



Heterogeneous Federated Learning

Fairness and Client Behaviour Exploration

Tharuka Kasthuri Arachchige

Department of Computer Science
Blekinge Institute of Technology

Licentiate Dissertation Series no. 2026:03

Heterogeneous Federated Learning

Fairness and Client Behaviour Exploration

Tharuka Kasthuri Arachchige

Blekinge Institute of Technology
Licentiate Dissertation Series No. 2026:03

Heterogeneous Federated Learning

Fairness and Client Behaviour Exploration

Tharuka Kasthuri Arachchige

Licentiate Dissertation in Computer Science



Department of Computer Science
Blekinge Institute of Technology
SWEDEN

Copyright pp Tharuka Kasthuri Arachchige
Paper 1 © 2024, IEEE
Paper 2 © Springer Nature Switzerland AG, 2026
Paper 3 © By the authors (Manuscript Unpublished, Under Evaluation)

Blekinge Institute of Technology
Department of Computer Science

Blekinge Institute of Technology Licentiate Dissertation 2026:03
ISBN 978-91-7295-525-7
ISSN 1650-2140
urn:nbn:se:bth-urn:nbn:se:bth-29327

Printed in Sweden by Media-Tryck, Lund University, Lund 2026



Media-Tryck is a Nordic Swan Ecolabel
certified provider of printed material.
Read more about our environmental
work at www.mediatryck.lu.se

MADE IN SWEDEN 

Dedication

මම ගෙදර එනතුරු මඟ බලන තාත්තීට...
මා යන මඟ නොවරදින්න යැයි දෙවියන්, බුදුන් වදින අම්මාට...

To my father, who waits for me to return home...

To my mother, who prays that I may not lose my way...

Progress comes from persistence, even when results are not immediately visible.

Abstract

Federated Learning (FL) is a distributed learning paradigm that enables multiple clients to collaboratively train a shared model without centralizing their data. This design supports learning in decentralized, heterogeneous, and data-constrained settings while providing privacy benefits by keeping raw data local. However, in practical implementations, client data are typically non-independent and identically distributed (non-IID). This resulting in heterogeneous learning dynamics and unequal benefits across participants. Improvements in average global performance can mask performance degradation for disadvantaged clients, highlighting a structural fairness challenge in FL. This thesis argues that achieving fairness under non-IID FL requires explicit understanding and modeling of client behavioral heterogeneity rather than uniform aggregation of client updates.

In addressing the issue of fairness in FL under data heterogeneity, the thesis first studies and analyzes clients' deviating behavior during the federated training process. An eccentricity-based approach is introduced to quantify deviations in local models and data representations within the global model, enabling systematic identification of atypical contribution and benefit patterns. The insights gained lay the foundation for our further research into developing novel, fairness-aware FL solutions for heterogeneous, distributed learning setups.

Then it proposes a fairness-aware aggregation framework called FEDABOOST that adapts client influence based on local performance signals. By dynamically weighting client updates and adjusting local optimization to emphasize hard examples, the method reduces disparities across heterogeneous clients while maintaining competitive global performance. Later, the thesis introduces DEFFT, a clients distribution-aware framework that models latent similarities among clients through persistent grouping based on label distributions. Cluster-level models and hierarchical knowledge distillation integrate inter-client structure into the learning process, enhancing fairness metrics along with overall accuracy.

Across multiple benchmark datasets, the proposed approaches demonstrate that a principled way to modeling heterogeneity can lead to measurable improvements in fairness without compromising global performance. The three discussed studies together establish a structured framework for mitigating unequal benefits in FL under non-IID data distributions.

Acknowledgements

I would like to express my sincere gratitude to all those who have supported me throughout my licentiate studies.

First, I am deeply thankful to my examiner, *Veselka*, for her continuous support, patience, and valuable guidance. Her understanding and encouragement have meant a great deal to me. I would like to extend my sincere gratitude to my supervisor, *Shahrooz*, for his invaluable guidance and continuous support throughout this work. I am especially grateful for his willingness to listen and support me beyond the immediate scope of this research. I would also like to thank my supervisors, *Håkan* and *Emiliano*, for their guidance, constructive feedback, and support during the course of this research. Furthermore, I would like to thank *Selim Ickin* for his collaboration and contributions as a co-author for my first paper. All of yours input has been important to the development of this work.

I would also like to acknowledge the financial support that made this research possible. This work was partly funded by the *Knowledge Foundation*, Sweden, through the HINTS project. I am grateful to my colleagues and fellow PhD students at BTH for providing a stimulating and supportive research environment. The discussions, shared experiences have contributed positively to both my academic and personal development. Also I would like to thank all those who, directly or indirectly, have contributed to this work.

I would also like to thank *Eugene Charles*, my bachelor's thesis supervisor, for encouraging me to apply for a PhD position. Working under his supervision was my first experience with research, and it played a key role in opening the door to this journey. I would also like to sincerely thank *Jason Mars* for his encouragement and support. At a time when I was uncertain, his advice and belief in my ability to continue gave me the motivation to move forward.

I am deeply grateful to my parents for their unwavering support, for always believing in me, and for standing by me in every decision I have made. I also thank my brothers for taking on the responsibility of caring for our parents while I pursue my dreams far from home.

I am grateful to my friends for their companionship and support throughout these years, which made this journey less isolating and more meaningful. I also extend my sincere thanks to those who supported me during difficult moments, helping me remain resilient and move forward through a challenging period in my personal life.

List of Papers

Paper I

Tharuka Kasthuri Arachchige, Selim Ickin, Shahrooz Abghari and Veselka Boeva. Clients Behavior Monitoring in Federated Learning via Eccentricity Analysis. IEEE International Conference on Evolving and Adaptive Intelligent Systems, IEEE EAIS 2024 (23–24 May 2024, Madrid, Spain).

Paper II

Tharuka Kasthuri Arachchige, Veselka Boeva and Shahrooz Abghari. FeDABoost: Fairness Aware Federated Learning with Adaptive Boosting. WAFL@ECML-PKDD 2025 (3rd Workshop on Advancements in Federated Learning collocated with ECML-PKDD 2025 in Porto, Portugal, 15-19 September 2025). I. Koprinska et al. (Eds.): ECML PKDD 2025, CCIS 2841, pp.1–16, 2026.

Paper III

Tharuka Kasthuri Arachchige, Veselka Boeva and Shahrooz Abghari. Hierarchical Knowledge Distillation for Fair Federated Learning. Manuscript submitted to a peer-reviewed international conference (2026).

Author's contribution to the papers

Paper I

The author co-defined the research problem, contributed to the development of the conceptual and methodological framework, implemented the full codebase, conducted all experimental work, co-interpreted the results, and prepared the majority of the manuscript.

Paper II

The author co-defined the research problem, developed the conceptual and methodological framework, implemented the full codebase, conducted all experimental work, interpreted the results, and drafted the manuscript.

Paper III

The author defined the research problem, developed the conceptual and methodological framework, implemented the full codebase, conducted all experimental work, interpreted the results, and drafted the manuscript.

Abbreviations

AI	Artificial Intelligence.
CFL	Clustered Federated Learning.
FL	Federated Learning.
IID	Independent and Identically Distributed.
KDE	Kernel Density Estimation.
ML	Machine Learning.
non-IID	non-Independent and Identically Distributed.
TEDA	Typicality and Eccentricity-based Data Analytics.

Table of Contents

Abstract	i
Acknowledgements	iii
List of Papers	v
Abbreviations	vii
Chapter 1 Introduction	1
1.1 Rationale	1
1.1.1 Heterogeneous Federated Learning	1
1.1.2 Fairness in Federated Learning	2
1.2 Aim and Objectives	2
1.3 Research Questions	3
1.4 Thesis Contributions	4
1.5 Thesis Outline	6
Chapter 2 Background	7
2.1 Federated Learning	7
2.2 Eccentricity Analysis	8
2.3 Hessian Matrix	9
2.4 Multiclass Ada Boosting	9
2.5 Focal Loss	10
2.6 Knowledge Distillation	10
2.7 Clustering	11
Chapter 3 Related Work	13
Chapter 4 Methodology	17
4.1 Research Methodology	17
4.2 Evaluation Framework	17
4.2.1 Federated Learning Setup	17
4.2.2 Assumptions and Constraints	18
4.2.3 Datasets	18
4.2.4 Baselines	19
4.2.5 Experimental Protocol	19
4.2.6 Evaluation Metrics	20
4.3 Threats to Validity	22
4.3.1 Internal Validity	22

4.3.2 Construct Validity	22
4.3.3 External Validity	23
4.3.4 Statistical Conclusion Validity	24
Chapter 5 Findings and Discussions	25
5.1 Study 1: Client Behavior Monitoring	25
5.1.1 Main Findings	26
5.1.2 Answer to RQ1	27
5.2 Study 2: Fairness in FL via Improving Under Performing Clients	28
5.2.1 Main Findings	29
5.2.2 Answer to RQ2.	29
5.3 Study 3: Leveraging Distributional Similarities to Improve	
Fairness	30
5.3.1 Main Findings	31
5.3.2 Answer to RQ3.	32
5.4 Summary and Key Takeaways	33
Chapter 6 Conclusion and Future Works	35
6.1 Conclusion	35
6.2 Future Work	36
Chapter 7 Experiences and Learning Outcomes	37
Bibliography	39
Paper I Clients Behavior Monitoring in Federated Learning via Eccentricity Analysis	43
1 Introduction	44
2 Background	45
2.1 Federated Learning	45
2.2 Contribution Evaluation via Deletion	46
2.3 Eccentricity Analysis	47
2.4 Neural Network Parameters and Hessian Matrix	47
2.5 Rank correlation	48
3 Related Work	48
3.1 Clients Contribution Evaluation in FL	49
3.2 Clients Behaviour Monitoring in FL	50
4 Methodology	51
4.1 Data	51
4.2 Problem Formulation and Eccentricity-based Method	51
4.3 Experimental and Training Setup	52
4.4 Applicability	53
5 Empirical Evaluation and Results	54
References	59
Paper II FedABoost: Fairness Aware Federated Learning with Adaptive Boosting	63
1 Introduction	64

2	Problem Formulation	65
3	Background	65
	3.1 Multi-class Adaptive Boosting	65
	3.2 Focal Loss for Challenging Cases	66
4	Methodology	66
	4.1 Aggregation Mechanism in FEDABOOST	66
	4.2 FEDABOOST Boosting Mechanism	68
	4.3 The proposed FEDABOOST algorithm	69
	4.4 FEDABOOST Fairness and Convergence	70
5	Experimental Setup	71
6	Evaluation and Results	73
7	Literature Review	76
	7.1 Model Aggregation in FL	76
	7.2 Fairness in FL	77
8	Conclusion and Future Directions	78
	References	78

Paper III Hierarchical Knowledge Distillation for Fair Federated Learning 81

1	Introduction	82
2	Problem Formulation	83
3	Related Work	83
4	The Proposed DEFFT Framework	85
	4.1 Hierarchical Client Organization	85
	4.2 Cluster-Level Aggregation and Global Model Updates	86
	4.3 Knowledge Distillation Across Hierarchical Levels	87
	4.4 Overall DEFFT Procedure	87
	4.5 Convergence Analysis of the Optimization Objective	88
5	Experimental Setup	90
6	Results and Discussion	92
	6.1 Control Study Results Analysis	92
	6.2 Evaluation on Benchmark Federated Datasets	94
7	Conclusion and Future Directions	95
	References	96

1 Introduction

1.1 Rationale

Machine Learning (ML) is a branch of Artificial Intelligence (AI) that develops algorithms to learn from data and make predictions or decisions without explicit programming. Traditional ML approaches involve centralized data collection from multiple sources to train a model in one location. Federated Learning (FL) is a distributed ML framework in which multiple client devices collaboratively train a shared global model under the coordination of a central server. Each client trains the model locally on its own data and shares only model updates (e.g., gradients or parameters), not the raw data. FL can be categorized into horizontal, vertical, and transfer settings. This thesis focuses on horizontal FL, in which clients share a common feature space but possess different local data samples.

The training process begins with the server initializing a global model with random parameters, and then sharing it with all participating clients or a selected subset of clients in the federation. The participants perform several local optimization steps on their own data and return the updated model to the server. The server aggregates the received model updates to form a new global model, which is then shared with the clients for the next communication round. This is an iterative process and continues until the global model converges. Federated aggregation implicitly assumes client updates are sufficiently compatible to be meaningfully combined; however, in practice, under heterogeneous data distributions, this assumption rarely holds.

1.1.1 Heterogeneous Federated Learning

In this thesis, we focus on statistical heterogeneity arising from non-Independent and Identically Distributed (non-IID) data distributions across clients. Horizontal FL commonly exhibits non-IID data distributions. Although clients share a common feature and label space, each client collects data from different users, contexts, and operating conditions, resulting in systematically different underlying distributions across clients. Consequently, each client optimizes a distinct empirical objective, and local updates often diverge in different directions within the parameter space. When such heterogeneous client updates are aggregated, gradient interference arises, which destabilizes global model training and slows convergence. Furthermore, standard

aggregation mechanisms often assign weights to client updates based on the size of their local datasets. As a result, clients with larger datasets have a greater impact on the global model, which biases the global model toward dominant data distributions.

1.1.2 Fairness in Federated Learning

As a consequence of the biased optimization process caused by heterogeneity of clients' data distributions, FL systems often show uneven optimization results across clients. Improvements for some participants may occur at expense of some other clients. In such cases, evaluating solely the average global accuracy is inadequate, as it masks significant performance disparities and does not ensure acceptable behavior for individual clients. This introduces the concept of performance fairness in FL, aiming to reduce dissimilarities in model performance among clients rather than solely optimizing a single global objective. While several fairness-aware FL methods have been proposed to address this issue, they remain limited. Most of the proposed approaches rely on heuristic adjustments to client aggregation weights, modification of global objective, or additional regularization terms during local optimization [1], [2]. Some methods further adopt personalized models coupled with regularization to constrain deviation from the shared global representation [3]. While these strategies can mitigate performance disparities to some extent, they lack principled guarantees and often introduce trade-offs between fairness and overall model performance.

Taken together, heterogeneity in FL therefore presents two fundamental challenges. First, the decentralized training process provides limited visibility into how each client behaves and influences the evolution of the global model. Second, existing aggregation mechanisms offer limited principled control over the performance disparities caused by non-IID data distributions. These challenges cannot be resolved through ad hoc adjustments alone. Addressing these challenges requires systematic methods to characterize client behavior under heterogeneous conditions and to design FL training approaches that explicitly account for such heterogeneity.

1.2 Aim and Objectives

The aim of this thesis is to develop systematic methods for understanding and leveraging client heterogeneity in FL to enhance convergence behavior, improve global model performance, and reduce performance disparities across clients under non-IID data conditions.

To achieve this aim, the thesis follows a structured progression across three interconnected stages. It begins by establishing a principled approach for characterizing client behavior during training, enabling the identification of deviations in learning dynamics without access to raw data. Building on this diagnostic foundation, the thesis then

investigates how such heterogeneity can be mitigated to reduce performance disparities across clients while maintaining overall model effectiveness. Finally, the thesis advances toward leveraging the underlying structure of heterogeneity by exploiting similarities among clients to further improve learning dynamics and fairness.

Based on this progression, the thesis addresses the following objectives:

Ob1. Systematically characterize client behavioral deviations during federated training without access to raw client data.

This objective focuses on developing mechanisms based on client model updates to monitor heterogeneous client behavior across training rounds and quantify deviations in learning dynamics. The goal is to establish measurable indicators that relate client behavior to its influence on, and benefit from, the federation.

Ob2. Understand, mitigate, and leverage the effects of heterogeneity to improve performance and fairness in non-IID federated learning.

This objective focuses on investigating how statistical heterogeneity (non-IID data) leads to unequal client-level performance during federated training, even when global performance improves. It aims to identify how these effects can be mitigated or exploited to improve both global model performance and fairness across clients.

1.3 Research Questions

The objectives outlined above define the operational goals of this thesis. To structure the investigation and assess whether these objectives have been met, they are formalized into the following research questions.

RQ1. *How can client behavioral deviations in federated learning be systematically characterized and monitored during training?*

FL training is inherently partially observable, as the central server receives model updates without access to client data or internal optimization processes. As a result, deviations in client behavior such as drift, instability, or atypical contribution patterns are difficult to detect and quantify. Existing approaches primarily rely on performance-based metrics, which provide limited insight into the underlying learning dynamics of clients.

This research question investigates how client behavioral deviations can be systematically characterized and monitored during training, beyond outcome based evaluation, by capturing differences in how clients learn and contribute to the global model.

RQ2. *How can performance disparities across clients be reduced while maintaining competitive global performance under non-IID conditions in federated learning?*

In non-IID settings, uniform or data-size averaging can favor clients with data distributions that align with the global objective, marginalizing those with skewed or challenging data. This can lead to improvements in average performance while increasing disparities among clients. While fairness-aware FL methods have been proposed, many create explicit trade-offs between average model performance and the variance in performance among clients, or focus solely on aggregation-level adjustments without addressing the training dynamics of under-performing clients.

This research question investigates design strategies that jointly optimize global performance and inter-client fairness. The central concern is whether fairness can be improved without sacrificing, and ideally while enhancing, overall model performance.

RQ3. *How can the underlying structure of client heterogeneity be leveraged to improve learning dynamics and fairness in federated learning?*

Most FL frameworks treat clients as independent and exchangeable participants, despite the possibility that meaningful similarities exist among subsets of clients. In practice, clients may share comparable label distributions or data characteristics, forming latent groups with related learning objectives. Ignoring such structure can lead to inefficient aggregation and increased performance disparities among clients, particularly under highly non-IID conditions. While fairness-aware aggregation strategies adjust client contributions at a global level, they often overlook intermediate structural relationships among clients.

This research question explores how distributionally similar client groups can be identified using privacy-preserving summaries, and incorporating this structure into multi-level aggregation, to reduce inter-client disparity while maintaining strong global performance. The emphasis is on leveraging heterogeneity rather than merely compensating for it.

1.4 Thesis Contributions

This thesis advances FL under heterogeneous and non-IID conditions through three interconnected methodological contributions. The contributions progress from diagnostic analysis of client behavior to adaptive mitigation strategies and finally to structure-aware federated optimization.

C1. A Framework for Monitoring Client Behavioral Deviations from Model Updates

This thesis introduces a framework for characterizing client behavioral deviations during federated training using signals derived from model updates. Unlike prior approaches that rely on performance metrics, the proposed method captures geometric properties of client models to detect atypical learning dynamics without access to raw data.

Empirical analysis demonstrates that the derived deviation measures correlate with both; client influence on the global model, and client benefit obtained from federated collaboration. This establishes a measurable and interpretable link between client learning dynamics and federation-level outcomes.

This contribution is presented in Paper I.

C2. A Mechanism for Reducing Performance Disparities through Adaptive Federated Optimization

This thesis proposes an adaptive federated optimization approach that reduces inter-client performance disparities under non-IID data. The method dynamically adjusts client contributions during training and improves learning for underperforming clients, resulting in more balanced performance while maintaining competitive global accuracy.

This contribution is presented in Paper II.

C3. A Framework for Reducing Performance Disparities through Distribution-Aware Federated Learning

This thesis introduces a framework that leverages latent similarities among clients to guide federated training under heterogeneous data conditions. By incorporating distributional relationships into the learning process, the approach reduces interference across clients and improves the consistency of client-level performance. Empirical results show that this leads to reduced performance disparities across clients while maintaining competitive global model accuracy.

This contribution is presented in Paper III.

Collectively, these contributions establish a coherent approach to FL under heterogeneity, progressing from behavioral characterization, to adaptive mitigation, and finally to structural exploitation of client similarities in order to enhance both global performance and equity across clients.

1.5 Thesis Outline

The thesis is organized as follows. **Chapter 2** provides the necessary background on FL, and the key methodological concepts used throughout this work. **Chapter 3** presents a structured review of the related work around the research questions. **Chapter 4** describes the research methodology and evaluation framework. It formalizes the FL setup, defines the system assumptions, and presents the experimental protocol used throughout the thesis. This chapter also introduces the datasets, baselines, and evaluation metrics, providing a unified framework for analyzing client behavior, fairness, and performance across the proposed methods. **Chapter 5** presents the findings and discussion, structured around the research questions. It first analyzes client behavior, then examines methods for reducing inter-client performance disparities, and finally evaluates distribution-aware approaches that leverage relationships among clients. This chapter also synthesizes results across the proposed methods to provide a unified analysis of how client heterogeneity influences learning dynamics. Finally, **Chapter 6** concludes the thesis by summarizing the main contributions and outlining directions for future work.

2 Background

2.1 Federated Learning

Machine learning (ML) studies the development of algorithms that learn patterns from data to create prediction models that facilitate human decision-making. It can be broadly categorized into supervised learning, which uses labeled data to predict known outcomes, and unsupervised learning, which discovers hidden patterns in unlabeled data. Traditional ML approaches rely on centralized data collection, where data from multiple sources are aggregated and processed in a single location. While effective, this paradigm introduces significant challenges related to data privacy, communication overhead, and regulatory constraints. Moreover, centralized training requires significant data and computational resources at a single entity, creating scalability issues as data volumes grow. These motivate the need for distributed learning, which decentralizes both data and computation across multiple participants.

Federated Learning (FL) is a decentralized framework, introduced by Google [4], where multiple clients collaboratively train a shared global model with a central server's coordination. In FL, each client performs local optimization on its private data and shares only model updates, such as gradients or parameters, thereby preserving data locality and privacy. This approach is particularly suitable for settings where data is sensitive, geographically distributed, or subject to access restrictions.

Formal FL Objective. Formally, consider a federation of K clients, where each client k holds a local dataset \mathcal{D}_k drawn from a client-specific data distribution. The goal of FL is to learn a global model \mathcal{M} by minimizing the average empirical loss across all clients:

$$\min_{\mathcal{M}} \sum_{k=1}^K \frac{n_k}{n} \ell_k(\mathcal{M}),$$

where $n_k = |\mathcal{D}_k|$ is the size of the local dataset at client k , $n = \sum_{k=1}^K n_k$ is the total number of samples across all clients, and $\ell_k(\mathcal{M})$ denotes the local loss at client k . This objective implicitly assumes that client data are Independent and Identically Distributed (IID), ensuring that aggregated updates approximate a coherent global objective.

Statistical Heterogeneity Definition. Statistical heterogeneity in FL refers to differences in the data distributions across participating clients. Formally, let P_k denote the data distribution of client k . A federated setting is statistically heterogeneous when $P_i \neq P_j$ for some $i \neq j$, indicating that client data are not non-IID. These differences may arise from label imbalance, feature distribution shifts, or variations in data collection processes. As a result, local model updates become inconsistent across clients, which affects the stability of aggregation and degrades the generalization of the global model.

Federated Optimization under Statistical Heterogeneity. The presence of statistical heterogeneity fundamentally challenges federated optimization. In practical FL systems, the data distributions across clients are rarely identical, leading each client to optimize its model with respect to its own local objective. Consequently, local updates become misaligned, resulting in conflicting gradients during aggregation and reducing the effectiveness of global training.

Client Drift. Under statistical heterogeneity, local optimization steps lead to client models diverging from the global model, a phenomenon called client drift. This results in misalignment of aggregated updates, as local models optimize based on their specific data distributions.

Objective Inconsistency. Client drift causes objective inconsistency, as local objectives optimized by individual clients can differ from the global objective. While the global model aims to minimize the average loss across all clients, each client focuses on minimizing its own local loss. In statistically heterogeneous settings, this misalignment can result in aggregated updates that fail to achieve coherent global optimization.

Effect on Convergence. The combined effects of statistical heterogeneity, client drift, and objective inconsistency degrade the convergence behavior of federated optimization. Conflicting local updates increase the variance of aggregated gradients, leading to slower convergence, instability, and, in some cases, divergence. Consequently, standard federated optimization methods may fail to achieve optimal performance in highly heterogeneous environments.

