study, even highly constrained global FL models, trained on only six harmonized features, performed comparably to more complex PFL approaches. This directly challenges the dominating narratives in existing literature, where PFL often delivers substantial gains in more controlled, synthetic non-IID setups.

In real-world deployments, the assumption of a shared representational foundation across institutions does not always hold. When clients differ not only in data distribution, but also in schema, feature definitions, and available variables, exactly like the case with MIMIC-III and eICU datasets, the potential benefits of personalization start diminishing.

These results suggest that in practice, PFL must be approached with greater caution and contextual awareness. The outcomes presented in Chapter 5 indicate that, in certain real-world settings, greater value may lie in developing robust global models using fully harmonized and semantically aligned features, rather than defaulting to personalization methods.

More importantly, institutions considering FL must first assess their data for alignment in label definitions, schema compatibility, and feature distributions. Without this foundational compatibility, federated collaboration (particularly when using simplified personalization strategies adapted for constrained environments, as explored in this thesis) is unlikely to yield meaningful performance improvements.

6.6 Future directions

This section outlines future directions for this experiment and research, building upon the insights gained from the current study regarding the effectiveness and trade-off of the FL and PFL in highly heterogeneity healthcare environments.

To advance the capabilities of the PFL, several key areas open up for further investigation such as:

- Enhancing model stability and personalization depth
- Addressing Scalability and fundamental heterogeneity
- Harmonizing data schemas and features
- Improving performance for imbalanced data and clinical utility

The observed instability of FO Per-FedAvg and the MOCHA-inspired model suggest the need for more robust PFL methods. Future work should focus on implementing more advanced meta-learning techniques such as HF Per-FedAvg which might provide greater stability through the use of second-order information.

Additionally, a full implementation of the MOCHA optimization strategy, including the mechanism for learning inter-client relationships, could enable a deeper and more effective personalization by leveraging shared structure across diverse clients.

The current study utilized two highly heterogeneous clients. Expanding experiments to larger federations would allow evaluation of scalability and the impact of varying degrees of heterogeneity In such settings exploring the clustered FL strategies becomes crucial. Such techniques group clients with similar data distributions to train specific models for each cluster, potentially offering a more tailored approach to model training that can better accommodate the severe domain shift observed in real-world healthcare data.

The feature filtering applied in this thesis created an information-poor baseline for global models, highlighting the challenges of working with raw, heterogeneous EHR data. Future work should focus on the architectural and statistical alignment before personalization. This includes implementing robust data strategies across clients and potentially leveraging standardized medical terminologies and data structuring methods to ensure consistent representation of clinical concepts and data schemas. Such upfront data engineering would reduce inherent heterogeneity and potentially enable FL models, both global and personalized, to utilize a more comprehensive set of features consistently across institutions.

A critical limitation observed in the results was the near-zero precision, recall and F1-scores across several models. This indicates a systemic failure to convert probabilistic outputs into reliable binary predictions, even though decent class separation was reflected in ROC-AUC values. Future work must address this by integrating threshold optimization, better calibration strategies, and FL-compatible techniques for mitigating class imbalance, such as adaptive loss weighting or federated oversampling. Improving calibration and decision boundary placement will be critical to enhance practical utility in real-world clinical tasks.

While patient privacy was preserved by ensuring raw data remained local, formal privacy-enhancing technologies were not implemented in this study. Future work should integrate techniques to improve privacy guarantees. This would add a crucial layer of protection against potential data inference attacks from shared model updates, which is essential for the ethical and secure deployment of FL in sensitive healthcare domains.

Ultimately, the findings of this thesis underscore the complexity of deploying FL in real-world healthcare scenarios. They reveal where current FL methods encounter significant limitations when confronted with deep structural and statistical heterogeneity. Future research should emphasize a co-design approach, where algorithms, data engineering strategies, and privacy techniques are developed in concert.

This integrated approach will be essential to overcome the challenges identified and realize the full potential of FL for robust and effective healthcare applications.

6.7 Key insights

This chapter has shown that personalization alone does not guarantee better performance under extreme cross-silo heterogeneity. Even with access to richer local data, personalized FL models failed to significantly outperform restricted global baselines. These findings underscore the need to address structural heterogeneity such as misaligned feature schemas and limited shared representations before personalization can be effective. So, architectural design alone is insufficient when the data lacks meaningful overlap.

7 Conclusion

This thesis tested the effectiveness of PFL in addressing data heterogeneity in real-world healthcare environments. Despite the substantial gains reported in controlled benchmark studies, PFL failed to significantly outperform simpler global FL baselines when evaluated on two institutionally distinct ICU datasets (the MIMIC-III and eICU). Statistical analysis across multiple runs showed no meaningful performance advantage for the PFL methods. These findings directly answer the core research question: ”How can PFL improve model robustness and stability when handling heterogeneous and missing data in healthcare?”. However, in its current form, PFL does not reliably improve model robustness or stability under the real-world constraints of healthcare data.

The experimental setup intentionally constrained global models to only six shared static features, while allowing PFL models access to hundreds of client-specific static and dynamic features. This design aimed to test whether personalization could leverage this richer local context. However, the Wilcoxon signed-rank test showed that the observed numerical improvements in the PFL models were not statistically significant across three different random seeds, indicating that the performance gains could not be confidently attributed to the personalization strategy. This challenges the intuitive assumption that more client-specific information necessarily leads to improved model quality under severe inter-client heterogeneity.

Moreover, a critical insight emerged from the evaluation metrics. Across most model variants (centralized, global FL, and PFL) precision, recall, and F1-score were close to zero, despite acceptable ROC-AUC values. This revealed that models consistently failed to make balanced predictions and, in several cases, collapsed into predicting only the minority class (mortality). This led to near-zero accuracy, precision, and recall, despite acceptable ROC-AUC scores. The collapse was largely driven by aggressive `pos_weight` settings intended to correct for class imbalance, which unintentionally skewed the loss function toward over-predicting mortality. The issue was further enhanced by minimal feature overlap across clients, limited hyperparameter tuning, and only three evaluation seeds. These factors particularly impacted FO Per-FedAvg, which is known to be highly sensitive to learning rate and meta-step size.

Nevertheless, despite these constraints, the FL pipeline and MLP architecture demonstrated SOTA performance and efficient convergence on MNIST, confirming their functional correctness under IID settings. The performance drop on real healthcare data highlights that the primary challenge stemmed from structural and statistical heterogeneity, not from implementation flaws. Furthermore, the instability of FO Per-FedAvg and the MOCHA-inspired model’s inability to yield significant gains on clinical data, even after performing well on MNIST, suggests that the assumptions of shared structure and transferable patterns in PFL break down in truly diverse cross-silo environments.

In conclusion, this thesis reveals that in healthcare settings marked by deep heterogeneity, federated personalization as implemented here is not simply a ”take it out of the box” solution. The findings, aligned with the project title ”Federated learning for tackling data heterogeneity in healthcare applications” show that effective FL collaboration demands a more foundational approach. This includes robust schema harmonization, careful calibration, and engineered feature alignment before personalization can be meaningfully deployed. These conclusions directly address the sub-research questions on the degree of PFL improvement, its ability to support client-specific recovery, and the trade-offs in convergence stability and communication efficiency.

Bibliography

[1] Brownlee, and Jason. Failure of accuracy for imbalanced class distributions, 2020.

[2] Caballar, R. D., and Stryker, C. What is meta learning?, 2024.

[3] Dagang, W. Essential math for machine learning: Confusion matrix, accuracy, precision, recall, f1-score, 2024.

[4] Fallah, A., Mokhtari, A., and Ozdaglar, A. Personalized federated learning: A meta-learning approach, 2020.

[5] GeeksforGeeks. Introduction to multi-task learning (MTL) for deep learning, 2023.

[6] Harutyunyan, H., Khachatrian, H., Kale, D. C., Ver Steeg, G., and Galstyan, A. Multitask learning and benchmarking with clinical time series data. *Scientific Data 6*, 1 (June 2019).

[7] Johnson, A. E. W., Pollard, T. J., and Mark, R. G. MIMIC-III Clinical Database (version 1.4), 2016.