2.2 Eccentricity Analysis

The Typicality and Eccentricity-based Data Analytics (TEDA) framework, introduced by Angelov [5], provides a data-driven approach for analyzing structural relationships in datasets without relying on assumptions such as independence. It characterizes

each data sample relative to all others, making it suitable for finite and potentially dependent data. Within TEDA, the eccentricity ξ_k^n of a data sample x_k (for $n > 2$) is defined as:

$$\xi_k^n = \frac{2 \sum_{i=1}^n d_{ki}}{\sum_{i=1}^n \sum_{j=1}^n d_{ij}} \quad (2.1)$$

where d_{ij} denotes the distance between samples x_i and x_j . The typicality is defined as its complement:

$$\tau_k^n = 1 - \xi_k^n \quad (2.2)$$

In this thesis, the TEDA is extended beyond data samples to analyze structural properties of client models and their representations in FL. In particular, Paper I adapts eccentricity measures to quantify client behavior during training, enabling the identification of clients with deviating learning dynamics (see Sec. 5.1).

2.3 Hessian Matrix

A neural network (NN) consists of layers with parameters that are adjusted during training to minimize a loss function. This loss defines the objective of training and depends on the model parameters.

To capture higher-order properties of the loss landscape, second-order information can be considered through the *Hessian matrix*. Formally, the Hessian of a scalar function $f : \mathbb{R}^X \rightarrow \mathbb{R}$ is the matrix of second-order partial derivatives. This describes the local curvature of f at a given point [6]. In NN, the *Hessian* reveals how changes in model parameters collectively impact the loss, providing a richer characterization than gradients alone.

In this thesis, *Hessian* based representations are used to capture geometry of client models in the parameter space. In particular, Paper I leverages *Hessian* derived information to compute eccentricity measures, enabling the characterization of client behavior during training.

2.4 Multiclass Ada Boosting

Adaptive Boosting (AdaBoost) is an ensemble learning method that constructs a strong classifier by combining weak learners (classifiers) sequentially. It reweights training samples at each iteration to focus on previously misclassified instances, with the final prediction being a weighted combination of all learners.

SAMME (Stagewise Additive Modeling using a Multi-class Exponential loss function) [7] extends AdaBoost to multi-class classification. It preserves the iterative

reweighting principle of AdaBoost while adjusting the learner weights to account for the number of classes.

The weight assigned to the l -th classifier is defined as:

$$\alpha_l = \ln \left(\frac{1 - \mathcal{E}_l}{\mathcal{E}_l} \right) + \ln(C - 1), \quad (2.3)$$

where C denotes the number of classes and \mathcal{E}_l is the weighted classification error.

In this thesis, boosting-inspired weighting mechanisms are used to adaptively regulate client contributions in FL. In particular, Paper II builds on the SAMME formulation to design a performance-aware aggregation strategy that emphasizes reliable client updates.

2.5 Focal Loss

Focal loss [8] addresses class imbalance by modifying the standard cross-entropy loss to focus on hard-to-classify examples. It introduces a scaling factor that reduces the contribution of well-classified samples while emphasizing difficult ones. The focal loss is defined as:

$$\mathcal{L}_{\text{Focal}}(p_t) = -\beta_t(1 - p_t)^\gamma \log(p_t), \quad (2.4)$$

where p_t is the predicted probability of the ground-truth class, β_t is a class-balancing factor, and $\gamma \geq 0$ is the focusing parameter.

In this thesis, focal loss is used to emphasize hard-to-learn patterns in client data during training. In particular, Paper II integrates focal loss into the local training objective to prioritize difficult samples under heterogeneous data distributions.

2.6 Knowledge Distillation

Knowledge distillation (KD) [9] is a technique for transferring knowledge from a reference model (teacher) to another model (student) by aligning their output distributions. The student learns not just from ground-truth labels but also from the teacher’s softened predictions, which provide extra structural information about class relationships.

The distillation objective is typically defined using a combination of standard cross-entropy loss and a divergence term between teacher and student outputs:

$$\mathcal{L} = (1 - \lambda)\mathcal{L}_{\text{CE}} + \lambda\tau^2\mathcal{L}_{\text{KD}}, \quad (2.5)$$

where \mathcal{L}_{KD} is commonly implemented as the Kullback–Leibler divergence between the softened output distributions of the teacher and student, $\lambda \in [0, 1]$ controls the

influence of distillation, and $\tau > 0$ is a temperature parameter that smooths the predicted probabilities.

In this thesis, KD is used to transfer information between related client models in FL. In particular, Paper III employs distillation to enable structured knowledge sharing across groups of clients, improving learning under heterogeneous data distributions

2.7 Clustering

Clustering is an unsupervised ML technique that partitions data into groups such that samples within the same group are more similar to each other than to those in other groups. This grouping is defined with respect to a chosen similarity or distance metric, which determines the structure of the resulting clusters.

Clustering methods can be broadly categorized into distance-based approaches, such as hierarchical clustering, and probabilistic approaches, such as Gaussian Mixture Models (GMM) [10], which represent data as a mixture of underlying distributions.

In this thesis, clustering is used to identify groups of similar clients in federation. In particular, Paper III treats clustering as a modular component, allowing the framework to remain agnostic to the specific clustering algorithm. The choice of clustering method is driven by data characteristics and is instantiated differently in the experimental evaluation.

3 Related Work

This section reviews prior work in FL along three dimensions that align with the research questions of this thesis: (i) characterization of client behavior (**RQ1**), (ii) fairness-aware learning mechanisms (**RQ2**), and (iii) distribution-aware learning under statistical heterogeneity (**RQ3**). Existing approaches largely address these aspects in isolation. This section highlights their contributions and limitations, motivating the need for a unified perspective.

Client Behavior Characterization and Monitoring (RQ1). Understanding the behavior of individual clients in FL is challenging due to its decentralized and partially observable nature: the server observes only model updates, without access to local data or optimization dynamics.

A significant number of prior studies address this problem indirectly through *contribution evaluation*. These methods aim to quantify the importance of individual clients, often using cooperative game-theoretic formulations such as Shapley value-based methods (e.g., leave-one-out (LOO) [11] and Truncated Monte Carlo Shapley Value (TMC-SV) [12]). These approaches estimate clients' importance by measuring performance changes when clients are excluded. While theoretically grounded, they are computationally expensive and typically applied as post hoc analyses. Approximation-based methods, including FedCCEA [13], CAFL [14], FedCM [15], and more recent methods such as VPM and IEM [16], reduce the computational cost but remain focused on contribution.

However, contribution evaluation primarily explains *how* each client contributes, rather than explaining *how* clients behave during training. As a result, these methods provide limited insight into the underlying learning dynamics and fail to capture temporal or structural variations in client behavior.

A separate line of work focuses on *behavior monitoring*, detecting anomalous clients using signals derived from model updates, such as similarity metrics or visualization systems like VADAF [17] and HFLens [18]. While useful for diagnosis, these approaches do not provide systematic or quantitative characterizations of client behavior.

Overall, existing methods either quantify contribution without understanding behavior or monitor behavior without integrating it into a systematic analytical framework.

Consequently, there is a lack of principled approaches for characterizing client behavior in a way that reveals underlying training dynamics. Such behavioral understanding is essential for exposing how client heterogeneity affects the learning process, including disparities in performance and influence across clients. While recent efforts [17] have begun to explore behavior-driven signals derived from model dynamics, these approaches remain limited in their ability to systematically capture client behavior throughout the training process.

This gap limits the ability to support deeper analysis of performance disparities among clients and learning dynamics in federated settings, motivating the need for methods that provide continuous, structured, and learning-relevant representations of client behavior.

Fairness in Federated Learning via Improving Under Performing Clients (RQ2).

In practice, the heterogeneity observed in client behavior manifests as disparities in model performance across clients, where some consistently benefit from the federation while others experience degraded outcomes. This challenge has motivated a growing body of work on fairness in FL, with the goal of reducing performance disparities while maintaining strong global accuracy.

A class of fairness-aware approaches modifies the learning objective to prioritize underperforming clients. For example, AFL [2] formulates training as a worst-case risk minimization problem, while Q-FFL [19] reweights client updates based on their local losses to emphasize disadvantaged clients. Extensions such as FCFL [20] and F3 [21] introduce multi-objective and adaptive regularization strategies to balance fairness and accuracy. Reinforcement learning-based approaches, such as FAIRFL [22], aim to jointly optimize these objectives under system constraints.

A complementary line of work addresses fairness through the design of personalization and aggregation. Methods such as DITTO [3] maintain client-specific models to account for heterogeneous data distributions, while approaches such as FAIRFED [23] and FEDFAIM [24] adjust aggregation weights or introduce incentive mechanisms to improve fairness. More recent methods, such as FEDMH [25], leverage historical training signals, and FADE [26] explicitly mitigates bias during aggregation.

In parallel, several methods focus on mitigating statistical heterogeneity through optimization stabilization rather than fairness-aware objectives. Techniques such as FEDPROX [1], FEDNOVA [27], SCAFFOLD [28], and MOON [29] improve convergence stability and alignment between local and global models under non-IID data. While these approaches enhance overall performance, they do not explicitly address disparities in client-level outcomes.

Despite these advances, fairness is primarily addressed through reactive adjustments at the optimization, personalization or aggregation level, without explicitly account-

ing for the underlying causes of performance disparity. This highlights the need for approaches that can reduce disparities across clients under statistical heterogeneous conditions while maintaining, or ideally improving, overall model performance by encouraging more effective participation from under performing clients during training.

Distribution-Aware Federated Learning under Heterogeneity (RQ3). The limitations of existing fairness-aware approaches suggest that addressing performance disparities through optimization-level adjustments alone is insufficient. In particular, these methods treat clients as independent participants, without explicitly modeling the relationships that arise from similarities in their data distributions or learning dynamics.

In practice, clients' statistical heterogeneity is not arbitrary but often exhibits latent structure, where subsets of clients share common characteristics such as label distributions, feature patterns, or training behavior. Ignoring this structure can lead to inefficient aggregation and persistent disparities, as the learning process fails to account for these underlying relationships. This has motivated a growing line of work that seeks to identify and exploit structure among clients to improve both learning dynamics and fairness in federated settings.

A prominent approach in this direction is Clustered Federated Learning (CFL) [30], which explicitly groups clients based on similarities in their data distributions or model updates. By partitioning the clients in the federation into more homogeneous subgroups, CFL aims to reduce the negative effects of statistical heterogeneity and improve both convergence and client-level performance. Recent works extend this idea by incorporating knowledge transfer across clusters. For instance, CFLKD [31] groups clients according to the similarity of their local model parameters and applies cross-group knowledge distillation to enable information sharing between cluster-specific models without direct parameter aggregation. Similarly, DisUE [32] introduces a distillation-based CFL framework that constructs a universal expert model by aggregating knowledge from multiple clusters, improving generalization across diverse client groups.

While these approaches leverage structure in client heterogeneity to improve generalization and training efficiency, their impact on reducing performance disparities remains limited, as fairness is not explicitly incorporated. This highlights the need for distribution-aware strategies that utilize relationships among clients to improve learning dynamics and reduce disparities in federated settings.

Summary and Research Gap. The reviewed literature highlights three areas of research that have largely been studied independently: client behavior analysis, fairness-aware optimization, and distribution-aware learning. Existing methods typically fall

into one of three categories (i) analyze client behavior without integrating it into the learning process, (ii) enforce fairness through optimization or aggregation mechanisms without understanding client dynamics, or (iii) exploit structure for scalability without explicitly addressing fairness. This thesis addresses these gaps through three complementary studies. Together, they characterize and monitor client heterogeneity, incorporate it into the optimization process, and leverage its structure to enhance fairness and robustness in FL.

4 Methodology

4.1 Research Methodology

This thesis adopts an **experimental** research methodology, where controlled experiments are used to evaluate the effects of specific variables, to investigate client behavior, fairness, and performance dynamics in FL. The study is **empirical** and relies on controlled, simulation-based experiments to evaluate the proposed methods under varying conditions of statistical heterogeneity across clients. This approach enables systematic comparison across methods by isolating specific factors such as data distribution, client participation, and training dynamics. Given the decentralized, partially observable nature of FL, analytical evaluation alone is insufficient to capture the complex interactions between clients and the global model. Therefore, experimental analysis on benchmark datasets and controlled settings is used to provide evidence for the effectiveness and limitations of the proposed approaches.

4.2 Evaluation Framework

This section outlines the evaluation framework used throughout the thesis. It describes the FL setup, the system assumptions, the datasets used in the evaluation, and the evaluation metrics used to assess model performance and fairness across clients.

4.2.1 Federated Learning Setup

The research presented in this thesis considers a *horizontal* FL setting. In this setting, multiple clients collaboratively train a shared global model (\mathcal{M}) under the coordination of a central server while keeping their local data private.

In this setup, we consider a federation consisting of K clients. Each client k holds a private local dataset defined as $\mathcal{D}_k = \{(x_i^k, y_i^k)\}_{i=1}^{n_k}$, where x_i^k denotes an input sample and y_i^k the corresponding target label. The data is drawn from a distribution that is specific to each client. These distributions may differ across clients, resulting in statistical heterogeneity (i.e., non-IID data) across the federation. Training proceeds through iterative communication rounds. At each round (e), the server distributes the current global model ($\mathcal{M}^{(e)}$) to a subset of participating clients. Each selected client

k performs local training on its private dataset, \mathcal{D}_k and returns the updated model parameters $\mu_k^{(e)}$ to the server. The server aggregates the received updates to produce a new global model ($\mathcal{M}^{(e+1)}$) for the next communication round.

While the basic communication protocol remains unchanged, some proposed methods extend the standard FL procedure by allowing clients to transmit additional training statistics alongside model updates without violating the privacy constraints. These statistical signals guide the server’s aggregation and training strategies.

4.2.2 Assumptions and Constraints

The FL environment considered in this thesis is defined under the following assumptions.

Data privacy constraint. Clients do not share their raw data with the server or with other participants. Communication is limited to model parameters and, where required by the specific method, auxiliary statistics derived from local training that do not expose raw client data.

Partial observability. The server has access only to information communicated by participating clients, such as model updates and auxiliary training statistics. The internal training processes and local data of individual clients remain inaccessible.

Client heterogeneity. Clients may differ in terms of their underlying data distributions, dataset sizes, and optimization dynamics. Such heterogeneity leads to divergent local updates and uneven learning outcomes across the federation.

4.2.3 Datasets

The experimental evaluation in this thesis uses several publicly available datasets. MNIST, FEMNIST and CIFAR-10 are used in Papers II and III, and the Key–Value store monitoring dataset is used in Paper I for client behavior analysis.

MNIST. The MNIST dataset [33] consists of grayscale images of handwritten digits with a resolution of 28×28 pixels and 10 classes. The original dataset is centralized. To simulate statistically heterogeneous FL environments, the training data is partitioned across clients using *Dirichlet* [34] sampling, which produces uneven label distributions and varying dataset sizes across clients.

CIFAR-10. CIFAR-10 [35] contains 32×32 RGB images representing 10 object categories. Similar to MNIST, the dataset is originally centralized. In the FL experiments, the training data is partitioned across clients using *Dirichlet* sampling to introduce statistical heterogeneity across participants.

FEMNIST. FEMNIST [36] is an extension of the MNIST dataset containing hand-

written characters from 62 classes. Unlike the previous two datasets, FEMNIST is naturally partitioned by writer, where each client corresponds to a unique user. This structure results in inherently non-IID data distributions across clients and therefore provides a realistic benchmark for FL experiments.

Key–Value store monitoring dataset. For the empirical analysis of client behavior, a publicly available system monitoring dataset collected from a distributed Key–Value store database cluster is used [37]. The dataset contains runtime measurements from both server and client nodes, including CPU utilization per core, memory utilization, network utilization, and disk I/O statistics. In Paper I, the target variable is the average read latency of the Key–Value store system. Predicting this latency enables operators to identify abnormal client behavior and take corrective actions to maintain service-level agreements.

4.2.4 Baselines

The proposed methods in this thesis are evaluated against a few standard and fairness-aware FL baselines. These baselines are selected to represent different approaches to handling statistical heterogeneity and performance disparity across clients.

FEDAVG [4] serves as the standard baseline for federated optimization. It aggregates local model updates using weights proportional to the size of each client’s dataset. This method provides a reference for evaluating improvements in both global model performance and training stability under non-IID data.

Q-FEDAVG [19] is a fairness-aware extension of FEDAVG that reweights client updates based on their current loss values. Clients with higher loss receive greater emphasis during aggregation, aiming to reduce performance disparities across clients. This method serves as a baseline for loss-based fairness optimization.

DITTO [38] introduces a personalized FL approach by maintaining both a global model and client-specific local models. A regularization term controls the deviation between the global and personalized models, allowing clients to adapt to their local data distributions while retaining shared knowledge. This baseline provides a reference for evaluating personalization-based approaches to heterogeneity.

4.2.5 Experimental Protocol

All experiments were conducted in a simulated FL environment implemented in Python, with model training performed using the *PyTorch* framework. All baseline methods are implemented within the same framework and evaluated under a consistent experimental setup to ensure fair comparison. The experimental framework follows the operational assumptions described below.

Centralized server architecture. A simulated central server coordinates the client devices and performs aggregation of client model updates. Since the experiments are conducted in simulation, all clients are assumed to have identical computational capacity in order to isolate the effects of statistical heterogeneity from system heterogeneity.

Synchronous training configuration. All participating clients perform the same number of local training epochs using identical training configurations, including batch size, learning rate, optimizer, and other optimization hyperparameters. The server waits for all selected clients to complete their local training before performing model aggregation in each communication round.

Static local datasets. Each client maintains a fixed local dataset that remains unchanged throughout the training process. When a client is selected to participate in a communication round, it trains on its full local training dataset for the prescribed number of local epochs.

Non-IID data generation. For datasets that are originally homogeneous, statistical heterogeneity is simulated using *Dirichlet* partitioning. For datasets that are naturally distributed across clients, the original client data partitions are preserved.

Client participation. Depending on the experimental setting, either full client participation or partial participation is used. In partial participation scenarios, a fixed number of clients are randomly selected in each communication round.

Implementation Details. Unless otherwise stated, detailed hyperparameter settings for each experiment (e.g., number of clients, learning rate, local epochs, batch size, participation ratio, and *Dirichlet* concentration parameter) are provided in the corresponding experimental sections in papers.

4.2.6 Evaluation Metrics

We evaluate trained FL models at the client level along three complementary dimensions: (i) overall predictive performance, (ii) inter-client performance fairness, and (iii) client-level influence and benefit. The first dimension is used across all studies in this thesis, while fairness and client-level behavior analysis are examined in specific papers. In particular, we quantify the influence of individual clients and assess its relationship with the benefit they obtain from the federation by computing *Kendall's rank correlation coefficient* [39] between the corresponding client rankings.

The global model, denoted as \mathcal{M} , is evaluated consistently across all studies in this thesis using each client's local test dataset derived from \mathcal{D}_k (Sec. 4.2.1). Let $\ell_k(\mathcal{M})$ denote the empirical loss, and let $\varphi_k(\mathcal{M})$ denote a task-specific performance metric (such as classification accuracy or mean absolute error) of the global model on client k .

The *overall performance* of the model is evaluated using the average client performance:

$$\bar{\varphi}(\mathcal{M}) = \frac{1}{K} \sum_{k=1}^K \varphi_k(\mathcal{M}).$$

Similarly, the *average global model loss* is defined as:

$$\bar{\ell}(\mathcal{M}) = \frac{1}{K} \sum_{k=1}^K \ell_k(\mathcal{M}).$$

In Papers II and III, fairness across clients is quantified using the *variance of client performance*:

$$\text{Var}(\varphi(\mathcal{M})) = \frac{1}{K} \sum_{k=1}^K (\varphi_k(\mathcal{M}) - \bar{\varphi}(\mathcal{M}))^2. \quad (4.1)$$

A lower variance indicates more uniform performance across clients.

In addition, Papers II and III report *Jain's fairness index* [40, 41], defined as:

$$J(\varphi(\mathcal{M})) = \frac{\left(\sum_{k=1}^K \varphi_k(\mathcal{M})\right)^2}{K \sum_{k=1}^K \varphi_k(\mathcal{M})^2}, \quad (4.2)$$

where $J(\varphi(\mathcal{M})) \in \left[\frac{1}{K}, 1\right]$, and values closer to 1 indicate higher fairness.

In Paper I, a client deletion-based approach is adopted to quantify client influence [11]. For client k , we train a reference model \mathcal{M}' without client k , defining the influence of client k as:

$$\text{Influence}^k = |\bar{\varphi}(\mathcal{M}) - \bar{\varphi}(\mathcal{M}')|, \quad (4.3)$$

This formulation captures a client's influence on overall model performance, providing a task-agnostic measure applicable across both classification and regression settings. A higher influence value indicates that removing client k significantly affects average model performance, suggesting client k has a stronger impact on the global model.

In addition to influence, we define a client-level benefit measure to quantify the performance gain for client k from participating in FL compared to training in isolation.

$$\text{Benefit}^k = \varphi_k(\mathcal{M}) - \varphi_k(\mathcal{M}_k^{\text{local}}), \quad (4.4)$$

Here, $\mathcal{M}_k^{\text{local}}$ represents a model trained solely on client k 's local dataset \mathcal{D}_k , without federation. A higher benefit indicates greater advantages from participating FL, while a lower or negative value suggests minimal or no benefit from the federation.

To assess the relationship between client influence and benefit, Paper I computes *Kendall's rank correlation coefficient*, denoted as \mathcal{T} , between the corresponding client rankings. A higher value of \mathcal{T} indicates stronger agreement between the rankings, while lower or negative values indicate weak or inverse relationships.

4.3 Threats to Validity

This section addresses potential limitations that could impact the validity of the experimental findings and interpretations in this thesis, categorized into internal, construct, external, and statistical conclusion validity.

4.3.1 Internal Validity

In experimental research, internal validity concerns whether the observed effects can be reliably attributed to the proposed methods rather than to confounding factors or uncontrolled variables.

The experiments are conducted in a simulated FL environment with controlled conditions, including synchronous training, identical client computational capabilities, and fixed local datasets. While this design isolates the effects of statistical heterogeneity, it ensures that observed differences can be attributed to the proposed methods rather than system-level variability. However, this controlled setup removes factors such as stragglers, communication delays, and hardware differences, which may influence training dynamics in practical deployments.

Additionally, the proposed methods introduce multiple interacting components, such as performance-aware aggregation and boosting in FEDABOOST, and hierarchical clustering with distillation in DEFFT. The observed improvements likely stem from these combined components rather than individual mechanisms. Although later sections provide insights from ablation studies, fully disentangling these components is limited.

Finally, the results are sensitive to hyperparameter choices (e.g., boosting rate, distillation weight, clustering), which are chosen empirically and may affect the trade-offs between performance and fairness.

4.3.2 Construct Validity

In experimental evaluation, construct validity concerns whether the chosen metrics and measurements accurately capture the abstract concepts they are intended to represent. The evaluation of the proposed methods depends on several proxy metrics to quantify abstract concepts such as influence, benefit, client behavior, and fairness.

Client influence is quantified using a deletion-based approach (4.3). This assumes that the marginal impact of a client can be isolated independently, which may not fully capture complex interactions between client updates in federated optimization. Similarly, the benefit metric (4.4) assumes that improvements over local training reflect meaningful gains from federation, which may not fully account for differences in model capacity or optimization dynamics. The proposed eccentricity-based measures are interpreted as indicators of client behavior and deviation. However, these scores are derived from distances between model representations and do not directly correspond to semantic notions such as data quality or usefulness. Their interpretation as behavioral signals, therefore, depends on this modeling assumption.

Finally, fairness is primarily evaluated using the variance of client performance (II.2) and *Jain's fairness index* (III.3), which quantify dispersion across clients. While these metrics capture overall distributional equity, they do not fully represent all fairness notions, such as strict worst-case guarantees or group-level fairness constraints. To partially address these limitations, we report tail-focused metrics, including minimum client performance, 10th-percentile accuracy, and worst-10% mean accuracy, to highlight disparities among under-performing clients. We also analyze client performance distributions using histograms and Kernel Density Estimation (KDE) plots, providing insights into spread, skewness, and tail behavior. This analysis helps identify improvements in lower-performing clients, although it remains empirical and does not offer formal fairness guarantees.

4.3.3 External Validity

In empirical studies, external validity concerns the extent to which experimental findings generalize beyond the specific datasets, settings, and assumptions used in the evaluation.

The evaluation is conducted on standard benchmark datasets (MNIST, CIFAR-10, FEMNIST) and a Key-Value store monitoring dataset (see Sec. 4.2.3). These datasets include synthetic and natural settings but may not fully capture complex, evolving, or multi-modal data distributions in real-world FL scenarios. For centrally homogeneous datasets (MNIST and CIFAR10), non-IID conditions are simulated using *Dirichlet* partitioning. Although this method is widely used, it may not accurately reflect the real-world heterogeneity, which can involve temporal drift, feature shifts, or label inconsistencies.

The experimental setup primarily reflects cross-silo or controlled FL settings, with synchronous training, stable client availability, and homogeneous computational assumptions. While the evaluation on FEMNIST includes a large number of clients and incorporates partial participation through random client sampling in each round, this mechanism represents a simplified participation model. In particular, client

selection is assumed to be uniformly random and independent of system constraints. In real-world cross-device FL, participation is often influenced by factors such as device availability, network conditions, and energy constraints, leading to non-random and potentially biased participation patterns. As a result, the findings may not fully generalize to highly dynamic deployments where participation is system-driven rather than randomly sampled.

In addition, the clustering assumptions in DEFFT, based on label distributions or derived embeddings, rely on the availability of reliable client-level statistics. In practical deployments, such information may be incomplete, noisy, or restricted due to privacy constraints, which could reduce the effectiveness of the clustering process.

4.3.4 Statistical Conclusion Validity

In empirical analysis, statistical conclusion validity concerns whether the data, experimental design, and statistical methods support reliable and sound inferences about observed effects.

The reported results are averaged over multiple runs; however, FL exhibits inherent variability due to random initialization, client sampling, and data partitioning. Limited repetitions may not fully capture this variability, especially for fairness-related metrics that are sensitive to tail behavior.

The use of *Kendall's correlation* to relate influence and benefit introduces additional sensitivity, as small changes in client performance can alter rankings. Consequently, conclusions regarding relationships between client properties should be interpreted as indicative rather than definitive.

Although convergence plots are reported to assess training dynamics, the final comparisons remain tied to fixed training budgets and selected evaluation points. As a result, part of the observed performance differences may reflect differences in convergence speed or stability rather than only in the quality of the converged solution.

5 Findings and Discussions

Building on the challenges of heterogeneity and fairness outlined in Sec. 1.1, this chapter presents a structured investigation of how client-level differences can be systematically understood and addressed in FL. Rather than treating heterogeneity as a purely disruptive factor, the following studies examine how it: (i) manifests in training dynamics, (ii) influences client-level outcomes, and (iii) can be incorporated into the design of learning algorithms.

To this end, the chapter provides an overview of the results and main findings of the three complementary studies included in this thesis, and discusses the answers to the RQs. The first study (**Paper I**) focuses on monitoring and characterizing client behavior through model-based signals derived from training dynamics. The second study (**Paper II**) introduces adaptive optimization mechanisms to mitigate performance disparities across clients. The third study (**Paper III**) extends this perspective by explicitly modeling similarities between clients, enabling structure-aware aggregation and knowledge transfer.

5.1 Study 1: Client Behavior Monitoring

Client behavior is characterized using three complementary eccentricity measures that capture structural deviations in local training dynamics and global model interaction.

Local Model Eccentricity. The local eccentricity score ξ_k^L quantifies how distinct the local optimization landscape of client k is relative to other clients. It is defined using pairwise distances between *Hessian* matrices of local models:

$$\xi_k^L = \frac{2 \sum_{j=1}^K d(\mathcal{H}_{\mu_k}, \mathcal{H}_{\mu_j})}{\sum_{l=1}^K \sum_{j=1}^K d(\mathcal{H}_{\mu_l}, \mathcal{H}_{\mu_j})}, \quad (5.1)$$

where \mathcal{H}_{μ_k} denotes the *Hessian* evaluated at the local model parameters μ_k , and $d(\cdot, \cdot)$ is the Euclidean distance.

Global Model Eccentricity. To capture how client k is represented in the global model, the global eccentricity score ξ_k^G is defined as:

$$\xi_k^G = \frac{2 \sum_{j=1}^K d(\mathcal{H}_{\mathcal{M}}^k, \mathcal{H}_{\mathcal{M}}^j)}{\sum_{l=1}^K \sum_{j=1}^K d(\mathcal{H}_{\mathcal{M}}^l, \mathcal{H}_{\mathcal{M}}^j)}, \quad (5.2)$$

where $\mathcal{H}_{\mathcal{M}}^k$ denotes the *Hessian* of the global model evaluated using client k 's data. This measure reflects how the global model behaves with respect to each client's data distribution.

Parameter-Based Approximation. To reduce computational cost, a parameter-based approximation ξ_k^P is computed directly from model weights and biases. Eccentricity is evaluated per layer and aggregated, providing a lightweight alternative that avoids second-order computations.

These *eccentricity scores* (ξ_k^L , ξ_k^G , and ξ_k^P) can be computed at selected communication rounds or throughout the entire training process. This results in a sequence of values for each client, forming a behavioral signature that captures how client dynamics evolve over time. These signatures enable the analysis of client stability, drift, and consistency during training, providing insights into the role of individual clients in the federation.

5.1.1 Main Findings

Sensitivity to structural deviations. Controlled experiments, designed to isolate variations in data volume, data distribution, and local training epochs, show that all three eccentricity measures consistently identify deviating clients. This indicates that the measures capture structural differences in client behavior rather than stochastic variation.

Limitations of performance-based metrics. Client influence (4.3) and benefit (4.4) exhibit weak and negative rank correlation, as measured by *Kendall's tau* (τ), under both evaluated training regimes: (i) a setting with many global rounds and minimal local updates ($\mathcal{T} = -0.22$), and (ii) a setting with fewer global rounds and more intensive local training ($\mathcal{T} = -0.34$). This demonstrates that performance-based metrics alone do not adequately reflect client roles in the federation.

Global eccentricity as a robust indicator. Among the proposed measures, ξ_k^G provides the most consistent signal. It shows a stable negative correlation with client benefit ($\mathcal{T} \approx -0.6$) and a positive correlation with influence ($\mathcal{T} \approx 0.3-0.4$), across both early and late training stages. This indicates that structurally distinct clients are more likely to influence the global model.

In contrast, ξ^L and ξ^P exhibit weaker and less stable relationships, limiting their standalone interpretability.

Temporal behavioral signatures. Tracking *eccentricity scores* (ξ_k^L , ξ_k^G , and ξ_k^P) across communication rounds yields client-specific behavioral signatures that capture the evolution of client dynamics over time. These signatures reveal two distinct patterns: (i) persistent deviations, where clients remain consistently eccentric, and (ii) transient deviations, where clients switch between typical and eccentric behavior during training.

Notably, clients exhibiting high eccentricity often correspond to those with high influence scores, indicating a link between behavioral deviation and impact on the global model. Furthermore, the observed temporal variability across clients demonstrates that client behavior is inherently dynamic rather than static. This enables continuous monitoring of client dynamics throughout the training process.

Impact on model performance. When clients identified as highly deviating were removed and the model retrained, the global error decreased (MAE reduced from 0.47 to 0.45). This provides evidence that eccentricity-based signals can identify clients whose behavior has a measurable impact on global model quality.

5.1.2 Answer to RQ1

The findings indicate that client behavioral deviations in FL can be systematically characterized using model-based structural signals, specifically eccentricity measures (ξ_k^L , ξ_k^G , and ξ_k^P) derived from both local and global models. In particular:

- Eccentricity measures capture deviations arising from differences in data distribution, training dynamics, and model behavior.
- Global-model eccentricity (ξ^G) provides a consistent and interpretable indicator of structurally distinct and potentially influential clients.
- Tracking eccentricity over training rounds yields temporal behavioral signatures that enable continuous monitoring of client drift, instability, and evolving behavior.

Overall, client behavior in FL can be represented as a time-evolving structural property of model updates, enabling systematic monitoring beyond static performance-based evaluation.

5.2 Study 2: Fairness in FL via Improving Under Performing Clients

This section introduces the FEDABOOST, a federated optimization framework that combines performance-aware aggregation with adaptive boosting to reduce inter-client performance disparities under non-IID data. FEDABOOST design is inspired by multi-class ADABOOST (see Sec. 2.4), but adapts it to a parallel federated optimization setting where clients contribute local model updates rather than sequentially trained weak learners.

Performance-aware aggregation. Clients are treated as weak learners, and their contributions are weighted based on local performance. At each global round e , a client k is assigned a weight:

$$\alpha_k^e = \ln \left(\frac{1 - \mathcal{E}_k^e}{\mathcal{E}_k^e} \right) + \ln(C_k - 1), \quad (5.3)$$

where \mathcal{E}_k^e is the local error and C_k is the number of classes. Clients with lower error receive higher weights, increasing their influence on aggregation. The global model is updated as:

$$\mathcal{M}^{e+1} = \frac{\sum_{k \in S_e} \alpha_k^e \mu_k^e}{\sum_{k \in S_e} \alpha_k^e}. \quad (5.4)$$

This aggregation scheme emphasizes higher-performing client updates while reducing the impact of noisy or low-quality contributions.

Boosting of under-performing clients. To complement aggregation, FEDABOOST increases the training emphasis on under-performing clients. Each client maintains a weight w_k^e , updated based on its performance:

$$w_k^e = w_k^{e-1} \exp(-\eta \alpha_k^e I_k^e). \quad (5.5)$$

where $\eta \in [0, 1]$ is a boosting rate that controls the magnitude of updates, I_k^e indicates whether the client meets a predefined performance threshold.

The updated client weights are incorporated into the local optimization process by adjusting the focusing parameter of the focal loss (see Sec. 2.5).

Training dynamics. These two mechanisms operate jointly: aggregation prioritizes reliable updates, while boosting improves weaker clients. This creates a feedback loop between client performance, aggregation weights, and local training dynamics, enabling balanced learning across heterogeneous clients.

5.2.1 Main Findings

Global performance is preserved or improved. Across all evaluated datasets (MNIST, CIFAR10, and FEMNIST), FEDABOOST achieves equal or higher global performance compared to FEDAVG and DITTO. Performance gains are most pronounced under low client participation(20%), indicating improved robustness and communication efficiency.

Inter-client performance variation is reduced. FEDABOOST produced more concentrated client performance distributions, indicating more uniform improvements across clients. Quantitatively, FEDABOOST reduces the variance of client performance: (i) on MNIST, variance is reduced by 24.4% compared to FEDAVG (ii) on FEMNIST, variance is reduced by 5.88% compared to FEDAVG and 11.87% compared to DITTO. These reductions, supported by non-overlapping confidence intervals, confirm that the method improves fairness across clients.

Effect of adaptive aggregation. The ablation study shows that alpha-based aggregation (5.4) alone improves performance, but the full FEDABOOST algorithm achieves further gains. This indicates that weighting clients based on performance contributes to robustness under non-IID data, but is insufficient on its own to reduce inter-client disparities.

Effect of targeted boosting. The boosting mechanism selectively increases the training emphasis on under-performing clients, leading to improved performance among weaker participants. This is reflected in the improved lower tail of the performance distribution and reduced variance across clients. This effect is particularly strong under partial participation, where weaker or underrepresented clients benefit most.

Robustness to heterogeneity and participation. The benefits of FEDABOOST are most significant in highly non-IID settings and under low participation rates. This suggests that adaptive weighting and boosting are particularly effective when client updates are sparse and heterogeneous, where standard averaging methods struggle to balance contributions.

5.2.2 Answer to RQ2.

The results demonstrate that reducing inter-client performance variation while maintaining competitive global performance can be achieved through a combination of:

- **Adaptive aggregation:** weighting client updates based on their performance to prioritize reliable contributions to the global model
- **Boosted training:** dynamically increasing the training focus on under-performing clients to improve their local models

The combination of these mechanisms creates a complementary effect: high-performing clients guide global optimization, while low-performing clients are actively improved through targeted updates. This leads to both improved global performance and reduced disparity across clients.

Therefore, adaptive client weighting alone is insufficient; it must be coupled with boosted training mechanisms to balance influence and learning across heterogeneous clients. The proposed FEDABOOST framework demonstrates that this joint design enables more fair and robust FL under non-IID conditions.

5.3 Study 3: Leveraging Distributional Similarities to Improve Fairness

In this section we introduce DEFPT, a hierarchical FL framework that explicitly accounts for non-IID data by organizing clients into distributionally similar groups and enabling structured knowledge transfer within these groups.

Label distribution based clustering: Prior to training, each client shares a label distribution vector with the server, which is used to estimate pairwise similarities and partition clients into clusters. This grouping defines an intermediate structure between individual clients and the global model and remains fixed throughout training. The framework is agnostic to the choice of similarity measure and clustering algorithm.

Hierarchical Aggregation: Aggregation is performed at two levels. At the cluster level, local models within each cluster are aggregated using data-size-based weighting:

$$\mathcal{M}_g^{e+1} = \sum_{k \in \mathcal{C}_g} w_k \mu_k^e, \quad w_k = \frac{n_k}{\sum_{j \in \mathcal{C}_g} n_j}, \quad (5.6)$$

where n_k denotes the number of local samples at client k , and \mathcal{C}_g is number of clients belongs to cluster g .

At the global level, aggregation incorporates both dataset size and cluster-level importance:

$$\mathcal{M}^{(e+1)} = \sum_k \tilde{w}_k \mu_k^{(e)}, \quad \tilde{w}_k = \frac{n_k \rho_g^{(e)}}{\sum_j n_j \rho_{g(j)}^{(e)}}, \quad (5.7)$$

where $\rho_g^{(e)}$ denotes the priority score of cluster g , derived from smoothed cluster-level losses, and $g(k)$ maps client k to its cluster. This formulation assigns greater influence to clusters exhibiting better training dynamics.

Hierarchical knowledge transfer. To mitigate client drift under non-IID data, DEFPT incorporates a hierarchical knowledge distillation mechanism. Each client

receives both the global model (as initialization of the student) and its cluster model (as a teacher), and optimizes a combined objective consisting of cross-entropy loss and a knowledge distillation term. This encourages clients to align with distributionally similar peers while maintaining global consistency.

Overall, DEFFT integrates clustering, cluster-aware aggregation, and hierarchical knowledge distillation into a unified framework that improves robustness and performance in heterogeneous federated settings.

5.3.1 Main Findings

Client similarities can be identified in a way that is operationally useful. Clustering based on client-level representations produces meaningful groups across datasets, enabling a stable intermediate structure for training. These groups preserve non-IID heterogeneity while organizing clients in a way that supports structured aggregation.

DEFFT improves low-performing and data-scarce clients. DEFFT consistently improves the performance of low- and mid-performing clients relative to both FEDAVG and Q-FEDAVG. This effect is driven by cluster-aware aggregation, which reallocates influence toward more difficult client groups, and by knowledge distillation, which provides a stabilizing signal for data-scarce clients. The knowledge distillation gain is largest in the early stages of training and is consistently higher for smaller clients than for larger ones. This suggests that the cluster-level teacher model provides a useful regularization signal particularly for data-scarce clients, helping stabilize local optimization and mitigate drift. Thus, the benefit of the learned structure is not only at the aggregation level but also in the transfer of information within clusters.

The learned structure remains relevant during training. Cluster-level importance weights adapt during early rounds and stabilize without collapsing to uniform values. This indicates that the learned grouping continues to influence optimization throughout training, rather than serving only as an initialization mechanism.

Simultaneous improvement in performance and fairness. Across controlled and benchmark settings, DEFFT reduces inter-client performance variance and improves lower-tail performance, while maintaining or improving average accuracy. These gains are distributed across the client population rather than concentrated in a subset of clients.

Generalization across heterogeneous settings. The benefits of DEFFT persist across datasets with varying levels of heterogeneity. In settings where label distributions are insufficient (e.g., FEMNIST), embedding-based representations still enable effective clustering, demonstrating that the approach generalizes beyond simple similarity measures.

Compared with loss-based reweighting, structure-aware aggregation is more con-

sistent. Across datasets, Q-FEDAVG provides some fairness improvements but does so less consistently. In several cases it improves selected tail metrics while failing to improve mean accuracy or leaving substantial residual variance. In contrast, DEFFT produces a more coherent pattern: the client-performance distributions become tighter, the left tail improves, and the average performance is either improved or remains competitive. This suggests that organizing clients through latent similarity provides a stronger foundation for fairness-aware optimization than reweighting clients only through instantaneous loss.

Structure-aware aggregation is more consistent than loss-based reweighting. Compared to loss-based methods such as Q-FEDAVG, DEFFT produces more consistent improvements across metrics on all evaluated datasets. While loss-based reweighting improves selected clients, it often leaves residual variance or degrades average performance. In contrast, structure-aware aggregation yields more balanced and stable outcomes.

5.3.2 Answer to RQ3.

The results show that latent label-distribution similarities among clients can be identified by constructing client-level representations that capture the structure of local data and then clustering clients into distributionally similar groups. For datasets with relatively clean label-distribution signals, this can be done directly using divergence-based clustering on label proportions. For more sparse and irregular settings, such as FEMNIST, more robust client embeddings are needed to capture semantic composition, concentration, and data scale before clustering.

Once identified, these similarities can be incorporated into FL through a hierarchical training structure in which:

- Clients first contribute within similarity-based clusters,
- Cluster-level importance is used to guide global aggregation, and
- Cluster models act as teachers for hierarchical knowledge distillation.

This distribution-aware design improves fairness by reducing inter-client performance dispersion and strengthening lower-tail client performance, while maintaining or improving global model accuracy. The effect is strongest in settings where client heterogeneity is substantial and where standard averaging or loss-based reweighting fails to account for underlying distributional structure.

Therefore, the answer to RQ3 is that latent client similarities should be identified through client-level distributional representations and incorporated through clustered aggregation and intra-cluster knowledge transfer. This yields a more robust balance

between global performance and fairness than flat aggregation schemes that ignore the internal structure of client heterogeneity.

5.4 Summary and Key Takeaways

This chapter has addressed the three research questions by progressively examining heterogeneous client behavior, and mitigating and modeling heterogeneity to reduce performance disparities in FL. The findings establish a transition from observation to mitigation and, ultimately, to distribution-aware learning.

First study answers RQ1 and shows that client behavior can be systematically characterized using model-based signals. In particular, eccentricity-based measures reveal that client deviations are not random but reflect underlying differences in data distribution, training dynamics, and model interaction. These deviations evolve over time, indicating that client behavior is inherently dynamic and requires continuous monitoring.

Second study answers RQ2 and demonstrates that reducing inter-client performance disparities requires more than observing client behavior. Adaptive aggregation improves robustness but is insufficient on its own. By combining performance-aware weighting with targeted boosting of under-performing clients, it is possible to achieve both competitive global performance and more balanced outcomes across clients.

Third study answers RQ3 and shows that further improvements of fairness can be achieved by explicitly modeling the distribution of client heterogeneity. By organizing clients into similarity-based groups and enabling hierarchical aggregation and knowledge transfer, distribution aware learning provides a more consistent and robust approach to balancing performance and fairness than flat aggregation or loss-based reweighting.

Overall, the results show that effective FL under heterogeneous data requires moving beyond independent client treatment toward behavior-aware, performance-aware, and distribution-aware learning strategies. Moreover, the three proposed approaches differ in their capabilities for monitoring, contribution evaluation, and fairness improvement.

- **Study 1 (Eccentricity-based monitoring):** Provides explicit monitoring of client behavior through signals derived from model dynamics. It enables partial evaluation of client contributions by identifying influential and anomalous clients. However, it does not introduce any mechanism to modify training dynamics and therefore does not reduce inter-client performance disparities.
- **Study 2 (FedABoost):** Introduces adaptive aggregation and boosting to actively reduce performance disparities across clients. Client behavior is partially

monitored through aggregation weights (α_k), which reflect local performance and implicitly capture contribution. This approach improves fairness but operates at the level of individual clients and does not explicitly model relationships among them.

- **Study 3 (DEFFT):** Extends beyond individual client adaptation by explicitly modeling similarities among clients. It organizes clients into clusters and enables distribution-aware aggregation and knowledge transfer. This approach improves fairness and robustness more consistently, particularly for low-performing clients, but depends on the quality of the similarity representation.

Key takeaway. The three studies can be interpreted as a structured progression from understanding to controlling and finally exploiting client heterogeneity. The first study establishes that client behaviour is not random but rather exhibits a measurable and evolving structure that can be captured through model-based signals. However, this study is purely observational and does not influence training. The second study addresses this limitation by introducing adaptive mechanisms that act on these differences, rebalancing contributions and improving under-performing clients. While this reduces disparities, it still treats clients largely as independent entities. The third study removes this assumption by explicitly modeling relationships between clients, organizing them into distributionally similar groups and enabling structured aggregation and knowledge transfer. This creates a coherent progression from identifying heterogeneity, to mitigating its negative effects, and finally to leveraging its underlying structure. Each study can be interpreted as resolving a limitation of the previous one, moving from static observation to dynamic intervention and ultimately to structure-aware optimization that aligns performance and fairness in FL.

6 Conclusion and Future Works

6.1 Conclusion

This thesis investigated how client heterogeneity affects learning dynamics and performance disparities in FL. The central objective was to understand, control, and leverage client heterogeneity to improve both model performance and fairness across clients.

The findings demonstrate that client behavior in FL is not random but exhibits measurable patterns. First, it was shown that client behavioral deviations can be systematically characterized using model-based signals. In particular, eccentricity-based measures enable continuous monitoring of client dynamics, revealing both persistent and transient deviations that are not captured by standard performance metrics.

Second, the results show that reducing inter-client performance disparities requires active intervention in the training process. Performance-aware aggregation improves robustness but is insufficient on its own. By combining adaptive weighting with targeted boosting of under-performing clients, it is possible to achieve both competitive global performance and more balanced outcomes across clients.

Third, the thesis demonstrates that further improvements can be achieved by explicitly modeling the structure of client heterogeneity. By organizing clients into distributionally similar groups and enabling hierarchical aggregation and knowledge transfer, distribution-aware learning delivers more consistent improvements in fairness and robustness than approaches that treat clients independently.

Taken together, these contributions establish a progression from behavior characterization to performance-aware intervention and finally to distribution-aware learning. The results show that effective FL under heterogeneous data requires moving beyond independent client treatment toward approaches that explicitly account for both behavioral and structural properties of client populations.

6.2 Future Work

While this thesis provides a framework for understanding and addressing client heterogeneity in federated learning, several directions remain for further investigation.

Efficient eccentricity-based monitoring. The eccentricity-based approach for client behavior monitoring relies on *Hessian* based representations, which can be computationally expensive in large-scale settings. Ongoing work focuses on reducing this computational overhead through more efficient approximations, while preserving the ability to capture clients behavioral deviations. In addition, further evaluation of the proposed monitoring framework under diverse federated settings is required to better understand its robustness and practical applicability.

Client drift and dynamic behavior. The analysis in this thesis shows that client behavior evolves over time. While the proposed methods partially address client drift through mechanisms such as hierarchical knowledge distillation, they do not explicitly model or predict drift dynamics. Future work could focus on developing methods to better capture and anticipate changes in client behavior, enabling more stable training under non-stationary conditions.

Integration with real-world applications. The proposed methods are evaluated primarily in controlled experimental settings using benchmark datasets. Applying these approaches in real-world scenarios introduces additional challenges, including data sparsity and heterogeneous data distributions across clients. One promising direction is the application of FL in sensitive domains such as mental healthcare, where client data is limited, highly heterogeneous, and often imbalanced and weakly labeled. Extending the proposed methods to such settings requires ensuring robust performance under data scarcity while maintaining fairness across clients.