[8] Kairouz, P., Mcmaha, B., Avent, B., Bellet, A., and Bennis, M. *Advances and Open Problems in Federated Learning*, vol. 14 of *Foundations and Trends® in Machine Learning*. 2021.

[9] Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. Federated optimization in heterogeneous networks, 2020.

[10] Lv, K., Ye, R., Huang, X., Yang, J., and Chen, S. Learn what you need in personalized federated learning, 2024.

[11] McMahan, H. B., Moore, E., Ramage, D., and y Arcas, B. A. Federated learning of deep networks using model averaging. *CoRR abs/1602.05629* (2016).

[12] Mudadla, S. What is the main difference between the t-test and the wilcoxon test in terms of their assumptions? https://medium.com/@sujathamudadla1213/what-is-the-main-difference-between-the-t-test-and-the-wilcoxon-test-in-terms-of-their-assumptions-ebff4c8385aa, 2023. Accessed: 2025-06-10.

[13] N, D., Sachin, B, Annappa, Hegde, Saumya, Abhijit, Sri, C., and Ambesange, S. Fedcure: A heterogeneity-aware personalized federated learning framework for intelligent healthcare applications in iomt environments. *IEEE Access 12* (2024), 15867–15883.

[14] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems* (2019), pp. 8024–8035.

[15] Pollard, T. J., Johnson, A. E. W., Raffa, J. D., Celi, L. A., Badawi, O., and Mark, R. G. eICU Collaborative Research Database (version 2.0), 2019.

[16] Radečić, D. Roc and auc – how to evaluate machine learning models in no time, dec 2020.

[17] Richardson, T. G., Carter, A., Nielsen, M., and Peters, B. The receiver operating characteristic curve accurately assesses imbalanced datasets. *Patterns 5*, 6 (2024), 100994.

[18] Smith, V., Chiang, C.-K., Sanjabi, M., and Talwalkar, A. Federated multi-task learning. *NeurIPS Proceedings* (2017).

[19] Tan, Ziying, A., Yu, Han, Lizhen, C. A., Yang, and Qiang. Towards personalized federated learning. *IEEE Transactions on Neural Networks and Learning Systems 34*, 12 (2023), 9587–9603.

[20] Wang, S., McDermott, M. B. A., Chauhan, G., Hughes, M. C., Naumann, T., and Ghassemi, M. Mimic-extract: A data extraction, preprocessing, and representation pipeline for MIMIC-III. *CoRR abs/1907.08322* (2019).

[21] Yang, L., Bahadori, M. T., Liu, Y., and Sun, J. Rethinking mortality prediction in intensive care units using deep learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining* (2023), ACM, pp. 2580–2588.

[22] Yang, Q., Liu, Y., Chen, T., and Tong, Y. Federated machine learning: Concept and applications, 2019.

[23] Zeng, D., Wu, Z., Liu, S., Pan, Y., Tang, X., and Xu, Z. Understanding generalization of federated learning: the trade-off between model stability and optimization, 2025.

[24] Zhao, Yue, Li, Meng, Lai, Liangzhen, Suda, Naveen, Civin, Damon, Chandra, and Vikas. Federated learning with non-iid data.

A Appendix

A.1 Evaluation metrics equations

$$TPR = \frac{TP}{TP + FN} \tag{A.1}$$

$$FPR = \frac{FP}{FP + TN} \tag{A.2}$$

True Positive Rate (TPR) and False Positive Rate (FPR) equations, both A.1 and A.2 are cited from source [16].

$$\text{Precision} = \frac{TP}{TP + FP} \tag{A.3}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{A.4}$$

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{A.5}$$

Precision, recall and F1 score equations A.3, A.4 and A.5 are all from the source [3].

Through rigorous investigation into federated learning for healthcare, this work highlights both the complexities and critical necessity of tackling data heterogeneity. It envisions a future where secure, collaborative AI models bridge institutional divides, transforming patient care without compromising data privacy

Technical
University of
Denmark

DTU Compute - Building 324-241
2800 Kgs. Lyngby
Tlf. 4525 1700

https://www.dtu.dk/uddannelse/bachelor/uddannelsesretninger/kunstig-intelligens-og-data