7 Experiences and Learning Outcomes

Before starting this PhD journey, I was working as a machine learning engineer in industry, where the focus was on building and deploying models for practical use cases. While prior experience emphasized achieving performance under practical constraints, research required a deeper understanding of why models behave differently under varying conditions. In particular, this shift involved moving beyond treating deep learning models as black-box tools toward analyzing their optimization dynamics, generalization behavior, and sensitivity to data distributions. In federated learning, such understanding became essential, as challenges like data heterogeneity, limited observability, and decentralized training expose limitations not evident in centralized settings. Consequently, the focus evolved from improving performance alone to understanding and addressing model behavior through principled, system-level design.

The first study in this thesis was conducted in collaboration with the *Ericsson Research* team and provided a strong entry point into academic research. At the time I joined, the initial research gap had already been identified, allowing me to focus on understanding the problem while contributing to the conceptualization and development of the approach. This experience helped me quickly integrate into the academic environment, adapt to collaborative research workflows, and gain early exposure to structured practices, including experimental design, analysis, and scientific writing.

A central learning outcome emerged during the development of the second study (FEDABOOST). The work was initially rejected twice, and the critical, constructive feedback received during these review cycles highlighted weaknesses in the problem definition, the presentation of the idea, and the evaluation. Addressing these issues required substantial refinement of the problem formulation, explicit declaration of assumptions, and more rigorous experimental design. This process directed to a more structured and disciplined approach to research, improving both clarity and methodological rigor. Importantly, the lessons learned from this experience extended beyond the study itself and contributed to a noticeable improvement in the quality and maturity of subsequent work.

Another key aspect of my development was learning to navigate uncertainty, par-

ticularly during the DEFFT study. The initial research direction did not lead to a satisfactory outcome and required reassessment and partial reformulation. Although the work initially drew inspiration from swarm intelligence methods such as *Spider Monkey Optimization*, it became clear that this approach was not directly applicable and that a step back was needed to rethink the problem. Some intermediate ideas had to be abandoned, which strengthened my ability to critically analyze limitations and explore alternative approaches. The final direction of DEFFT emerged through this process, leading to a modified perspective on structured interactions among clients. This experience reinforced the importance of flexibility and the ability to reinterpret problems rather than adhering rigidly to initial ideas.

In parallel with research, I actively engaged in academic and professional development activities. I participated in several conferences as an author, oral presenter, and poster presenter. These experiences strengthened my ability to communicate complex technical ideas clearly and to defend them under questioning. They also improved my public speaking and my ability to think and respond under pressure. Managing conference deadlines alongside ongoing research further developed my organizational and time-management skills.

I was also involved in teaching activities at the university. Explaining material to students required clarity and structure, often revealing gaps in my own understanding. Teaching therefore served both as a means of knowledge transfer and as a mechanism for strengthening foundational knowledge, while also improving my ability to communicate with audiences of varying levels of expertise.

Beyond the academic work, this period involved adapting to a new country, culture, and academic environment while maintaining research progress. This required developing independence, resilience, and the ability to manage both personal and professional challenges over an extended period. Completing the licentiate phase marked an important milestone in this progression.

Overall, this journey is shaping my development from an application-oriented engineer to a researcher capable of defining problems, analyzing complex systems, and developing structured solutions. The key learning outcomes include critical thinking, rigorous problem formulation, the ability to handle uncertainty, and effective communication, which together form the foundation for my continued work as an independent researcher.

Bibliography

- [1] T. Li et al. *Federated Optimization in Heterogeneous Networks*. 2020. arXiv: 1812.06127 [cs.LG].
- [2] M. Mohri and et al. *Agnostic Federated Learning*. 2019. arXiv: 1902.00146 [cs.LG].
- [3] T. Li and et al. *Ditto: Fair and Robust Federated Learning Through Personalization*. 2021. arXiv: 2012.04221 [cs.LG].
- [4] B. McMahan et al. “Communication-efficient learning of deep networks from decentralized data.” In: *AI and Statistics*. PMLR. 2017, pp. 1273–1282.
- [5] P. Angelov. “Anomaly detection based on eccentricity analysis.” In: *2014 IEEE Symposium on Evolving and Autonomous Learning Systems (EALS)*. 2014, pp. 1–8. DOI: 10.1109/EALS.2014.7009497.
- [6] G. K. Nilsen and et al. *Efficient Computation of Hessian Matrices in TensorFlow*. 2021. arXiv: 1905.05559 [cs.LG].
- [7] T. Hastie, S. Rosset, J. Zhu, and H. Zou. “Multi-class adaboost.” In: *Statistics and its Interface 2.3* (2009), pp. 349–360.
- [8] T.-Y. Lin et al. “Focal loss for dense object detection.” In: *Proc. of the IEEE Int. Conference on Computer Vision*. 2017, pp. 2980–2988.
- [9] G. Hinton, O. Vinyals, and J. Dean. “Distilling the knowledge in a neural network.” In: *arXiv preprint arXiv:1503.02531* (2015).
- [10] J. D. Banfield and A. E. Raftery. “Model-based Gaussian and non-Gaussian clustering.” In: *Biometrics* (1993), pp. 803–821.
- [11] G. Wang and et al. “Measure Contribution of Participants in Federated Learning.” In: *2019 IEEE International Conference on Big Data (Big Data)*. 2019, pp. 2597–2604. DOI: 10.1109/BigData47090.2019.9006179.
- [12] T. Wang, J. Rausch, C. Zhang, R. Jia, and D. Song. *A Principled Approach to Data Valuation for Federated Learning*. arXiv:2009.06192 [cs, stat]. Sept. 2020. DOI: 10.48550/arXiv.2009.06192. URL: <http://arxiv.org/abs/2009.06192> (visited on 01/26/2024).

- [13] S. K. Shyn, D. Kim, and K. Kim. *FedCCEA : A Practical Approach of Client Contribution Evaluation for Federated Learning*. arXiv:2106.02310 [cs]. June 2021. DOI: 10.48550/arXiv.2106.02310. URL: <http://arxiv.org/abs/2106.02310> (visited on 01/18/2024).
- [14] Z. Liu, Y. Chen, Y. Zhao, H. Yu, Y. Liu, R. Bao, J. Jiang, Z. Nie, Q. Xu, and Q. Yang. “Contribution-Aware Federated Learning for Smart Healthcare.” English. In: vol. 36. 2022, pp. 12396–12404. ISBN: 978-1-57735-876-3.
- [15] B. Yan, B. Liu, L. Wang, Y. Zhou, Z. Liang, M. Liu, and C.-Z. Xu. “FedCM: A Real-time Contribution Measurement Method for Participants in Federated Learning.” en. In: *2021 International Joint Conference on Neural Networks (IJCNN)*. Shenzhen, China: IEEE, July 2021, pp. 1–8. ISBN: 978-1-66543-900-8. DOI: 10.1109/IJCNN52387.2021.9534451. URL: <https://ieeexplore.ieee.org/document/9534451/> (visited on 11/21/2023).
- [16] X. Hu, C. Luo, D. Zeng, Z. Xu, P. Guo, and I. King. “Flexible Contribution Estimation Methods for Horizontal Federated Learning.” In: *2023 International Joint Conference on Neural Networks (IJCNN)*. ISSN: 2161-4407. June 2023, pp. 1–8. DOI: 10.1109/IJCNN54540.2023.10191625. URL: <https://ieeexplore.ieee.org/document/10191625> (visited on 01/18/2024).
- [17] L. Meng, Y. Wei, R. Pan, S. Zhou, J. Zhang, and W. Chen. “VADAF: Visualization for Abnormal Client Detection and Analysis in Federated Learning.” In: *ACM Transactions on Interactive Intelligent Systems* 11.3-4 (Sept. 2021), 26:1–26:23. ISSN: 2160-6455. DOI: 10.1145/3426866. URL: <https://dl.acm.org/doi/10.1145/3426866> (visited on 01/18/2024).
- [18] Q. Li, X. Wei, H. Lin, Y. Liu, T. Chen, and X. Ma. “Inspecting the Running Process of Horizontal Federated Learning via Visual Analytics.” In: *IEEE Transactions on Visualization and Computer Graphics* 28.12 (Dec. 2022). Conference Name: IEEE Transactions on Visualization and Computer Graphics, pp. 4085–4100. ISSN: 1941-0506. DOI: 10.1109/TVCG.2021.3074010. URL: <https://ieeexplore.ieee.org/document/9408377> (visited on 01/18/2024).
- [19] T. Li, M. Sanjabi, A. Beirami, and V. Smith. “Fair resource allocation in federated learning.” In: *International Conference on Learning Representations*. 2019.
- [20] S. Cui, W. Pan, J. Liang, C. Zhang, and F. Wang. “Addressing algorithmic disparity and performance inconsistency in federated learning.” In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 26091–26102.
- [21] J. Pei. “F3: Fair Federated Learning Framework with adaptive regularization.” In: *Knowledge-Based Systems* 316 (2025), p. 113392.

- [22] D. Y. Zhang, Z. Kou, and D. Wang. “Fairfl: A fair federated learning approach to reducing demographic bias in privacy-sensitive classification models.” In: *IEEE Int. Conference on Big Data*. 2020, pp. 1051–1060.
- [23] Y. H. Ezzeldin et al. “Fairfed: Enabling group fairness in federated learning.” In: *Proc. of the AAAI Conference on AI*. Vol. 37. 6. 2023, pp. 7494–7502.
- [24] Z. Shi, L. Zhang, Z. Yao, L. Lyu, C. Chen, L. Wang, J. Wang, and X.-Y. Li. “Fedfair: A model performance-based fair incentive mechanism for federated learning.” In: *IEEE Transactions on Big Data* 10.6 (2022), pp. 1038–1050.
- [25] T. Zhu, Y. Lin, Y. Qu, Z. Liu, Y. Luo, T. Mao, and Z. Chen. “Federated learning with empirical insights: Leveraging gradient historical experiences for performance fairness.” In: *Pervasive and Mobile Computing* (2025), p. 102061.
- [26] A.-A. Bendoukha, H. H. Arcolezi, N. Kaaniche, A. Boudguiga, R. Sirdey, and P.-E. Clet. “FADE: Federated Aggregation with Discrimination Elimination.” In: *Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency*. 2025, pp. 3182–3195.
- [27] J. Wang et al. “Tackling the objective inconsistency problem in heterogeneous federated optimization.” In: *Advances in neural infor. proc. syst.* 33 (2020), pp. 7611–7623.
- [28] S. P. Karimireddy et al. “Scaffold: Stochastic controlled averaging for federated learning.” In: *International conference on ML*. PMLR. 2020, pp. 5132–5143.
- [29] Q. Li, B. He, and D. Song. “Model-contrastive federated learning.” In: *Proc. of the IEEE/CVF conf. on comp. vision and pattern recognition*. 2021, pp. 10713–10722.
- [30] F. Sattler, K.-R. Müller, and W. Samek. “Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints.” In: *IEEE transactions on neural networks and learning systems* 32.8 (2020), pp. 3710–3722.
- [31] J. Zheng, S. Zhao, P. Hu, and X. Shen. “CFLKD: Clustered Federated Learning via Cross-Group Knowledge Distillation.” In: *Future Generation Computer Syst.* (2025), p. 108253.
- [32] Z. Leng, C. Zhang, G. Long, R. Xia, and B. Yang. “Distilling A Universal Expert from Clustered Federated Learning.” In: *arXiv e-prints* (2025), arXiv–2506.
- [33] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. “Gradient-Based Learning Applied to Document Recognition.” In: *Proceedings of the IEEE*. Vol. 86. 11. 1998, pp. 2278–2324.

- [34] T. Lin et al. “Ensemble distillation for robust model fusion in federated learning.” In: *Advances in neural information processing systems* 33 (2020), pp. 2351–2363.
- [35] A. Krizhevsky. *Learning Multiple Layers of Features from Tiny Images*. Tech. rep. University of Toronto, 2009.
- [36] S. Caldas, P. Wu, T. Li, J. Konečný, H. B. McMahan, V. Smith, and A. Talwalkar. “LEAF: A Benchmark for Federated Settings.” In: *arXiv preprint arXiv:1812.01097* (2019).
- [37] X. Lan et al. “Federated Learning for Performance Prediction in Multi-Operator Environments.” In: *ITU Journal on Future and Evolving Technologies* 4.1 (2023).
- [38] T. Li et al. “Ditto: Fair and robust federated learning through personalization.” In: *Int. Conference on ML*. PMLR, 2021, pp. 6357–6368.
- [39] M. G. Kendall and J. D. Gibbons. *Rank correlation methods*. Ed. by E. A. (Ed.) A Division of Hodder & Sloughton (5th ed.), London: Oxford University Press, 1990.
- [40] R. K. Jain, D.-M. W. Chiu, W. R. Hawe, et al. “A quantitative measure of fairness and discrimination.” In: *Eastern Research Laboratory, Digital Equipment Corporation, Hudson, MA* 21.1 (1984), pp. 2022–2023.
- [41] D. M. Chiu. *A quantitative measure of fairness and discrimination for resource allocation in shared computer systems*. Tech. rep. Digital Equipm. Corporat., 1984.

Paper I

Clients Behavior Monitoring in Federated Learning via Eccentricity Analysis

Tharuka Kasthuri Arachchige, Selim Ickin, Shahrooz Abghari, Veselka Boeva

In: Proceedings of the IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS), 2024, Madrid, Spain, June 2024

This research was funded partly by the Knowledge Foundation, Sweden, through the Human-Centered Intelligent Realities (HINTS) Profile Project (contract 20220068).

Abstract

The success of Federated Learning (FL) hinges upon the active participation and contributions of edge devices as they collaboratively train a global model while preserving data privacy. Understanding the behavior of individual clients within the FL framework is essential for enhancing model performance, ensuring system reliability, and protecting data privacy. However, analyzing client behavior poses a significant challenge due to the decentralized nature of FL, the variety of participating devices, and the complex interplay between client models throughout the training process. This research proposes a novel approach based on eccentricity analysis to address the challenges associated with understanding the different clients' behavior in the federation. We study how the eccentricity analysis can be applied to monitor the clients' behaviors through the training process by assessing the eccentricity metrics of clients' local models and clients' data representation in the global model. The Kendall ranking method is used for evaluating the correlations between the defined eccentricity metrics and the clients' benefit from the federation and influence on the federation, respectively. Our initial experiments on a publicly available data set demonstrate that the defined eccentricity measures can

provide valuable information for monitoring the clients' behavior and eventually identify clients with deviating behavioral patterns.

1 Introduction

Machine Learning Operations (MLOps) is a set of practices to manage and improve ML workflows. It includes high-level functions such as model initialization, training, hyper-parameter tuning, model sharing, and many more. In recent 3rd Generation Partnership Project (3GPP) standardization efforts in 5G, an exposure interface is introduced for Network Data Analytics Function (NWDAF) to provide consumers with rich data sets that contain network performance and mobility information from different network and application functions. As the number of use cases and consumers grow, the number of ML models that realize the use cases increases proportionally, leading to scalability issues in MLOps.

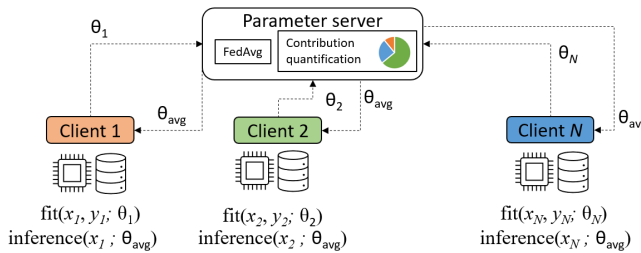


Figure 1: Client contribution quantification in FL.

Sustainable MLOps requires a set of actions that should be taken to maximize model efficacy with minimum operational costs concerning memory, storage, computation, and communication. This makes functions such as model sharing and collaborative training appealing. MLOps deployed in mobile networks may include a scenario where a global model is trained collaboratively by decentralized client data sets, where these data sets are collected locally from similar functional units potentially sourced from various network equipment with a strict data-sharing policy. In this case, sharing raw data sets with consumers may not be possible, which makes privacy-preserving distributed learning techniques such as Federated Learning (FL) appealing. In FL, the learning from different clients, the so-called learned model weights on decentralized data sets, may be heterogeneous, meaning that the participants of such federation may have different contributions and benefits to/from the global model as illustrated in Fig. 1. Therefore, methods to quantify the contribution of the model trained with individual client data sets are necessary to ensure fair collaboration and incentivize clients to participate actively in the training process.

In distributed ML model training in telecommunications, incentive quantification is required, as distributed learning may consist of model updates received from

collaborating honest-but-curious [1] clients. As such, it is important to quantify the contribution of every local model update in the global model for one main reason. When the model fails to perform its task after joint training, it should locate underlying reasons, e.g., model update issues in all participating entities, and mitigate the problem. Identified participants that degrade the model performance can be temporarily removed from the federation. This way, global model efficacy is expected to be sustained while reducing the communication and computation costs. A healthy incentive mechanism should motivate potential participants with an appropriate reward mechanism to motivate all clients to actively be part of the federation and improve the global model with their contributions. Moreover, such feedback indicating the contribution level may trigger the selection of important data samples for the client when training the local model.

In this study, we introduce the eccentricity metric to analyze the correlation between the behavior of clients' local models during training and the quality of the federation. Additionally, we aim to explore the correlation between the representation of each client's data in global models and the quality of the federation. In particular, we evaluate the eccentricity of the clients' local models and study their correlation with the clients' influence and benefit. The eccentricity scores can be calculated for each client's local model parameters at every or selected global training rounds. In that way, we can eventually monitor how the eccentricity of the client's local model is correlated with its influence on the shared model and, in addition, how it affects the client's benefit from being involved in the federation. The behavior of the global model eccentricity related to the representation of clients' data can also be assessed to gain further understanding. To the best of our knowledge, this is a novel approach within the scope of FL for analyzing how the clients' behaviors can be used to monitor and quantify their contribution to the global model.

2 Background

2.1 Federated Learning

FL is an ML approach, first introduced by Google in 2017 [2], where multiple decentralized devices (clients) train a shared global model without transmitting their data.

FL begins with initializing a global model using random weights and broadcasting the model to clients. These clients train the received model with their available data sets locally. After training, the updated model from each client is sent back to the central server. The server aggregates the received models using averaging techniques such as *FedAvg* method [3], which involves computing the arithmetic mean of all client model weights. According to [4], three different FL categories can be recognized: *horizontal* FL, where clients hold distinct samples of the same

features, *vertical* FL [5] where clients possess unique attributes for shared instances, and *transfer* FL, a specialized form focused on adapting knowledge from one domain to enhance models in related domains. Our study focuses exclusively on horizontal FL scenarios in cross-silos settings where a small number of clients, such as organizations, companies, or local data centers, accommodate various data distribution patterns and knowledge transfer requirements.

We have implemented a setup considering the formal definition of FL given by the naive *FedAvg* algorithm [3]. Suppose a group of k distributed clients is involved in collaborative training of a global model \mathcal{M} under the supervision of a central server without sharing their private data. At each global round (communication round) i , for $i = 1, 2, \dots, n$, each client j , for $j = 1, 2, \dots, k$, trains a local model μ_j on its private data set \mathcal{D}_j and sends the model parameters to the server. The main steps of the algorithm are summarized in Algorithm 1.

Algorithm 1 FL Algorithm

- 1: Initialize the global model \mathcal{M} with random weights
- 2: **for** $i = 1, 2, \dots, n$ **do**
- 3: Share \mathcal{M} with all clients
- 4: **for** $j = 1, 2, \dots, k$ **do**
- 5: Set μ_j with \mathcal{M} weights.
- 6: Train μ_j with \mathcal{D}_j for n ' local training rounds
- 7: Send μ_j weights to central server
- 8: **end for**
- 9: \mathcal{M} weights \leftarrow Average local model weights

10: **end for**

Where:

n is the number of global training rounds.

k is the number of clients participating in each round.

2.2 Contribution Evaluation via Deletion

The influence (contribution) of each party involved in horizontal FL can be calculated using the client deletion approach [6]. In this strategy, the data set provided by a certain client (e.g., client i) is skipped, and the global model (\mathcal{M}') is built considering other clients' local models. The difference in the prediction results between the \mathcal{M}' and \mathcal{M} is assessed using Eq. I.1 in order to quantify the influence of the skipped client i .

$$Influence^i = \frac{1}{s^v} \sum_{j=1}^{s^v} |\hat{y}_j - \hat{y}_j^{-i}|, \tag{I.1}$$

where s^v is the size of the validation data set, \hat{y}_j is the prediction on j th instance made by \mathcal{M} , and \hat{y}_j^{-i} is the prediction on j th instance made by \mathcal{M}' , trained without the contribution of the i^{th} client.

2.3 Eccentricity Analysis

The typicality and eccentricity-based data analytics (TEDA) as an alternative statistical framework has been introduced in [7]. The proposed framework is a systematic methodology that does not require prior assumptions, i.e., it is entirely based on the data and its mutual distribution in the data space and does not require the independence of the individual data samples and an infinite number of data samples.

The *eccentricity*, ξ_k^n , of a particular data sample x_k when $n > 2$ non-identical data samples are available is defined as the sum of the distances of x_k to all other existing data samples, divided by the sums of the distances from all data samples to all other data samples, see Eq. I.2 below:

$$\xi_k^n = \frac{2 \sum_{i=1}^n d_{ki}}{\sum_{i=1}^n \sum_{j=1}^n d_{ij}}, \quad (\text{I.2})$$

where d_{ij} is the distance between two given data samples x_i and x_j . The *typicality* τ_k^n of x_k is defined as the complement of the eccentricity, as follows:

$$\tau_k^n = 1 - \xi_k^n. \quad (\text{I.3})$$

The normalised *eccentricity* and *typicality* can also be defined as follows:

$$\zeta_k^n = \frac{\xi_k^n}{2} \text{ and } t_k^n = \frac{\tau_k^n}{n-2}, \quad (\text{I.4})$$

where $\sum_{i=1}^n \zeta_i^n = 1$ and $\sum_{i=1}^n t_i^n = 1$, respectively. In addition, $0 < \zeta_i^n < \frac{1}{2}$ and $0 < t_i^n < \frac{1}{n-2}$, for any $i \in \{1, 2, \dots, n\}$. Note that when normalized eccentricity is above $1/n$, the data sample is considered untypical or eccentric. In the opposite case, i.e., when the value of typicality is above $1/n$, then the data sample is rather typical.

2.4 Neural Network Parameters and Hessian Matrix

A Neural Network (NN) typically contains an input layer, a few hidden layers, and an output layer. Input data is received through the input layer, processed through hidden layers, and then directed to the output layer. These layers are interconnected with their associated parameters called weights and biases. These weights and biases are initially set to random values and updated iteratively during training to minimize the prediction error over time. The prediction error, usually assessed using a loss function, is utilized for computing gradients. These gradients guide the update of weights and biases through optimization techniques such as gradient descent. The loss

function assesses the discrepancy between prediction and target values, functioning as an indicator of the model’s performance.

Hessian is a mathematical function that describes the curvature of a scalar function $f : \mathbb{R}^X \rightarrow \mathbb{R}$ at a certain point X [8]. The Hessian matrix can be used to capture curvature information of the loss function concerning changes in the NN’s parameters. Specifically, it provides insights into how small changes in individual parameters can collectively influence the change in the overall loss function. Let Mean Absolute Error (MAE) $f(\beta_0, \beta_1) = \sum_i |y_i - \beta_0 - \beta_1 x_i|$ be the scaled function given the parameters β_0 and β_1 , where y_i and x_i are the target value and the input value at i^{th} data point, respectively. The gradients of this scalar field is $\nabla f = \langle \sum_i \text{sign}(y_i - \beta_0 - \beta_1 x_i), \sum_i x_i \text{sign}(y_i - \beta_0 - \beta_1 x_i) \rangle$. Now, each component of ∇f is itself a scalar field. Taking gradients of those and setting them to be rows of a matrix, one can compute the Jacobian matrix. The Hessian of f is the same as the Jacobian of ∇f [9].

2.5 Rank correlation

Kendall’s rankings comparison method can be used to compare any two rankings [10]. It compares the nodes (clients) in pairs, i.e., the positions of pair clients within both rankings. If the position of client i is related to the position of client j in both rankings monotonically in the same direction, then this pair is well correlated. Kendall’s rank correlation coefficient (also known as Kendall’s \mathcal{T}) is a value in the range $[-1, 1]$, where 1 means that two rankings are perfectly correlated, 0 means no correlation between the two rankings, and -1 means that they are entirely different (i.e., inversely associated). If two rankings x and y are produced on the given clients, then Kendall’s \mathcal{T} can be calculated using Eq. I.5 [11].

$$\mathcal{T} = \frac{A - D}{\sqrt{(A + D + T_x)(A + D + T_y)}}, \quad (\text{I.5})$$

where, A is the number of pairs in agreement, D is the number of pairs in disagreement, T_x is the number of pairs tied w.r.t. x , and T_y is the number of pairs tied w.r.t. y .

3 Related Work

In the literature, there exist many different methods studied within the area of FairFL [12], including techniques such as Agnostic FL [13] and Quantization-FL [14] as well as personalization of FL models such as DITTO [15], FedProx [16], cluster-based FL [17]. Although those methods help mitigate performance issues, there is still a need for further understanding of the level of contribution of each participant on a collaboratively

trained model. Therefore, in this section, we discuss the related works in contribution evaluation and client behavior monitoring in FL.

3.1 Clients Contribution Evaluation in FL

Assessing the contribution of the participants in FL is challenging [18]. One of the primary challenges is data valuation, which can be subjective and task-specific while also being vulnerable to tampering. Evaluating participant contributions is a classic cooperative game problem, and it requires a fair and rational approach that can achieve an optimal balance among all participants. However, the evaluation process often involves exponential computational complexity, and data valuation through training models in FL can be time-consuming. While there has not been much research focused on evaluating client contributions in FL scenarios, a few interesting studies have been carried out in the literature.

In 2019, Guan et al. [6] proposed straightforward yet effective methods for equitably determining the contributions of multiple clients in horizontal FL contexts by using group instance deletion techniques and, in vertical FL, via a technique based on group Shapely values. Also, in 2021, a new approach called FedCCEA was introduced by Shyn et al. [19]. This method builds the Accuracy Approximation Model (AAM), which estimates a simulated test accuracy using inputs of sampled data size and extracts the clients' data quality and data size to measure client contribution. FedCCEA demonstrates superiority compared to methods such as Leave-one-out method (LOO) [6], Truncated Monte-Carlo Shapley Value (TMC-SV) [20] method through several experiments: client contribution distribution, client removal, and robustness test to partial participation. Further, FedCCEA allows the clients to select data size while providing evaluation in a timely manner regardless of the number of clients and precise estimation in non-IID settings. The Contribution-aware Federated Learning (CAFL) framework proposed in [21] provides an efficient and accurate approach to fairly assess FL participants' contribution to model performance without exposing their private data. This framework improves the FL model training protocol by distributing the best-performing intermediate models to participants for further training. The authors claim that CAFL is the first contribution-aware FL that has been successfully deployed in the healthcare industry.

For the real-time evaluation of contributions, a simple yet powerful method called FedCM [22] was introduced in 2021. It uses an attention mechanism to calculate the attention weight of all clients in FL. It considers both the current and the previous rounds to obtain the contribution rate of each client and updates the contribution at every global round. This aspect is crucial for FL systems to allocate computing power and communication resources. Experimental results show that FedCM is more sensitive to data quantity and quality in real time.

Recently, the paper [23] presents two plug-and-play methods, namely the Vector

Projection Method (VPM) and the Income Exclusion Method (IEM), which can be used to estimate the contribution of clients in FL. The authors have conducted experiments to demonstrate the effectiveness of these methods in contribution evaluation and show the convergence acceleration under the contribution-weighted aggregation mechanism.

3.2 Clients Behaviour Monitoring in FL

Li et al. [24] have discussed challenges when implementing FL systems where clients are autonomous and the server doesn't have full control over their actions. This can lead to clients intentionally or unintentionally deviating from the prescribed course of federated model training, leading to abnormal behaviors, such as becoming a malicious attacker or a malfunctioning client. To address this issue, they propose a solution that involves creating low-dimensional surrogates of model weight vectors and using them for anomaly detection on the server side. They have also evaluated the proposed solution through experiments on image classification model training over the FEMNIST dataset and showed that their detection-based approach outperformed conventional defense-based methods.

VADAF [25], a visual analytics system that helps interpret and analyze the FL training process. The system is designed to distinguish potential client anomalies in the FL setting. It uses a visualization scheme that supports massive training dynamics in the FL environment. It also introduces an anomaly detection method to detect potential client anomalies, which are further analyzed based on the client model's visual and objective estimation. Three case studies have demonstrated the effectiveness of VADAF in understanding the FL training process and supporting abnormal client detection and analysis. Another useful tool for analyzing client behavior in FL is HFLens [26]. This enables the inspection of the behaviors of the participating clients by supporting comparative visual interpretation at the overview, communication round, and client instance levels. HFLens aids users in analyzing the overall process of all clients, identifying potential anomalies, and assessing the contribution of each client. The system is evaluated through two case studies, and experts' feedback suggests that HFLens indeed assist in understanding and diagnosing the HFLens process better. In another publication [27], a technique has been introduced to identify misbehaving clients in FL with a cosine similarity-based technique to compute the similarity between the weights of the local and global models and further quantify the system's degree of deviation. A low cosine similarity score may indicate that the node misbehaves, as its local model does not align well with the global model.

4 Methodology

4.1 Data

An empirical study was performed on a publicly available data set [28]. The data set was obtained through experiments at a key-value store database in a cluster, where various measurements, such as CPU utilization per core, memory utilization, network utilization, and disk I/O, were provided at the kernel and service levels from the server and the clients during run-time, respectively. In this study, the target variable to predict was "average read latency", as estimating this value would allow an operator to take actions to keep it below agreed thresholds in service level agreements (SLA).

4.2 Problem Formulation and Eccentricity-based Method

Assume k clients are involved in a federation for training a global model. Each client i , for $i = 1, 2, \dots, k$, trains an ML model μ_i locally on its data set and sends the model parameters (weights and biases) to the central server, where a unified global model \mathcal{M} is created by averaging the model parameters received from all the clients. We assume that all clients participate in all rounds of the federation. Furthermore, the local model μ_i of client i , for $i = 1, 2, \dots, k$, is represented by a Hessian matrix \mathcal{H}_{μ_i} . The Hessian matrix of the global model \mathcal{M} is built on the training data set of each client i , for $i = 1, 2, \dots, k$, and denoted by $\mathcal{H}_{\mathcal{M}}^i$. In total, each client i is represented by two Hessian matrices \mathcal{H}_{μ_i} and $\mathcal{H}_{\mathcal{M}}^i$.

In the above context, we are interested in understanding clients' behavior in the distributed model training process. It would be useful if we could monitor and assess the client's behavior through the whole training process, e.g., at all or selected training rounds. The latter can provide an opportunity for analyzing and better understanding the clients' influence on the federated model. The identified clients' behavior patterns could eventually be used for earlier considerations and adjustments of the federation during the training process itself.

In this study, we address the above problem by calculating the eccentricity score ξ_i^L of each client i , for $i = 1, 2, \dots, k$, using the Hessian matrices of the client's local model parameters. This can be conducted at specially selected moments or at each training round through the whole learning process. We can also evaluate each client behavior i , for $i = 1, 2, \dots, k$, by assessing the eccentricity score ξ_i^G concerning the data representation of client i in the global model. The correlation of each client's ξ_i^L and ξ_i^G with its influence and benefit will be further studied.

The eccentricity score ξ_i^L of client i can be calculated as follows:

$$\xi_i^L = \frac{2 \sum_{j=1}^k d(\mathcal{H}_{\mu_i}, \mathcal{H}_{\mu_j})}{\sum_{l=1}^k \sum_{j=1}^k d(\mathcal{H}_{\mu_l}, \mathcal{H}_{\mu_j})}, \quad (\text{I.6})$$

where \mathcal{H}_{μ_i} is the Hessian matrix of client local model μ_i , and $d(\cdot, \cdot)$ is the Euclidean

distance between the Hessian matrices of two clients. This eccentricity score evaluates the client’s local model behavior.

The eccentricity score ξ_i^G of client i can be calculated as follows:

$$\xi_i^G = \frac{2 \sum_{j=1}^k d(\mathcal{H}_{\mathcal{M}}^i, \mathcal{H}_{\mathcal{M}}^j)}{\sum_{l=1}^k \sum_{j=1}^k d(\mathcal{H}_{\mathcal{M}}^l, \mathcal{H}_{\mathcal{M}}^j)}, \quad (\text{I.7})$$

where $\mathcal{H}_{\mathcal{M}}^i$ represents the Hessian matrix of the global model \mathcal{M} calculated using the data specific to client i , i.e., $\mathcal{H}_{\mathcal{M}}^i$ characterizes how the model \mathcal{M} performance changes with client i ’s data, and $d(\cdot, \cdot)$ is the Euclidean distance between the two Hessian matrices. This eccentricity score evaluates the global model behavior with respect to each client’s data representation in the global model.

We also propose to calculate the eccentricity scores for clients’ without transforming the NN local models into Hessian matrices. In this scenario, the weights and bias matrices are used directly to compare the clients’ local models. For each layer’s weights and biases, the respective eccentricity scores are computed separately and then averaged to get the overall eccentricity score, denoted by ξ_i^P . This measure evaluates each client’s local model eccentricity similarly to Eq. I.6, but does not spend resources on calculating the Hessian matrix of the model.

As described above, the eccentricity scores can be calculated for an individual or selected training rounds following step 8 in Algorithm 1. This allows the creation of a vector of eccentricity values, forming a unique signature for each client, representing its behavior during the training process.

The clients’ eccentricity signatures can be used for analyzing and better understanding how the clients’ behavioral patterns affect the quality of the federation, e.g., too many clients with deviating behavior through the training may imply a shared model that is not advantageous for all the clients. This methodological setup can be used for simulating and studying different FL scenarios.

The performance of the local models evaluated by MAE is compared against that of the global model to calculate the benefit each client gains from participating in the federation using Eq. I.8.

$$\mathcal{B}_i = \text{MAE}(\mu_i) - \text{MAE}(\mathcal{M}). \quad (\text{I.8})$$

4.3 Experimental and Training Setup

In the conducted experimentation, a regression problem was addressed using a publicly available dataset¹ from a distributed cluster, involving 24 distinct clients (nodes), each characterized by a data set containing 197 features and data points ranging from 19 400 to 26 500. A shallow NN with 3 layers was chosen as the training

¹Data Traces from KTH Data Center available online at <https://www.kaggle.com/datasets/jaliltaghia/data-traces-from-a-data-center-testbed>.

algorithm. The training process employed Stochastic Gradient Descent (SGD) as the optimization algorithm, with a learning rate set at 0.005. The MAE was chosen as the loss function. During training, mini-batch SGD was utilized with a batch size of 64. Through systematic exploration, it was determined that the optimal number of training epochs for this configuration was 500, resulting in the most effective convergence and performance enhancement, both in isolated training for each client and in the federated training setups. All of the following experimentation was carried out with a holdout validation set comprising 33% of the entire data, and this process was iteratively repeated to ensure the validation of the results. The experimental code is published on GitHub².

4.4 Applicability

The main benefit and difference from the existing contribution evaluation solutions of the eccentricity analysis-based approach proposed in this work is that although it is not applied as a separate evaluation procedure, it can be integrated into the training of the federated model. In that way, the results of the training process will lead to a built federated model, including clients' eccentricity signatures. The latter facilitates monitoring clients' behavior throughout the training process.

The three eccentricity-based measures proposed in this study are defined to be used in horizontal cross-silos FL scenarios with a small number of clients. However, these measures can scale to any NN model and be applied to any other data in the horizontal FL setup. The global model has been trained in approximately 15 minutes, calculating the ξ^L and ξ^G for one global round has taken around 8 minutes, and calculating the ξ^P for one global round has needed approximately 1 minute on a computer with an Apple M2 Max processor and 32GB memory in the current setup. However, it is worth mentioning that increasing the size of the NN and the size of the dataset of each client will lead to increased computational complexity. This limitation is planned to be addressed in our future work devoted to the scalability aspects of the proposed approach.

Note that the third measure defined, ξ^P , which is not based on the Hessian matrix calculation, can easily be generalized and used for other experimental horizontal FL scenarios, e.g., ones using ML models different from NN, and it does not have scalability issues as the other two.

Furthermore, we believe the proposed eccentricity-based evaluation approach can be applied to other distributed learning scenarios apart from horizontal FL. For example, the proposed approach can be used to assess the contribution of source models to the target model in transfer learning.

²<https://github.com/tharukackasthuri/FLTrack>

5 Empirical Evaluation and Results

The eccentricity-based measures defined in Section 4.2 are based on a well-established data analytics methodology (TEDA) that has been shown to be efficient in different outlier/novelty/change detection scenarios [7, 29]. In order to gain a better understanding of how our eccentricity-based method for monitoring clients’ behavior works, we have initially conducted a series of controlled experiments. These experiments have studied three different setups: (i) the first involves five clients, four of which with identical data, and one client uses significantly reduced volume of the same data set than the other four; (ii) the second includes five clients where four clients with the identical data and one client with the same volume, but different data; (iii) the third has five clients with identical data, but during the training one client has conducted a different (much lower) number of local training rounds than the others. Through these controlled experiments, we have observed that the eccentricity scores are indeed influenced by all three aspects, i.e., the reduced data volume and training epochs, and the distribution of data among the clients. These five clients have been identified as eccentric by our measures.

In addition to the above controlled experiments, we have further performed and explored two experimental scenarios on the full data set explained in Section 4.1. In the first scenario, entitled Ex.1, 500 global model updates are conducted, considering each client with one local training round. In the second experimental scenario, Ex.2, we have performed 25 local training rounds at each global round and trained the federated model over 20 global training rounds. In Ex.1, we have calculated clients’ benefit (Eq. I.8), influence (Eq. I.1), and eccentricity scores after all training rounds as presented in Table 1. In Ex.2, we do the same evaluation right after the first global round as given in Table 2. Our motivation behind these two experiments is to study and compare the defined eccentricity measures’ informativeness in two stages of the federated training process.

In Ex.1, most of the clients (17 out of 24) have positive benefit scores, showing that they distinctly benefit from participating in the federation. The six highest benefit scores in Table 1 are shown in **bold**. In Ex.2, on the other hand, all the clients have negative benefit scores. This may imply that all models at early training rounds learn generic knowledge and do not help each other, and later on, as they become more specialized, they have more to learn from each other.

To evaluate the clients’ influence on the federated model, we have chosen the deletion approach, discussed in Section 2.2, as the baseline approach and calculated the influence scores using Eq. I.1. The calculated influence scores in Ex.1 and Ex.2 are also presented in Table 1 and Table 2, respectively. The top six influence scores in the tables are shown in **bold**.

We have generated two rankings for all clients involved in the federation based on their benefits and influence scores. These rankings allow us to analyze whether clients who highly benefit from the federation are also important to the global model.

Table 1: Comparison of clients local models' benefits and influence, and eccentricity scores calculated in Ex.1.

Client	Benefit	Influence	Ecc. ξ^L	Ecc. ξ^G	Ecc. ξ^P
1	-0.0085	0.1055	0.0352	0.0367	0.0407
2	-0.0088	0.1187	0.0356	0.0471	0.0416
3	-0.0398	0.0890	0.0341	0.0475	0.0349
4	-0.0400	0.1447	0.0385	0.0784	0.0389
5	-0.0166	0.0954	0.0373	0.0445	0.0408
6	-0.0320	0.1218	0.0302	0.0467	0.0395
7	0.0205	0.1179	0.0318	0.0405	0.0384
8	0.0239	0.1098	0.0389	0.0414	0.0355
9	0.0245	0.0913	0.0409	0.0362	0.0426
10	-0.0502	0.1503	0.0473	0.0902	0.0364
11	0.0242	0.0928	0.0412	0.0366	0.0409
12	0.1735	0.1243	0.0437	0.0471	0.0430
13	0.0491	0.0799	0.0493	0.0309	0.0505
14	0.1152	0.1592	0.0404	0.0313	0.0392
15	0.0804	0.0829	0.0375	0.0315	0.0482
16	0.0681	0.1253	0.0486	0.0367	0.0374
17	0.0673	0.0821	0.0433	0.0335	0.0456
18	0.0843	0.0851	0.0443	0.0326	0.0412
19	0.0429	0.1029	0.0530	0.0342	0.0458
20	0.0406	0.1059	0.0397	0.0325	0.0378
21	0.0255	0.0983	0.0431	0.0352	0.0465
22	0.0327	0.1236	0.0438	0.0350	0.0385
23	0.0283	0.0837	0.0492	0.0370	0.0421
24	0.0434	0.0869	0.0530	0.0367	0.0539

We have applied Kendall’s rankings (Section 2.5) for this purpose (see Fig. 2a and Fig. 2b). The Kendall ranking scores for the two produced rankings of the clients in Ex.1 and Ex.2 with respect to benefit and influence is equal to -0.22 and -0.34 , respectively, i.e., for many of the clients, large influence to the federated model does not lead to high benefit and visa versa.

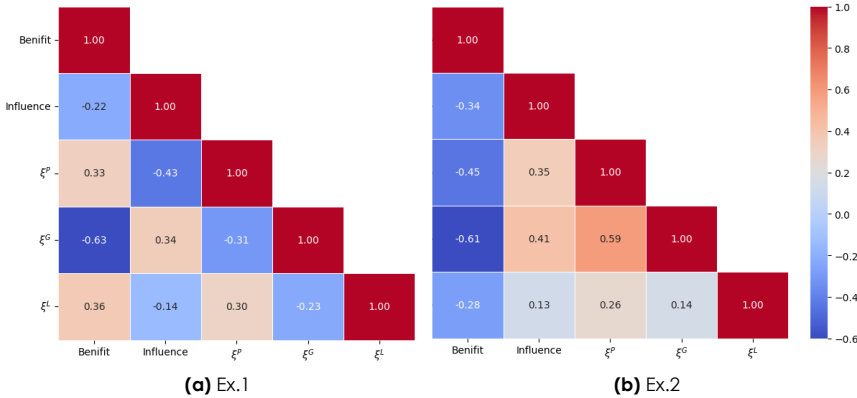


Figure 2: Heatmaps of Kendall ranking scores.

In Ex.1 and Ex.2, we have also calculated each client’s eccentricity scores, namely ξ^L , ξ^G , and ξ^P , described in Section 4.2. The calculated scores are presented in Table 1 and Table 2, respectively. The eccentric values above the threshold 0.0416

Table 2: Comparison of clients local models' benefits and influence, and eccentricity scores calculated in Ex.2.

Client	Benefit	Influence	Ecc. ξ^L	Ecc. ξ^G	Ecc. ξ^P
1	-0.0530	0.0991	0.0416	0.0383	0.0384
2	-0.0571	0.0931	0.0336	0.0428	0.0412
3	-0.0934	0.1073	0.0529	0.0553	0.0444
4	-0.1145	0.0996	0.0360	0.0672	0.0477
5	-0.0972	0.1000	0.0469	0.0571	0.0444
6	-0.1207	0.1083	0.0572	0.0449	0.0397
7	-0.0781	0.0988	0.0370	0.0368	0.0407
8	-0.0741	0.1013	0.0350	0.0376	0.0408
9	-0.0718	0.1028	0.0462	0.0469	0.0446
10	-0.1095	0.1339	0.0583	0.0695	0.0597
11	-0.0628	0.1094	0.0734	0.0440	0.0412
12	-0.0771	0.1027	0.0405	0.0440	0.0460
13	-0.0081	0.0839	0.0346	0.0328	0.0380
14	-0.0235	0.1367	0.0524	0.0322	0.0376
15	-0.0235	0.0854	0.0363	0.0341	0.0377
16	-0.0277	0.0907	0.0338	0.0339	0.0366
17	-0.0324	0.0766	0.0349	0.0345	0.0398
18	-0.0293	0.0856	0.0420	0.0347	0.0368
19	-0.0644	0.0918	0.0420	0.0336	0.0399
20	-0.0587	0.0869	0.0369	0.0341	0.0406
21	-0.0772	0.0885	0.0426	0.0353	0.0400
22	-0.0755	0.0994	0.0554	0.0367	0.0454
23	-0.0687	0.0792	0.0306	0.0364	0.0392
24	-0.0733	0.0780	0.0000	0.0375	0.0398

($1/n$, where $n = 24$ is the number of clients) are marked in **bold**.

As shown in Fig. 2a, the Kendall ranking between the ξ^L and the benefit in Ex.1 is 0.36, showing a significant correlation between them, while the correlation of ξ^L with influence is -0.14 , indicating no significant correlation. Interestingly, the correlations identified in Ex.2 (Fig. 2b) are opposite and very close to zero, i.e., ξ^L does not provide meaningful information about the clients' influences on the federated model at such an early stage of model training. In addition, in Ex.1, the Kendall correlation -0.63 between the clients' ξ^G and benefit implies a significant negative dependence, while between ξ^G and influence shows a positive correlation of 0.34. The correlations identified in Ex.2 are similar to Ex.1 (see Fig. 2b), i.e., there is a negative correlation between clients' ξ^G and their benefit (-0.61) and a positive correlation concerning clients' ξ^G and their influence (0.41). Evidently, even in the first round, ξ^G can identify clients that will potentially influence the federated model.

The Kendall rank correlation between the clients' ranking with respect to ξ^P and benefit in Ex.1 is 0.33 (see Fig. 2a). The Kendall ranking between the ξ^P and influence (-0.43) demonstrates a strong negative correlation. However, in Ex.2, the Kendall rankings scores between ξ^P and benefit (-0.45) and ξ^P and influence (0.35) have opposite signs but show comparatively strong correlations. This may indicate that eccentric clients contribute more to the global model at the earlier stage of training by supplying newly learned concepts without getting benefits. In general, eccentricity scores generated directly using NN parameters at a specific training checkpoint are not sufficient to make a firm conclusion.

In Ex.2, during each global training round, we have calculated the ξ^G and ξ^P (i.e., right after step 8 and before step 9 in Algorithm 1). In this way, we have obtained two eccentric signatures for each client as described in Section 4.2. These signatures are shown in Fig. 3 and Fig. 4, respectively.

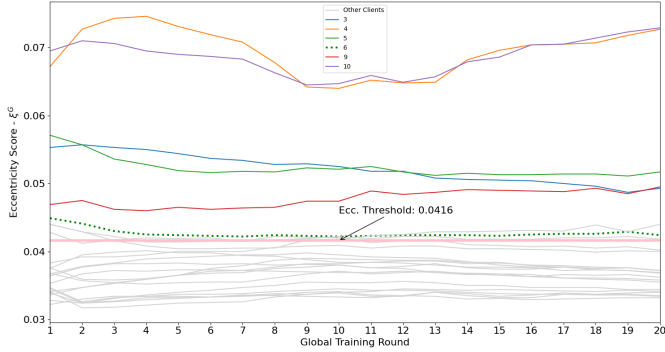


Figure 3: Clients' eccentricity signatures base on ξ^G metric calculated for each training round during the training process.

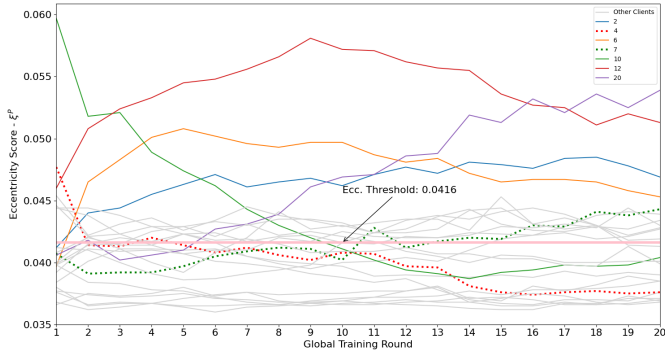


Figure 4: Clients' eccentricity signatures based on ξ^P metric calculated for each training round during the training process.

These clients' eccentricity signatures can be utilized to get deeper insights into their behavioral trends during the training. As one can notice in Fig. 3, five clients (3, 4, 5, 9, and 10) show extremely deviating behavioral patterns. Three of these clients (3, 9, and 10) have the highest influence scores according to ξ^G metric scores listed in Table 2. The same is valid for client 6, which has also demonstrated a deviating behavior as shown in Fig. 3. In general, all the clients that are eccentric according to ξ^G have high influence scores; see Table 2. Interestingly, in Fig. 4, client 10 appears with four new clients (2, 6, 12, and 20), all demonstrating highly deviating behavioral patterns. In addition, clients 10 and 12 are identified as eccentric according to ξ^P scores listed in Table 2, while both also have high influence scores. In fact, client 10 has the highest eccentric values calculated by both metrics (ξ^G and ξ^P). In both

figures, the eccentric threshold ($1/24$) is indicated in a horizontal pink color line. We can observe that in Fig. 4, client 4 exhibits eccentric behavior at the initial rounds of training and comes into a non-eccentric state in the middle, while client 7 follows the opposite trend, transforming from non-eccentric to an eccentric state. Evidently, it is important to monitor the clients' behaviors during the training process to better understand their influence on the global model.

We have also calculated the clients' average eccentricity scores over all 20 rounds for both eccentricity signatures. Interestingly, these clients deviating on ξ^G and ξ^P are eccentric according to their average eccentricity scores. In order to evaluate the influence of these clients on the global model, we have removed clients that are deviating in the ξ^P from the federation and retrained the model. The global model, trained only using selected clients, shows a reduced average MAE of 0.45, in contrast to the model trained with data from all 24 clients, which had an MAE of 0.47.

Conclusion

In this paper, we have studied how eccentricity analysis can be used to evaluate the clients' local models' behavior and its data representation in the global model. The defined eccentricity metrics have been benchmarked to the clients' benefits and influence to investigate whether there are correlations between these concepts using Kendall rankings. Initial experiments on a publicly available data set have shown these measures can provide useful information by being used separately or in combination to monitor the clients' behavior and identify clients with deviating behavioral patterns. One of the advantages of using the proposed eccentricity analysis-based approach is that it can be seamlessly integrated into the global model training process in real time without the need for a separate evaluation procedure. This means that the output of the training process will not just be the federated model but also the eccentric signatures of the clients. These signatures represent the behavior of the clients throughout the training process. We believe the proposed eccentricity analysis applies not only to horizontal FL but to any other distributed learning setting where models are shared among clients, such as transfer learning. This way, the target domain can assess the contribution of source domains, where, in this case, computation complexity would be less of a problem as this computation will then be performed only once, i.e., when the model is transferred, instead of at every training round.

Our future plans aim to test the proposed method on other publicly available decentralized data sets. We also intend to develop an evaluation technique that can determine whether clients' influence on the global model is positive or negative.

Acknowledgment

We appreciate and thank our colleagues Håkan Grahn and Emiliano Casalicchio for their support and involvement.

References

- [1] M. Malekzadeh and et al. “Honest-but-Curious Nets: Sensitive Attributes of Private Inputs Can Be Secretly Coded into the Classifiers’ Outputs.” In: *Proceedings of the 2021 ACM SIGSAC CCS*. Virtual Event, Republic of Korea, 2021, pp. 825–844. ISBN: 9781450384544. DOI: 10.1145/3460120.3484533.
- [2] B. McMahan and D. Ramage. “Federated learning: Collaborative machine learning without centralized training data.” In: *Google Research Blog* (Apr. 2017).
- [3] H. B. McMahan and et al. “Communication-Efficient Learning of Deep Networks from Decentralized Data.” In: *International Conference on Artificial Intelligence and Statistics*. 2016.
- [4] Q. Yang and et al. “Federated Machine Learning: Concept and Applications.” In: *ACM Trans. Intell. Syst. Technol.* 10.2 (2019). DOI: 10.1145/3298981.
- [5] S. Ickin and et al. “Independent Split Model Inference at Operator Network for Network Performance Estimation.” In: *IEEE/IFIP Network Operations and Management Symposium*. 2023, pp. 1–9. DOI: 10.1109/NOMS56928.2023.10154389.
- [6] G. Wang and et al. “Measure Contribution of Participants in Federated Learning.” In: *2019 IEEE International Conference on Big Data (Big Data)*. 2019, pp. 2597–2604. DOI: 10.1109/BigData47090.2019.9006179.
- [7] P. Angelov. “Anomaly detection based on eccentricity analysis.” In: *2014 IEEE Symposium on Evolving and Autonomous Learning Systems (EALS)*. 2014, pp. 1–8. DOI: 10.1109/EALS.2014.7009497.
- [8] G. K. Nilsen and et al. *Efficient Computation of Hessian Matrices in Tensor-Flow*. 2021. arXiv: 1905.05559 [cs.LG].
- [9] J. Stewart. *Calculus : early transcendentals*. Sixth. Belmont, CA: Thomson Brooks/Cole, 2008. ISBN: 9780495011668 0495011665 9780495383437 0495383430 9780495382737 0495382736.
- [10] M. G. Kendall and J. D. Gibbons. *Rank correlation methods*. Ed. by E. A. (Ed.) A Division of Hodder & Sloughton (5th ed.), London: Oxford University Press, 1990.

- [11] R. N. Forthofer and R. G. Lehen. “Rank Correlation Methods.” In: *Public Program Analysis: A New Categorical Data Approach*. Boston, MA: Springer US, 1981, pp. 146–163. ISBN: 978-1-4684-6683-6. DOI: 10.1007/978-1-4684-6683-6_9.
- [12] D. Y. Zhang and et al. “FairFL: A Fair Federated Learning Approach to Reducing Demographic Bias in Privacy-Sensitive Classification Models.” In: *2020 IEEE International Conference on Big Data (Big Data)*. 2020, pp. 1051–1060. DOI: 10.1109/BigData50022.2020.9378043.
- [13] M. Mohri and et al. *Agnostic Federated Learning*. 2019. arXiv: 1902.00146 [cs.LG].
- [14] K. Gupta et al. *Quantization Robust Federated Learning for Efficient Inference on Heterogeneous Devices*. 2022. arXiv: 2206.10844 [cs.LG].
- [15] T. Li and et al. *Ditto: Fair and Robust Federated Learning Through Personalization*. 2021. arXiv: 2012.04221 [cs.LG].
- [16] T. Li et al. *Federated Optimization in Heterogeneous Networks*. 2020. arXiv: 1812.06127 [cs.LG].
- [17] Harshvardhan and et al. *An Improved Algorithm for Clustered Federated Learning*. 2022. arXiv: 2210.11538 [stat.ML].
- [18] Y. Wang, G.-L. Li, and K.-Y. Li. “Survey on Contribution Evaluation for Federated Learning.” Chinese. In: *Ruan Jian Xue Bao/Journal of Software* 34.3 (2023), pp. 1168–1192. ISSN: 1000-9825. DOI: 10.13328/j.cnki.jos.006786.
- [19] S. K. Shyn, D. Kim, and K. Kim. *FedCCEA: A Practical Approach of Client Contribution Evaluation for Federated Learning*. arXiv:2106.02310 [cs]. June 2021. DOI: 10.48550/arXiv.2106.02310. URL: <http://arxiv.org/abs/2106.02310> (visited on 01/18/2024).
- [20] T. Wang, J. Rausch, C. Zhang, R. Jia, and D. Song. *A Principled Approach to Data Valuation for Federated Learning*. arXiv:2009.06192 [cs, stat]. Sept. 2020. DOI: 10.48550/arXiv.2009.06192. URL: <http://arxiv.org/abs/2009.06192> (visited on 01/26/2024).
- [21] Z. Liu, Y. Chen, Y. Zhao, H. Yu, Y. Liu, R. Bao, J. Jiang, Z. Nie, Q. Xu, and Q. Yang. “Contribution-Aware Federated Learning for Smart Healthcare.” English. In: vol. 36. 2022, pp. 12396–12404. ISBN: 978-1-57735-876-3.
- [22] B. Yan, B. Liu, L. Wang, Y. Zhou, Z. Liang, M. Liu, and C.-Z. Xu. “FedCM: A Real-time Contribution Measurement Method for Participants in Federated Learning.” en. In: *2021 International Joint Conference on Neural Networks (IJCNN)*. Shenzhen, China: IEEE, July 2021, pp. 1–8. ISBN: 978-1-66543-900-8. DOI: 10.1109/IJCNN52387.2021.9534451. URL: <https://ieeexplore.ieee.org/document/9534451/> (visited on 11/21/2023).

- [23] X. Hu, C. Luo, D. Zeng, Z. Xu, P. Guo, and I. King. “Flexible Contribution Estimation Methods for Horizontal Federated Learning.” In: *2023 International Joint Conference on Neural Networks (IJCNN)*. ISSN: 2161-4407. June 2023, pp. 1–8. DOI: 10.1109/IJCNN54540.2023.10191625. URL: <https://ieeexplore.ieee.org/document/10191625> (visited on 01/18/2024).
- [24] S. Li, Y. Cheng, Y. Liu, W. Wang, and T. Chen. *Abnormal Client Behavior Detection in Federated Learning*. arXiv:1910.09933 [cs, stat]. Dec. 2019. URL: <http://arxiv.org/abs/1910.09933> (visited on 11/20/2023).
- [25] L. Meng, Y. Wei, R. Pan, S. Zhou, J. Zhang, and W. Chen. “VADAF: Visualization for Abnormal Client Detection and Analysis in Federated Learning.” In: *ACM Transactions on Interactive Intelligent Systems* 11.3-4 (Sept. 2021), 26:1–26:23. ISSN: 2160-6455. DOI: 10.1145/3426866. URL: <https://dl.acm.org/doi/10.1145/3426866> (visited on 01/18/2024).
- [26] Q. Li, X. Wei, H. Lin, Y. Liu, T. Chen, and X. Ma. “Inspecting the Running Process of Horizontal Federated Learning via Visual Analytics.” In: *IEEE Transactions on Visualization and Computer Graphics* 28.12 (Dec. 2022). Conference Name: IEEE Transactions on Visualization and Computer Graphics, pp. 4085–4100. ISSN: 1941-0506. DOI: 10.1109/TVCG.2021.3074010. URL: <https://ieeexplore.ieee.org/document/9408377> (visited on 01/18/2024).
- [27] F. Malandrino and C. F. Chiasserini. “Toward Node Liability in Federated Learning: Computational Cost and Network Overhead.” In: *IEEE Communications Magazine* 59.9 (2021), pp. 72–77. DOI: 10.1109/MCOM.011.2100231.
- [28] X. Lan et al. “Federated Learning for Performance Prediction in Multi-Operator Environments.” In: *ITU Journal on Future and Evolving Technologies* 4.1 (2023).
- [29] C. G. Bezerra et al. “An evolving approach to data streams clustering based on typicality and eccentricity data analytics.” In: *Information Sciences* 518 (2020), pp. 13–28. ISSN: 0020-0255. DOI: <https://doi.org/10.1016/j.ins.2019.12.022>. URL: <https://www.sciencedirect.com/science/article/pii/S0020025519311363>.

Paper II

FedABoost: Fairness Aware Federated Learning with Adaptive Boosting

Tharuka Kasthuri Arachchige, Veselka Boeva, Shahrooz Abghari

In: Machine Learning and Principles and Practice of Knowledge Discovery in Databases, Springer Nature, Cham, Switzerland, 2026, pp. 1–16. Presented at the 3rd Workshop on Advancements in Federated Learning, co-located with ECML-PKDD 2025.

This research was funded partly by the Knowledge Foundation, Sweden, through the Human-Centered Intelligent Realities (HINTS) Profile Project (contract 20220068).

Abstract

This work focuses on improving the performance and fairness of Federated Learning (FL) in non-IID settings by enhancing model aggregation and boosting the training of under-performing clients. We propose FEDABOOST, a novel FL framework that integrates a dynamic boosting mechanism and an adaptive gradient aggregation strategy. Inspired by the weighting mechanism of the Multiclass AdaBoost (SAMME) algorithm, our aggregation method assigns higher weights to clients with lower local error rates, thereby promoting more reliable contributions to the global model. In parallel, FEDABOOST dynamically boosts under-performing clients by adjusting the focal loss focusing parameter, emphasizing hard-to-classify examples during local training. These mechanisms work together to enhance the global model’s fairness by reducing disparities in client performance and encouraging fair participation. We have evaluated FEDABOOST on three benchmark datasets: MNIST, FEMNIST, and CIFAR10, and compared its performance with those of FEDAVG and DITTO. The results show that FEDABOOST achieves improved fairness and competitive performance. The FEDABOOST code and the experimental results are available at GitHub.

1 Introduction

Federated learning (FL) is a solution to the problem of centralized learning, which requires a large amount of data and causes privacy, security, and computational challenges. The fundamental idea of FL is decentralized learning, which does not require sending user data to a central server. As an emerging technique, FL effectively addresses the challenge of preserving data privacy by keeping data localized on devices and sharing only model updates rather than raw data to train a global model collaboratively. Although raw data never leaves the client devices, the shared model updates may still reveal information about local data distributions. Therefore, techniques such as encryption and secure aggregation are employed to protect these updates during transmission [1].

FL faces several critical challenges, particularly when working with non-IID (non-Independent and Identically Distributed) data. While traditional aggregation techniques, such as FEDAVG [2], are effective in IID settings, they often perform poorly in non-IID environments [3]. The non-IID nature of client data, which can include class imbalances or unique challenging examples for each client, results in variations in the quality of local model updates. This variability affects the global model’s ability to generalize effectively [3]. Furthermore, ensuring fairness in client participation during the FL process is a critical issue [4]. Current methods often favor clients with more powerful resources, higher data quality, or faster response times. These methods unintentionally marginalize clients with fewer resources and result in biased global models. Similarly, fairness issues can arise in sharing incentives, as this may overlook unequal contributions from clients, potentially discouraging their future participation.

To improve fairness across clients while maintaining reasonable overall performance in non-IID FL settings, we propose FEDABOOST, a dynamic boosting-based FL algorithm. The key goal of FEDABOOST is to reduce disparities in model performance across clients while keeping the average predictive accuracy high. In FEDABOOST, at each round, after receiving the global model, each client evaluates it on their local data, and this feedback is used to dynamically boost the influence of underperforming clients in subsequent rounds. This boosting is achieved by adaptively tuning the focal loss focusing parameter to emphasize hard-to-classify examples. In parallel, FEDABOOST employs a novel weighting mechanism that assigns higher aggregation weights to clients whose local performance is strong, allowing the global model to better leverage reliable updates. Together, these mechanisms promote fairness by mitigating the effects of data heterogeneity and client imbalance, without resorting to personalized models. We compared FEDABOOST against FEDAVG and DITTO using MNIST, FEMNIST, and CIFAR10 datasets. The experimental results on these datasets show that FEDABOOST achieves improved fairness and competitive performance compared to those of FEDAVG and DITTO.

2 Problem Formulation

Let k denote the total number of clients participating in the FL setup. Each client i , for $i = 1, 2, \dots, k$, owns a local dataset \mathcal{D}_i , which follows a distinct data distribution P_i , where $P_i \neq P_j$ for some $i \neq j$. Clients collaboratively train a global model \mathcal{M} without sharing raw data. The \mathcal{M} is evaluated on each client’s holdout test set (sampled from \mathcal{D}_i) using loss $\ell_i(\mathcal{M})$ and performance measure $\varphi_i(\mathcal{M})$, where φ_i , e.g., can be F1 score or accuracy. We define the average loss (II.1):

$$\bar{\ell}(\mathcal{M}) = \frac{1}{k} \sum_{i=1}^k \ell_i(\mathcal{M}). \quad (\text{II.1})$$

and the variance of performance measure (II.2):

$$\text{Var}(\varphi(\mathcal{M})) = \frac{1}{k} \sum_{i=1}^k (\varphi_i(\mathcal{M}) - \bar{\varphi}(\mathcal{M}))^2, \quad \text{where } \bar{\varphi}(\mathcal{M}) = \frac{1}{k} \sum_{j=1}^k \varphi_j(\mathcal{M}). \quad (\text{II.2})$$

A lower $\text{Var}(\varphi(\mathcal{M}))$ indicates that the model performs more uniformly across all clients. Therefore, given two models \mathcal{M} and \mathcal{M}' , if $\text{Var}(\varphi(\mathcal{M})) < \text{Var}(\varphi(\mathcal{M}'))$, then \mathcal{M} is considered to be fairer than \mathcal{M}' [5].

Our aim is to develop an improved model \mathcal{M} that significantly reduces both $\bar{\ell}(\mathcal{M})$ and $\text{Var}(\varphi(\mathcal{M}))$ compared to a given baseline \mathcal{M}' , i.e.,

$$\bar{\ell}(\mathcal{M}) < \bar{\ell}(\mathcal{M}') \quad \text{and} \quad \text{Var}(\varphi(\mathcal{M})) < \text{Var}(\varphi(\mathcal{M}')). \quad (\text{II.3})$$

3 Background

3.1 Multi-class Adaptive Boosting

Adaptive Boosting [6], known as ADABOOST, is an ensemble learning method originally developed for binary classification. It builds a strong classifier by sequentially training weak learners, increasing focus on misclassified instances in each iteration. The final prediction is a weighted vote of these weak learners.

SAMME (Stagewise Additive Modeling using a Multi-class Exponential loss function) [7], extends ADABOOST to multi-class classification problems. The SAMME algorithm retains the core principles of ADABOOST but modifies the weight update factor (α) by incorporating the number of classes to ensure proper adaptation to the multi-class setting, as shown in (II.4).

$$\alpha_l = \ln \left(\frac{1 - \mathcal{E}_l}{\mathcal{E}_l} \right) + \ln(C - 1), \quad (\text{II.4})$$

where, C represents the number of classes, and \mathcal{E}_l denotes the weighted classification error of the l th classifier, calculated as

$$\mathcal{E}_l = \sum_{i=1}^N w_i I(T_l(x_i) \neq y_i).$$

The indicator function $I(\cdot)$ equals 1 if the sample x_i is misclassified and 0 otherwise.

To ensure $\alpha_l > 0$, the weak classifier must perform better than random guessing, i.e., $(1 - \mathcal{E}_l) > 1/C$. This requirement is critical in boosting, as weak classifiers that perform worse than random chance would otherwise negatively impact the ensemble model. Additionally, SAMME combines weak classifiers slightly different from ADABOOST by incorporating $\log(C - 1)$, expressed as

$$\log(C - 1) \sum_{i=1}^N I(T(l)(x_i) = c).$$

When $C = 2$, SAMME behaves similarly to ADABOOST.

3.2 Focal Loss for Challenging Cases

Focal loss [8] was originally proposed to address the class imbalance in object detection by modifying the standard cross-entropy loss to emphasize hard-to-classify examples and reduce the influence of easy ones. The formula of focal loss is given by (II.5):

$$\mathcal{L}_{\text{Focal}}(p_t) = -\beta_t(1 - p_t)^\gamma \log(p_t), \quad (\text{II.5})$$

where p_t is the predicted probability assigned to the ground-truth class, β_t is a balancing factor (analogous to α_t in [8]), and $\gamma \geq 0$ is the focusing parameter. The modulating factor $(1 - p_t)^\gamma$ dynamically scales the loss based on the prediction confidence; when p_t is high (i.e., the prediction is correct and confident), the factor approaches zero, reducing the loss contribution from easy examples. Conversely, for hard or misclassified examples where p_t is low, the modulating factor remains near one, preserving a high loss and encouraging the model to focus on these samples. Increasing γ amplifies this effect, further prioritizing difficult examples during training.

4 Methodology

4.1 Aggregation Mechanism in FedABoost

Inspired by SAMME, we adapt the weight update factor (α), originally defined in (II.4), for the FL setup. In SAMME, weak learners (classifiers) are trained sequentially, with

each iteration adjusting the sample weights based on previous errors. However, in FL, clients train independently and in parallel [2], making sequential re-weighting infeasible.

FEDABOOST treats clients as weak learners. Instead of weighting weak classifiers, it dynamically weights client updates based on their local performance. Clients with lower error rates receive higher α values, increasing their influence on the global model. Conversely, clients with higher error rates receive lower α values, reducing their contribution and helping to mitigate the risk of noise or bias from unreliable updates. This approach improves the performance of the global model by down-weighting weak clients in the presence of non-IID data.

In each global round e , let $S_e \subseteq \{1, 2, \dots, k\}$ denote the set of participating clients, where k is the total number of clients and $|S_e| = m$. Each client j trains a simple neural network (NN), μ_j^e on its local dataset D_j . Drawing from (II.4), we define the weighting factor α for client j in the e th training round as:

$$\alpha_j^e = \ln \left(\frac{1 - \mathcal{E}_j^e}{\mathcal{E}_j^e} \right) + \ln(C_j - 1), \quad (\text{II.6})$$

where C_j denotes the number of classes handled by client j . Unlike SAMME, where α is based on the error of the weak classifier on a common dataset, here \mathcal{E}_j^e denotes the error rate of client j 's local model (μ_j^e) on its own validation set drawn from D_j . The value α_j^e is positive only when $1 - \mathcal{E}_j^e > 1/C_j$, meaning only clients whose performance exceeds random guessing are included in the aggregation. Clients with non-positive α_j^e values are ignored, as their updates are likely to degrade the global model.

Crucially, as the number of classes that a client handles increases, so does the acceptable error rate for inclusion. This design choice is beneficial for FL settings with non-IID data, as it enables clients with moderate performance to contribute positively by leveraging the ensemble effect. However, including too many weak client models can still negatively impact performance, highlighting the importance of selective weighting and client filtering.

When \mathcal{E}_j^e approaches 0 or 1, α_j^e tends toward infinity, causing excessive influence from the local model of that client. Such extreme weighting can adversely impact the aggregation process, introducing overfitting or bias into the global model. To prevent this, FEDABOOST clips \mathcal{E}_j^e to the range $[\epsilon, 1 - \epsilon]$, where ϵ is a small constant (e.g., 10^{-6}). This clipping approach ensures that no individual client disproportionately influences the global model, while preserving the core principle of FEDABOOST, which is to amplify contributions from high-performing clients while controlling the impact from low-performing ones.

Each client computes α_j^e locally and communicates it alongside its model update μ_j^e to the server. The server then aggregates the global model \mathcal{M}^{e+1} as:

$$\mathcal{M}^{e+1} = \frac{\sum_{j=1}^m \alpha_j^e \mu_j^e}{\sum_{j=1}^m \alpha_j^e}. \quad (\text{II.7})$$

4.2 FedABOOST Boosting Mechanism

We utilize a weight calculation mechanism inspired by the SAMME (Sec. 3.1) and focal loss (Sec. 3.2) to boost the training of underperforming clients in the FL setup. In SAMME, the weights for each classifier (l) at iteration “ e ” is updated using the following equation: $w_l^e = w_l^{e-1} \cdot \exp(\alpha_l \cdot I(l \text{ Performance}))$, where α_l is based on the l ’s error rate (II.4). The term I is an indicator function that returns 0 when l ’s meets a defined performance threshold.

A common challenge in ADABOOST and SAMME is the rapid increase in weights due to the exponential factor of α , which can lead to overfitting. While SAMME somewhat eases this issue by integrating the number of classes into the α calculation as shown in (II.4), it does not completely resolve it. To address this, we incorporate a boosting rate $\eta \in [0, 1]$ into the weight update mechanism [9, 10]. This adjustment mitigates the steep increase in weights during the later stages of training, ultimately resulting in a more stable and generalizable model performance.

The weights are calculated by (II.8):

$$w_j^e = w_j^{e-1} \exp(-\eta \alpha_j I(\mathcal{M} \text{ Performance on } \mathcal{D}_j)), \quad (\text{II.8})$$

The negative sign inverts the influence of α_j , ensuring that clients with higher error rates (and thus lower α_j) receive increased weights: clients with poorer performance receive higher weights, encouraging their improvement during training.

In the initial global training round, all clients are assigned equal weights, represented as $w_j = 1/m$, for $j = 1, 2, \dots, m$, where m denotes the number of clients that participate in the initial global round. After each global round, weights are updated using (II.8), with α_j derived from the error rate of client j ’s local model (II.4). $I(\mathcal{M} \text{ Performance on } \mathcal{D}_j)$ is the indicator function that equals 0 if the \mathcal{M} ’s performance meets a predetermined error threshold (τ), which should be established empirically. This means that once the model achieves the specified performance for client j , the algorithm will no longer boost client j ’s model training.

Subsequently, FEDABOOST utilizes these weights to adjust the γ values in the focal loss (II.5) function. At each global round, γ is incrementally increased by the updated weight, enabling the model to emphasize harder samples during training. The γ value is constrained to the range $[0, 5]$ as suggested in [8]. As we assume static data across rounds, we do not update the class imbalance parameter β_t ; it is set empirically and remains fixed. Finally, weight updates are computed only for clients that actively participate in each global training round.

4.3 The proposed FedABoost algorithm

The FEDABOOST algorithm consists of two primary phases: *Initialization* and *Iteration*. In the *Initialization* phase, FEDABOOST initializes the global model, denoted as \mathcal{M}^e at iteration $e = 0$ with random parameters. Since each client model serves as a weak learner in FEDABOOST, a simple NN architecture is particularly well-suited for a global model. The client weights are also set to be equal at the initial round, with each client i is assigned an initial weight $w_i = 1/m$, where $i = 1, 2, \dots, k$ and m is the number of clients participate in the initial training round and k is the total clients in the federated setup.

Algorithm 2 FEDABOOST.

```

1: procedure SERVER-SIDE
2:   Initialize  $\mathcal{M}^1$  and weights  $w_i = 1/k$  for  $i = 1, \dots, k$ 
3:   for  $e = 1, 2, \dots, E$  do
4:     Set  $S_e \subseteq \{1, 2, \dots, k\}$ ,  $|S_e| = m$ 
5:     for each client  $j \in S_e$  in parallel do
6:        $\mu_j^e, \alpha_j^e \leftarrow \text{CLIENT-UPDATE}(j, \mathcal{M}^e)$ 
7:     end for
8:     Aggregate  $\{\mu_j^e, \alpha_j^e\}$  to compute  $\mathcal{M}^{e+1}$  via (II.7)
9:   end for
10: end procedure
11: procedure CLIENT-UPDATE(Client  $j$ , Global Model  $\mathcal{M}$ )
12:   Compute  $\mathcal{E}_j$  (error rate of  $\mathcal{M}$ ), then calculate  $\alpha_j$  using (II.4), and  $w_j$  using (II.8)
13:    $\mu_j \leftarrow$  Train  $\mathcal{M}$  on  $\mathcal{D}_j$  per local round, with loss influenced by  $w_j$ 
14:   Recompute  $\mathcal{E}_j$  and update  $\alpha_j$ 
15:   return  $\mu_j, \alpha_j$ 
16: end procedure

```

In the *Iteration* phase, at each global round e , the weights $\{w_i^e\}_{i=1}^k$ and the global model \mathcal{M}^e are shared by a subset of clients $S_e \subseteq \{1, 2, \dots, k\}$, where $|S_e| = m$, who participate in the next training round ($e + 1$). Upon receiving \mathcal{M}^e , client j , for $j = 1, 2, \dots, m$, computes the error rate (\mathcal{E}_j^e) of \mathcal{M}^e for its local data (\mathcal{D}_j). Client j then calculates α_j^e using (II.6) and updates the weight (w_j^e) using (II.8). Later, the client j , trains \mathcal{M}^e several local iterations with \mathcal{D}_j leading to μ_j^e . During the training process, the client's weight w_j^e boosts the training, as explained in Section 4.2. After completing the local training, the new error rate \mathcal{E}_j^e and new α_j^e are computed for the model μ_j^e , and both α_j^e and the trained local model μ_j^e are sent to the central server. Upon receiving updates from all the clients, a new global model is formed using (II.7). The global model \mathcal{M}^{e+1} is then sent back to all clients. The iteration process continues until the global model converges, as described in Algorithm 2.

4.4 FedABOOST Fairness and Convergence

The FEDABOOST achieves **convergence towards fairness and performance** (Sec. 2) by integrating two complementary mechanisms: performance-aware aggregation (Sec. 4.1) and an adaptive, client-specific boosting mechanism (Sec. 4.2). These two processes work in tandem: high-performing clients guide the global model aggregation (II.7), while underperforming clients receive boosted local training through adaptive focal loss, leading to a gradual reduction in inter-client performance disparity. This combination ensures that at across global round $e = 1, 2, \dots, E$, the global model loss $\ell(\mathcal{M}^e)$ decreases while the variance in client performance $\text{Var}(\varphi(\mathcal{M}^e))$ diminishes.

We adopt the following standard assumptions to establish convergence of FEDABOOST: [11]

- (A1) Each local objective $\ell_j(\cdot)$ is L -smooth:

$$\|\nabla\ell_j(\mathcal{M}^e) - \nabla\ell_j(\mu_j^e)\| \leq L\|\mathcal{M}^e - \mu_j^e\|, \quad \forall j, e.$$

ensuring the loss surface is smooth, bounding gradient variation between the global and local models.

- (A2) Since client j performs a finite number of deterministic local epochs, the deviation between its local and global model remains bounded:

$$\|\mu_j^e - \mathcal{M}^e\| \leq \Delta, \quad \forall j, e.$$

guaranteeing the stability of local updates.

- (A3) The performance-aware aggregation weights α_j^e from (II.6) are clipped to a bounded range:

$$0 < \underline{\alpha} \leq \alpha_j^e \leq \bar{\alpha} < \infty,$$

preventing any single client from dominating the global update. During aggregation (Eq. (II.7)), the weights are automatically normalized by their sum across the participating clients m at round e .

- (A4) After local optimization with the focal loss (focusing parameter $\gamma_j^e \in [0, 5]$), client j achieves a contraction of its gradient norm relative to the received global model:

$$\|\nabla\ell_j(\mu_j^e)\| \leq \rho_j^e \|\nabla\ell_j(\mathcal{M}^e)\|, \quad 0 \leq \rho_j^e \leq \rho < 1.$$

The contraction coefficient ρ_j^e reflects the efficiency of the boosted local update. Note that larger focusing parameters γ_j^e typically yield smaller ρ_j^e , indicating stronger local progress.

Under (A1)–(A4), the smoothness and bounded update assumptions ensure that the global objective $\bar{\ell}(\mathcal{M}^e)$ decreases monotonically across rounds, and \mathcal{M}^e converges to a stationary point of $\bar{\ell}$. Since underperforming clients receive stronger local updates through adaptive focal boosting (larger focusing parameter γ_j^e leading to smaller ρ_j^e in assumption (A4)), their local performance improves more rapidly in subsequent rounds. As the aggregation weights α_j^e are recomputed based on updated client (μ_j^e) performance, this mechanism gradually equalizes contributions across clients. Consequently, the inter-client performance variance contracts in expectation:

$$\mathbb{E}\left[\text{Var}(\varphi(\mathcal{M}^{e+1}))\right] \leq (1 - \rho_f) \text{Var}(\varphi(\mathcal{M}^e)), \quad \rho_f > 0.$$

A larger η accelerates convergence (higher ρ) but may induce oscillations, while a smaller η ensures more stable progress. Under these deterministic assumptions ((A1)–(A4)), FEDABOOST guarantees monotonic reduction of both the global objective (II.1) and the inter-client performance variance (II.2), achieving convergence toward fairness and overall performance.

5 Experimental Setup

The FEDABOOST algorithm is evaluated on three datasets: **MNIST**, **FEMNIST**, and **CIFAR10**¹. For both MNIST and CIFAR10, we simulate non-IID conditions by employing Dirichlet data partitioning [12]. Specifically, we allocated data to 264 clients for MNIST and to 196 clients for CIFAR10, sampling class proportions for each client from a Dirichlet distribution with a concentration parameter of 0.2 for MNIST and 0.4 for CIFAR10. Conversely, FEMNIST is inherently non-IID, containing user-annotated handwriting data distributed across individual writers.

We compare FEDABOOST with two FL baselines: FEDAVG [2] and DITTO [5]. FEDAVG aggregates local model updates by weighting them based on dataset size. In contrast, DITTO maintains both global and personalized local models, incorporating a regularization term (λ) that controls closeness between them.

Three experiments were conducted to study the performance of FEDABOOST under different scenarios. In the first experiment (**Ex.1**), we evaluated FEDABOOST on the MNIST dataset in comparison with FEDAVG. Each communication round involved randomly selected 30% of clients. To analyze the contribution of different components, we performed an ablation study using a variant of FEDABOOST that utilizes only the alpha-based aggregation and excludes the boosting mechanism. All models were optimized using stochastic gradient descent (SGD) with shared hyperparameters, which were first tuned using FEDAVG and reused for all other algorithms to ensure a

¹Dataset links: MNIST: <http://yann.lecun.com/exdb/mnist>, FEMNIST: <https://leaf.cmu.edu>, CIFAR-10: <https://www.cs.toronto.edu/~kriz/cifar.html>

controlled comparison. The local model architecture is a fully connected NN with one hidden layer.

In the second experiment (**Ex.2**), we used the FEMNIST dataset to compare FEDABOOST with both FEDAVG and DITTO. We randomly selected 20% of clients per round and repeated the experiment three times with different client selections to ensure statistical robustness. The model architecture was a lightweight convolutional NN, consisting of a single convolutional layer with batch normalization and max pooling, followed by dropout and a fully connected output layer. For FEDAVG, we empirically tuned the hyperparameters using SGD. These settings were reused for FEDABOOST to ensure a fair and controlled comparison. We also evaluated an alternative configuration of FEDABOOST using the AdamW optimizer. DITTO was trained using the same global model architecture and SGD optimizer as FEDAVG, while the personalized models maintained by each client were updated locally using the Adam optimizer.

In the third experiment (**Ex.3**), we evaluated FEDABOOST on the CIFAR10 dataset in comparison with FEDAVG and DITTO. This experiment was designed to investigate the impact of varying client participation rates on model performance. We conducted training runs using three different client participation fractions: 20%, 40%, and 60%, with clients randomly selected in each communication round. To ensure a fair and controlled comparison, we reused the SGD-tuned hyperparameters originally optimized for FEDAVG across all methods. DITTO was configured consistently with **Ex.2**. The global model was trained using the same settings as FEDAVG, while the personalized models maintained by each client were updated locally using the Adam optimizer. The local model architecture was a convolutional NN consisting of two convolutional layers with group normalization, followed by max pooling, dropout, and two fully connected layers. This setup allowed us to assess how different levels of client availability influence the relative effectiveness of FEDABOOST.

In all three experiments, we evaluated the global model at each training round using the loss and $F1$ score calculated as the macro-average across all classes on a global validation set comprising 20% of unseen data from each client. In all methods, the parameter β_t in (II.5), which controls the degree of focusing on class imbalance, was fixed at 1 across all experiments for consistency.

FEDABOOST introduces two key hyper parameters: the boosting rate (η) and the error threshold (τ) (II.8). These parameters should be selected based on the nature and complexity of the classification task, as well as the desired strength of the boosting effect. The boosting rate, η , controls the impact of boosting in each round. In practice, it is recommended to start with a relatively large value (e.g., $\eta = 0.1$) and gradually decrease it to 0.01 or 0.001 as training stabilizes. This annealing strategy prevents over-adjustment while maintaining the ensemble’s adaptability. The error threshold, τ , defines the performance cutoff used in the indicator function (II.8). Once a client’s local model achieves the specified performance level, boosting for that client ceases. Hence, a higher τ is suitable for simpler tasks with fewer classes, whereas a lower τ encourages stronger boosting for more complex tasks. For instance, we set $\tau = 0.7$ for

MNIST (10 classes), $\tau = 0.5$ for FEMNIST (62 classes), and $\tau = 0.4$ for CIFAR-10 (10 classes), reflecting the relative task difficulties of these datasets.

6 Evaluation and Results

The results of **Ex.1** are presented in Figure 1. We observe that FEDABOOST consistently outperforms FEDAVG. Specifically, FEDABOOST achieves an $F1$ score of approximately 0.88 in convergence, while FEDAVG saturates around 0.87. The comparison between FEDABOOST and its ablated variant (without boosting) clearly demonstrates the critical role of boosting in enhancing model performance. This indicates that boosting contributes to addressing non-IID data challenges and client heterogeneity in FL. Another critical observation is the communication efficiency demonstrated by FEDABOOST, as it consistently reaches higher $F1$ scores in fewer communication rounds compared to FEDAVG, indicating a reduction in communication overhead for a given performance target. This is particularly valuable in FL environments, where the communication cost is a bottleneck.

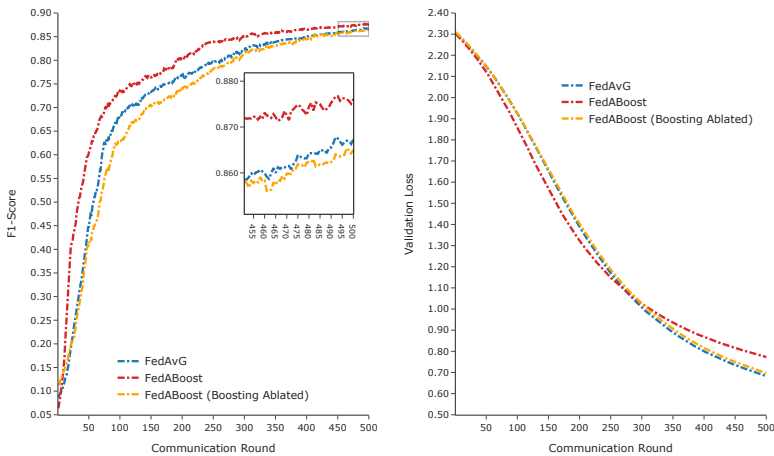


Figure 1: Ex.1: MNIST. All models are trained with SGD (learning rate = 1×10^{-3} , batch size = 32, weight decay = 1×10^{-3} , local epochs = 5); FedABoost $\eta = 0.01$ and $\tau = 0.3$. Total clients: 264.

It is also noted that the validation loss curves exhibit a temporary crossing around communication round 250, likely due to the dynamic adjustment of the focal loss γ parameter in FEDABOOST. As γ increases, the model prioritizes harder samples, which eventually increases loss without significantly impacting the $F1$ score, as prediction accuracy remains stable.

The results of **Ex.2** are shown in Figure 2. FEDABOOST-1 consistently achieves higher $F1$ scores across communication rounds when compared to FEDAVG. This highlights the effectiveness of its dynamic boosting and aggregation mechanisms.

However, FEDABOOST-1’s dependence limits optimization efficiency due to sensitivity to learning rate and lack of adaptivity of SGD. In contrast, FEDABOOST-2, using AdamW, significantly outperforms all baselines, achieving faster and more stable convergence. This gain is likely from AdamW’s adaptive learning rate and decoupled weight decay, which better accommodate the variability introduced by FEDABOOST’s α -based aggregation and dynamic γ adjustment in focal loss. We have also experimented with both optimizers across algorithms and found that SGD was more stable with FEDAVG, while AdamW worked better with FEDABOOST, resulting in a more stable learning curve.

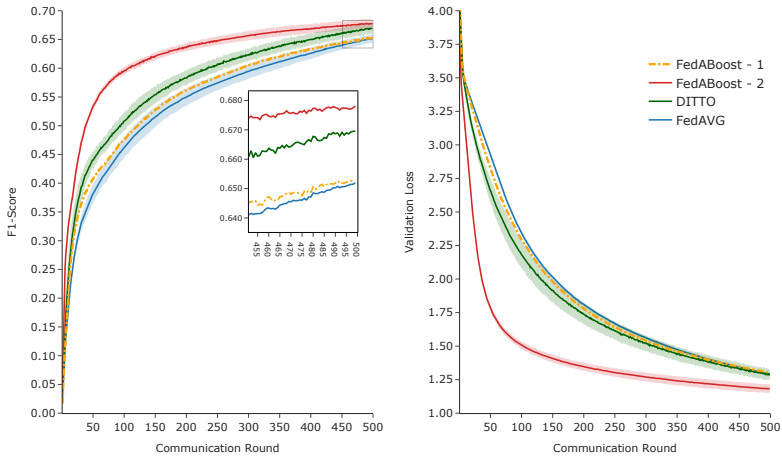


Figure 2: Ex.2: FEMNIST. All models are trained for 5 local epochs. The global models of FedAvg, FedABoost-1, and Ditto use SGD (learning rate = 10^{-3} , batch size = 64, weight decay = 5×10^{-4}). Ditto’s personalized models use Adam (learning rate = 10^{-3} , batch size = 64, weight decay = 5×10^{-4} , $\lambda = 0.1$). FedABoost-2 uses AdamW (learning rate = 2×10^{-4} , batch size = 64, weight decay = 10^{-6}). Both FedABoost versions use $\eta = 0.01$ and $\tau = 0.5$. Total clients: 3,550.

DITTO trained its global model using SGD, like FEDAVG, while maintaining personalized models for each client using Adam. We can see that DITTO consistently outperforms FEDABOOST-1. However, it does not surpass FEDABOOST-2, which leverages AdamW in a fully collaborative setting with dynamic client weighting and loss modulation. This suggests that while DITTO benefits from personalization through local adaptivity, it lacks the collective optimization enhancements introduced by FEDABOOST’s α -based aggregation and dynamic γ adjustment. Moreover, DITTO requires additional memory due to dual model maintenance, which can pose challenges for some clients, especially those with limited resources.

In Ex.1 and Ex.2, the distribution of $F1$ scores across all clients is presented in Figure 3. In both MNIST (left) and FEMNIST (right), FEDABOOST produces a right-shifted, more concentrated distribution compared to baselines, indicating improved performance and consistency. It achieves the highest median $F1$ scores in both cases; 0.847 on MNIST (versus 0.820 for FEDAVG) and 0.642 on FEMNIST (versus 0.590 for DITTO and 0.558 for FEDAVG).

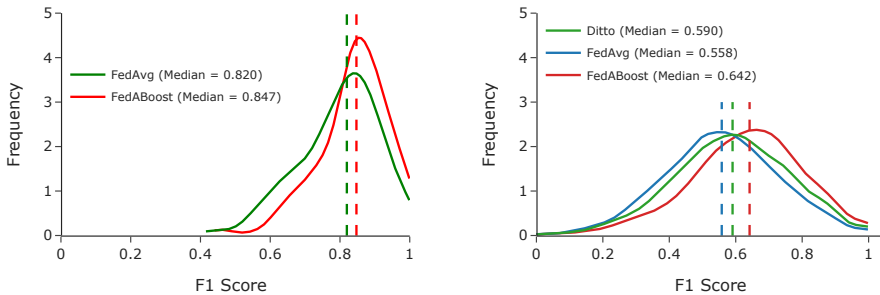


Figure 3: F_1 score distributions: Left – MNIST; Right – FEMNIST.

To quantify fairness improvements, we calculated the variance of macro F_1 scores across clients over the convergence windows (global rounds 245 to 255 for MNIST, and global rounds 205 to 210 for FEMNIST). The corresponding variances and 95% confidence intervals (CIs) are reported in Table 1. On MNIST, FEDABOOST achieves a lower variance of 0.0103, reducing variance by 24.4% compared to FEDAVG (0.0137). On FEMNIST, FEDABOOST achieves an average variance of 0.0279, representing a 5.88% reduction compared to FEDAVG (0.0296) and an 11.87% reduction over DITTO (0.0317). These results, supported by non-overlapping CIs, confirm that FEDABOOST not only enhances the performance but also improves fairness across clients, fulfilling the objective defined in (II.3).

Table 1: Fairness comparison across clients on **Ex.1** and **Ex.2** based on the variance of F_1 scores (lower is better). Results are reported with 95% confidence intervals (CI). The best results are in bold.

Dataset	Algorithm	Variance (95% CI)
MNIST	FedAvg	0.0137 [0.0135, 0.0138]
	FedABoost	0.0103 [0.0102, 0.0104]
FEMNIST	FedAvg	0.0296 [0.0295, 0.0297]
	Ditto	0.0317 [0.0313, 0.0320]
	FedABoost	0.0279 [0.0278, 0.0280]

The results of **Ex.3**, shown in Table 2, indicate that FEDABOOST outperforms both FEDAVG and DITTO across all client fractions in terms of F_1 score, with the most notable margin at 20% participation (0.716 versus 0.694 and 0.689). As can be observed, the performance gap is not as significant as in **Ex. 1** and **Ex. 2**, where the data were highly non-IID. Note that the CIFAR10 dataset client fraction was performed with a concentration parameter of 0.4, resulting in moderately non-IID data. FEDABOOST also shows lower validation loss at lower participation levels, indicating better generalization under constrained settings. This aligns with the design of FEDABOOST, which enhances underrepresented clients. This effect is most beneficial when fewer clients participate per round, allowing their influence to be more effectively integrated into the global model.

Table 2: Ex.3: CIFAR10. All global models of FedAvg, FedABoost, and Ditto are trained for 10 local epochs using SGD (learning rate = 10^{-2} ; batch size = 32). Ditto’s personalized models are trained with SGD (learning rate = 10^{-3} , batch size = 32, $\lambda = 0.1$). FedABoost uses $\eta = 0.002$, $\tau = 0.4$. The global model converged in approximately 60–70 rounds with 20% of the data, 35–45 rounds with 40%, and 25–30 rounds with 60%. Total clients: 196.

Algorithm	F1 Score			Validation Loss		
	20%	40%	60%	20%	40%	60%
FedAvg	0.694	0.679	0.675	1.198	1.149	1.156
Ditto	0.689	0.672	0.642	1.199	1.208	1.355
FedABoost	0.716	0.685	0.677	0.992	1.051	1.078

Our experiments have revealed that FEDABOOST is more efficient in non-IID settings than in IID, compared to the two baselines. Furthermore, FEDABOOST has demonstrated more stable behavior during federated training than FEDAVG, achieving a faster loss reduction.

7 Literature Review

In this section, we review the literature on two topics related to our work: the aggregation techniques used to build federated models and fairness in FL.

7.1 Model Aggregation in FL

A comprehensive survey of existing model aggregation techniques in FL is presented in [13], which classifies them into two categories: *Parameter-based aggregation* and *Output-based aggregation*. This classification is determined by the nature of the aggregated objects. Parameter-based aggregation [14] combines trainable parameters or gradients from local models, whereas output-based methods aggregate model representations such as logits or compressed sketches. An example of output-based aggregation is FEDMASK [15]. This enables each device to learn a binary mask, which the server aggregates to improve training efficiency under limited computational resources. ZENO [16] is an example of a parameter-based aggregation model for SGD ranking that utilizes a stochastic zero-order oracle. This ranking reflects the trustworthiness of clients in each iteration. Considering the average score among the candidates with the highest rankings allows ZENO to accommodate a significant number of incorrect gradients. This method ensures convergence at a rate similar to fault-free SGD, while the variance decreases as the number of non-faulty workers increases. To further improve robustness against model poisoning, the work in [17] introduces three adaptive aggregation functions—*Switch*, *Layered-Switch*, and *Weighted FedAvg* that dynamically switch based on training conditions.

7.2 Fairness in FL

Huang et al. [18] discuss two types of unfairness in the context of FL: *reward conflict* and *prediction biases*. This discussion results in two main methodological objectives: *collaboration fairness* and *performance fairness*. Collaboration fairness aims to provide greater rewards or incentives to participants with greater contributions, while performance fairness focuses on achieving equal accuracy distribution across participants.

Collaboration Fairness: CFFL [19] and RFFL [20] promote collaborative fairness through a reward mechanism that evaluates client contributions and iteratively adjusts rewards based on gradient updates. Qiuxian et al. [21] propose a fairness mechanism using rewards for improvements in clients' model performance and penalties for deviations from the global model. FEDAVE [14] uses an adaptive reputation calculation module to evaluate clients' reputations based on their local model performance and data similarity to a validation set. A dynamic gradient reward distribution module then allocates rewards based on these reputations, ensuring that more valuable contributions receive larger rewards. Wang et al. [22] discuss the disadvantages of approaches that achieve fairness by adjusting clients' gradients [14, 19, 20], noting that these methods often fail to maintain consistency across local models and do not adequately address the needs of high-contributing clients. To tackle this issue, the authors propose FEDSAC, which dynamically allocates sub-models to each client based on their contributions, rewarding those who contribute more to the learning process with higher-performing sub-models.

Performance Fairness: Zhang et al. [23] identify three key challenges related to fairness in FL, namely the trade-offs between fairness and performance, limited information availability, and constrained coordination. To address these challenges, they propose the FAIRFL, which uses deep multi-agent reinforcement learning alongside a secure information aggregation protocol to optimize both accuracy and fairness while ensuring privacy. FAIRFED [24] improves fairness in FL by adjusting the model aggregation weights based on local fairness, which assesses the model performance across different demographic groups within a client's dataset. DITTO [5] achieves fairness by creating personalized models that combine a global model and local objectives for each client, which helps address the variability in data across clients. FEDFAIM [25] is a non-monetary incentive mechanism for FL based on two principles: aggregation fairness and reward fairness. Aggregation fairness involves using an efficient gradient aggregation method that evaluates the quality of local gradients and combines them appropriately. In contrast, reward fairness is achieved through a Shapley value-based approach, which determines the contributions of different participants.

FEDABOOST aligns with the latter objective, adopting a parameter-based aggregation strategy. Unlike collaboration fairness approaches that rely on explicit reward or incentive mechanisms, FEDABOOST promotes fairness by encouraging underperforming clients to contribute more effectively. This design fosters balanced model

performance across clients without the need for additional incentive or reputation modules. Therefore, our work primarily addresses performance fairness in FL.

8 Conclusion and Future Directions

In this work, we have introduced FEDABOOST, a novel FL framework that enhances global model quality by fairly boosting clients based on local performance. Evaluations on MNIST, FEMNIST, and CIFAR10 data demonstrate that FEDABOOST generally outperforms FEDAVG and DITTO, particularly in non-IID settings and those with limited client participation. Furthermore, results suggest that FEDABOOST is particularly well suited to cross-silo FL settings, where fairness and interpretability of client contributions are critical.

Despite its promise, FEDABOOST shows sensitivity to its hyperparameters and can be fragile in certain settings. In our future work, therefore, we plan to explore alternative boosting strategies beyond focal loss, and incorporate adaptive mechanisms such as error-threshold (τ) and boosting rate (η) scheduling, and robust optimizer tuning to further improve stability and generalization across diverse scenarios.

References

- [1] H. Kaur et al. “Federated learning: a comprehensive review of recent advances and applications.” In: *Multimedia Tools and Applications* (2023), pp. 1–24.
- [2] B. McMahan et al. “Communication-efficient learning of deep networks from decentralized data.” In: *AI and Statistics*. PMLR. 2017, pp. 1273–1282.
- [3] X. Li et al. “On the convergence of fedavg on non-iid data.” In: *arXiv preprint arXiv:1907.02189* (2019).
- [4] Y. Shi, H. Yu, and C. Leung. “Towards Fairness-Aware Federated Learning.” In: *IEEE Transactions on NN and Learning Sys.* 35.9 (2024), pp. 11922–11938.
- [5] T. Li et al. “Ditto: Fair and robust federated learning through personalization.” In: *Int. Conference on ML*. PMLR. 2021, pp. 6357–6368.
- [6] Y. Freund and R. E. Schapire. “A decision-theoretic generalization of on-line learning and an application to boosting.” In: *J. of Comp. and Syst. Scienc.* 55.1 (1997), pp. 119–139.
- [7] T. Hastie, S. Rosset, J. Zhu, and H. Zou. “Multi-class adaboost.” In: *Statistics and its Interface 2.3* (2009), pp. 349–360.
- [8] T.-Y. Lin et al. “Focal loss for dense object detection.” In: *Proc. of the IEEE Int. Conference on Computer Vision*. 2017, pp. 2980–2988.

- [9] T. Hastie, R. Tibshirani, and J. Friedman. *Boosting and Additive Trees*. 2nd ed. New York: Springer, 2009. Chap. 10, pp. 337–387.
- [10] Scikit-learn. *Multi-class AdaBoosted Decision Trees*. URL: https://scikit-learn.org/stable/auto%5C_examples/ensemble/plot%5C_adaboost%5C_multiclass.html. (accessed: 2024-10-17).
- [11] T. Li et al. *Federated Optimization in Heterogeneous Networks*. 2020. arXiv: 1812.06127 [cs.LG].
- [12] T. Lin et al. “Ensemble distillation for robust model fusion in federated learning.” In: *Advances in neural information processing systems* 33 (2020), pp. 2351–2363.
- [13] P. Qi et al. “Model aggregation techniques in federated learning: A comprehensive survey.” In: *Future Generation Computer Systems* 150 (2024), pp. 272–293.
- [14] Z. Wang et al. “FedAVE: Adaptive data value evaluation framework for collaborative fairness in federated learning.” In: *Neurocomputing* 574 (2024), p. 127227.
- [15] A. Li, J. Sun, X. Zeng, M. Zhang, H. Li, and Y. Chen. “Fedmask: Joint computation and communication-efficient personalized federated learning via heterogeneous masking.” In: *Proceedings of the 19th ACM conference on embedded networked sensor systems*. 2021, pp. 42–55.
- [16] C. Xie, S. Koyejo, and I. Gupta. “Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance.” In: *International conference on machine learning*. PMLR. 2019, pp. 6893–6901.
- [17] S. Nabavirazavi et al. “Enhancing federated learning robustness through randomization and mixture.” In: *Future Gener. Comp. Syst.* 158 (2024), pp. 28–43.
- [18] W. Huang et al. “Federated learning for generalization, robustness, fairness: A survey and benchmark.” In: *IEEE Trans. on Pattern Analysis and Machine Intell.* (2024).
- [19] L. Lyu et al. “Collaborative fairness in federated learning.” In: *Federated Learning: Privacy and Incentive* (2020), pp. 189–204.
- [20] X. Xu and L. Lyu. “A reputation mechanism is all you need: Collaborative fairness and adversarial robustness in FL.” In: *arXiv preprint arXiv:2011.10464* (2020).
- [21] L. Qiuxian et al. “A Secure and Fair Federated Learning Protocol Under the Universal Composability Framework.” In: *Int. Conf. on Multimedia Modeling*. Springer. 2024, pp. 462–474.

- [22] Z. Wang et al. “FedSAC: Dynamic Submodel Allocation for Collaborative Fairness in Federated Learning.” In: *Proc. of the 30th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining*. 2024, pp. 3299–3310.
- [23] D. Y. Zhang, Z. Kou, and D. Wang. “Fairfl: A fair federated learning approach to reducing demographic bias in privacy-sensitive classification models.” In: *IEEE Int. Conference on Big Data*. 2020, pp. 1051–1060.
- [24] Y. H. Ezzeldin et al. “Fairfed: Enabling group fairness in federated learning.” In: *Proc. of the AAAI Conference on AI*. Vol. 37. 6. 2023, pp. 7494–7502.
- [25] Z. Shi, L. Zhang, Z. Yao, L. Lyu, C. Chen, L. Wang, J. Wang, and X.-Y. Li. “Fedfaim: A model performance-based fair incentive mechanism for federated learning.” In: *IEEE Transactions on Big Data* 10.6 (2022), pp. 1038–1050.

Paper III

Hierarchical Knowledge Distillation for Fair Federated Learning

Tharuka Kasthuri Arachchige, Veselka Boeva, Shahrooz Abghari

This is the authors' pre-printed version of the work submitted for conference. Work under evaluation. It is posted here for personal use. Not for redistribution.

This research was funded partly by the Knowledge Foundation, Sweden, through the Human-Centered Intelligent Realities (HINTS) Profile Project (contract 20220068).

Abstract

Federated Learning (FL) on heterogeneous data often leads to performance disparities across clients, where improvements in average accuracy may mask degradation for disadvantaged clients. Most existing FL frameworks treat all clients as homogeneous participants, ignoring latent similarities in their data distributions. In this study, we introduce a fairness-aware FL framework, DEFFT, which is specifically designed to enhance both global performance and fairness among clients. DEFFT identifies the distributional structure among clients based on label distributions, creating persistent client groups that reflect data similarities. For each of these groups, a cluster-level model is created using size-aware aggregation. Simultaneously, the global model is formed by aggregating all client updates using weights determined by both local dataset size and cluster priority scores derived from smoothed cluster losses. The cluster model serves as the teacher, while each client model, initialized from the global model, acts as the student during local optimization within a hierarchical knowledge distillation scheme. We evaluate DEFFT on MNIST, CIFAR-10, and FEMNIST, demonstrating improved fairness metrics and competitive global accuracy compared to FEDAVG and Q-FEDAVG.

1 Introduction

Federated Learning (FL) has emerged as a promising distributed training approach that allows edge clients to collaboratively develop a global model without sharing their raw data. In each training round, client models are trained in parallel on local datasets, with only the model updates being sent to a central server for aggregation. This architecture is particularly advantageous for privacy-sensitive applications, such as mobile health monitoring, smart home systems, and personalized recommendation services.

In real-world FL setups, clients exhibit notable heterogeneity in local data distributions, label spaces, and sampling frequencies, violating the IID (Independently and Identically Distributed) assumption underlying standard federated optimization [1]. Under such non-IID conditions, gradients aggregated across clients often conflict, leading to slower convergence and reduced model stability. This heterogeneity further induces systemic unfairness by biasing aggregation toward dominant data distributions, causing global models to underperform for minority clients. These expose the shortcomings of FL strategies that treat clients as interchangeable contributors and motivate algorithms that explicitly account for distributional heterogeneity.

Current efforts to address heterogeneity and fairness in FL can be categorized into two main approaches. The first category modifies the optimization process using techniques such as FEDPROX [2], FEDNOVA [3], and SCAFFOLD [4] to stabilize convergence with non-IID data. The second category focuses directly on fairness and personalization, exemplified by AGNOSTIC FL [5], which targets performance for the worst-performing client, and Q-FFL [6], which weights client updates based on their losses. However, optimization methods often overlook performance disparities caused by distributional imbalances, while fairness-focused techniques usually complicate learning dynamics. Additionally, clustered FL (CFL) [7] makes local training independent of cluster membership, while distillation methods often ignore useful subpopulations.

The proposed approach, DEFFT (Distillation-Enabled Fair Federated Training), aims to build a global model that explicitly accounts for distributional heterogeneity among clients to enable fairer and more effective collaboration. Instead of treating all clients as interchangeable contributors, DEFFT identifies distributional similarities within this heterogeneity and organizes clients into persistent groups accordingly. The core motivation is to ensure that each client primarily learns from peers with similar data characteristics. To support this, each group forms a cluster-level model that provides structured guidance to its members, while client models initialized from the global model are updated within this hierarchy. This similarity-aware intermediate layer allows each client to learn primarily from peers with aligned data characteristics, reducing interference under heterogeneity and promoting fairer optimization. Importantly, DEFFT operates without modifying the standard federated communication protocol, making it readily deployable in existing FL systems.

2 Problem Formulation

We consider an FL system with K clients. Each client $k \in \{1, \dots, K\}$ holds a private local dataset $\mathcal{D}_k = \{(x_i^k, y_i^k)\}_{i=1}^{n_k}$, drawn from an unknown client-specific data distribution. The data are non-IID across clients, i.e., the underlying distributions may differ between any two clients. Clients collaboratively train a global model \mathcal{M} under a central server, without exchanging raw data. After convergence \mathcal{M} is evaluated on each client’s local holdout test set (drawn from \mathcal{D}_k). Let $\ell_k(\mathcal{M})$ denote the empirical loss of \mathcal{M} on client k , and let $\varphi_k(\mathcal{M})$ denote a task-specific performance metric (e.g., accuracy). The average client loss and mean performance are denoted by $\bar{\ell}(\mathcal{M})$ and $\bar{\varphi}(\mathcal{M})$, respectively. To quantify inter-client performance disparity, we define the variance of client performance as

$$\text{Var}(\varphi(\mathcal{M})) = \frac{1}{K} \sum_{k=1}^K (\varphi_k(\mathcal{M}) - \bar{\varphi}(\mathcal{M}))^2. \quad (\text{III.1})$$

Our objective is to learn a global model that achieves both high average performance and equitable performance across clients. We formalize this trade-off using the following fairness-regularized objective:

$$\min_{\mathcal{M}} \mathcal{L}(\mathcal{M}) = \bar{\ell}(\mathcal{M}) + \lambda \text{Var}(\varphi(\mathcal{M})), \quad \lambda > 0, \quad (\text{III.2})$$

where λ controls the relative importance of performance equity. We emphasize that (III.2) serves as a conceptual objective guiding the design of the proposed method, rather than a loss that is optimized explicitly during federated training.

Given a baseline model \mathcal{M}' , the effectiveness of the proposed method is evaluated by verifying that the learned model \mathcal{M} satisfies

$$\bar{\ell}(\mathcal{M}) \leq \bar{\ell}(\mathcal{M}'), \quad \text{Var}(\varphi(\mathcal{M})) < \text{Var}(\varphi(\mathcal{M}')).$$

To complement this, we report *Jain’s fairness index* [8, 9], which quantifies the equity of performance across clients as:

$$J(\varphi(\mathcal{M})) = \frac{\left(\sum_{k=1}^K \varphi_k(\mathcal{M})\right)^2}{K \sum_{k=1}^K \varphi_k(\mathcal{M})^2}, \quad (\text{III.3})$$

where $J(\varphi(\mathcal{M})) \in \left[\frac{1}{K}, 1\right]$, and values closer to 1 indicate higher fairness, i.e., more uniform performance across clients.

3 Related Work

Research on FL in the horizontal setting [10], where clients share a common feature space but have distinct local samples, has highlighted the challenge of addressing statistical heterogeneity. Additionally, fairness in FL has become a critical consideration,

encompassing dimensions like performance fairness, collaboration fairness, and group fairness. Performance fairness aims to ensure a uniform performance distribution among participants, despite biased data distributions [10]. Since DEFFT specifically aims to enhance performance fairness in the context of statistical heterogeneity (non-IID data) in horizontal FL, this section highlights the most relevant literature on the topic.

Several approaches address statistical heterogeneity primarily through optimization stabilization, rather than fairness-aware objectives. For instance, FEDPROX [2] uses a proximal regularization term to prevent local updates from deviating excessively from the global model, thus mitigating client drift in non-IID environments. FEDNOVA [3] normalizes client updates based on varying local training efforts to enhance aggregation consistency with global objectives. SCAFFOLD [4] corrects client drift using control variates to align local and global optimization, leading to faster and more stable convergence in non-IID settings. Similarly, MOON [11] introduces a model-contrastive regularization term that encourages alignment between local and global models, mitigating overfitting to biased local data. While these methods effectively enhance convergence stability and overall performance, they remain fundamentally centered on global performance metrics. None of these approaches explicitly models or optimizes disparities in client-level performance, which means they do not directly tackle the fairness gaps that emerge when clients have imbalanced or highly skewed data distributions.

The AFL [5] frames training as a worst-case empirical risk minimization problem, aiming to create models that perform well across diverse client data distributions. Building on this, Q-FFL [6] and its implementation Q-FEDAVG introduce a fairness-aware objective that adjusts client updates based on their losses, focusing on underserved clients to reduce performance disparities. FAIRFL [12], employs deep multi-agent reinforcement learning with secure aggregation to optimize accuracy and fairness while preserving privacy. FCFL [13] formulates FL as a constrained multi-objective optimization problem that minimizes the maximum client loss under fairness constraints. DITTO [14] maintains personalized models for clients that are regularized toward the global model in order to improve fairness. More recently, F3 [15] introduces an adaptive regularization framework that balances global accuracy and performance fairness by considering client loss variance. Meanwhile, FEDMH [16] uses historical gradient information to manage the fairness-accuracy trade-off. Rather than modifying the optimization objective, a parallel line of work enforces fairness at the system and aggregation level. FEDFAIM [17] introduces non-monetary incentives based on aggregation and reward fairness principles. FADE [18] further targets bias in the aggregation process by explicitly mitigating discriminatory effects when combining client updates.

Several works address Non-IIDness by clustering clients with similar distributions, clustered FL (CFL), and transferring knowledge between models specific to each cluster through distillation. CFLKD [19] performs CFL by grouping clients according

to the similarity of their local model parameters and applies cross-group knowledge distillation to enable knowledge transfer between cluster models without direct parameter aggregation. DisUE [20] is a CFL framework enhanced by distillation that creates a universal expert by aggregating knowledge from cluster-specific models, thus improving generalization across diverse clients.

4 The Proposed DEFFT Framework

The proposed DEFFT framework consists of three primary components: Hierarchical client organization, Cluster-aware aggregation, and Hierarchical knowledge distillation. This section further discusses these components.

4.1 Hierarchical Client Organization

Before training begins, each client k sends its class distribution vector

$$d_k = [d_{k1}, d_{k2}, \dots, d_{kC}],$$

where C is the total number of classes and d_{kc} is the proportion of local samples in class c to the server. This minimal communication preserves data privacy. Using these distribution summaries, DEFFT organizes clients into groups according to distributional similarities.

This grouping is performed once and remains fixed throughout the training, resulting in persistent client groups that guide subsequent aggregation and knowledge transfer. At a high level, DEFFT requires a clustering mechanism that partitions clients into disjoint clusters based on a similarity measure capturing relationships among their local data distributions. The DEFFT itself is agnostic to the specific choice of similarity metric or clustering algorithm, provided that the resulting groups capture meaningful similarities among clients. In this work, we implement this step using a label distribution based hierarchical clustering approach, described below.

Implementation of Client Clustering. At the server, pairwise distributional dissimilarity between clients is measured using the Jensen–Shannon Divergence (JSD) [21]. JSD is preferred because it is symmetric, bounded, and remains finite with different label supports, making it suitable for non-IID data. Additionally, it offers an information theoretic interpretation of the loss from combining two label-generating sources, helping to group clients for optimal training.

The resulting pairwise JSD values form a condensed distance matrix $\mathbf{D} \in \mathbb{R}^{K(K-1)/2}$, which is used as input to Hierarchical Agglomerative Clustering (HAC) [22] with average linkage. HAC dendrogram systematically merges clients based on increasing distributional dissimilarity. A crucial aspect is to determine an appropriate cut thresh-

old for the dendrogram, which effectively distinguishes natural client groups without enforcing an arbitrary number of clusters.

Instead of pre-defining the number of clusters, we used a data driven threshold selection strategy to identify the largest merge gap in the hierarchical tree. Let $Z \in \mathbb{R}^{(K-1) \times 4}$ denote the linkage matrix returned by HAC, with $Z_{n,3}$ representing the JSD distance of the n -th merge. Distances close to the root of the dendrogram often reflect pathological merges between highly dissimilar groups. To mitigate this effect, the largest $(1 - \kappa)$ fraction of merge distances is discarded, with $\kappa = 0.95$. The remaining distances are sorted, and the largest successive difference is selected as the cut threshold T_{JSD} . Applying this threshold partitions clients into clusters $\{\mathcal{C}_g\}_{g=1}^G$, where clients within each cluster exhibit high label-distribution similarity. These clusters define the intermediate hierarchical level used for aggregation and knowledge distillation in the subsequent training stages.

4.2 Cluster-Level Aggregation and Global Model Updates

Once the partitioning of clients has been finalized, each FL round of DEFFT performs a cluster-aware aggregation driven by the post-training local losses. At the end of local training in round e , each client k reports its training loss $\ell_k^{(e)}$ and its updated model $\mu_k^{(e)}$. For each cluster g , the server computes the mean cluster loss, $\bar{\ell}_g^{(e)} = \frac{1}{|\mathcal{C}_g|} \sum_{k \in \mathcal{C}_g} \ell_k^{(e)}$, where \mathcal{C}_g is the set of clients assigned to cluster g . To minimize round to round noise, an exponential moving average (EMA) is applied:

$$\tilde{\ell}_g^{(e)} = \beta \tilde{\ell}_g^{(e-1)} + (1 - \beta) \bar{\ell}_g^{(e)}, \quad (\text{III.4})$$

with smoothing coefficient $\beta \in (0, 1)$ and initialization $\tilde{\ell}_g^{(0)} = \bar{\ell}_g^{(0)}$.

The smoothed clusters' losses resulting from EMA application, $\{\tilde{\ell}_g^{(e)}\}$, are subsequently linearly rescaled across clusters in each round. This rescaling ensures that the smallest value is adjusted to 0.1, while the largest value is adjusted to 1.0, with all intermediate clusters' values placed proportionally in between. To ensure numerical stability, a small constant is added prior to the scaling process. The resulting values establish the cluster priority scores, denoted by $\rho_g^{(e)}$ for cluster g at round e , and $\rho_g^{(e)} \in [0.1, 1.0]$.

For cluster g , the cluster model \mathcal{M}_g^e at round e is computed as a weighted average of the participating clients' updated models, where each client's weight is proportional to the size of its local dataset.

$$\mathcal{M}_g^{(e+1)} = \sum_{k \in \mathcal{C}_g} w_k \mu_k^{(e)}, \quad w_k = \frac{n_k}{\sum_{j \in \mathcal{C}_g} n_j}, \quad (\text{III.5})$$

where n_k represents the number of local training samples for client k . This data size-based weighting is effective due to the assumption that clients within a cluster possess

approximately IID data, ensuring a statistically sound and unbiased aggregation. Global aggregation then incorporates both dataset size and cluster priority scores. For a client k belonging to cluster \mathcal{C}_g , the global aggregation weight is defined as:

$$\mathcal{M}^{(e+1)} = \sum_k \tilde{w}_k \mu_k^{(e)}, \quad \tilde{w}_k = \frac{n_k \rho_g^{(e)}}{\sum_j n_j \rho_{gj}^{(e)}}. \quad (\text{III.6})$$

4.3 Knowledge Distillation Across Hierarchical Levels

DEFFT employs a hierarchical knowledge distillation mechanism to ensure that each client learns primarily from distributionally similar peers. Cluster models, formed by aggregating peer clients within the same group (III.5), provide structured guidance while preserving data privacy. For each round $e > 1$, the server provides two models to every active client k in cluster g . The first model, denoted by $\mathcal{M}_g^{(e)}$, which serves as the teacher. The second model is the global model, denoted by $\mathcal{M}^{(e)}$, which provides the student initialization. The teacher network is instantiated with the parameters of $\mathcal{M}_g^{(e)}$ and kept frozen during local optimization, while the student network is initialized with $\mathcal{M}^{(e)}$ and updated on clients local data. Let $\tilde{y}^{\mu_k^{(e+1)}}$ denote the student logits and $\tilde{y}^{\mathcal{M}_g^{(e)}}$ the teacher logits. The local objective is [23];

$$\mathcal{L} = (1 - \lambda) \mathcal{L}_{\text{CE}}\left(\tilde{y}^{\mu_k^{(e+1)}}, y\right) + \lambda \tau^2 \mathcal{L}_{\text{KD}}\left(\tilde{y}^{\mu_k^{(e+1)}}, \tilde{y}^{\mathcal{M}_g^{(e)}}\right), \quad (\text{III.7})$$

where λ regulates the influence of the distillation term and τ denotes the temperature. The KD term applies a temperature-scaled *Kullback–Leibler* divergence between the softened predictive distributions of the student and teacher, guiding the student toward the cluster-level representation and reducing client drift under non-IID data.

KD is disabled in two cases: (i) during the first global round, when there are no cluster models, and (ii) for clients belonging to inactive clusters in the previous round, whose cluster models are not updated and are therefore unsuitable as teachers. In these cases, the local objective reduces to the standard cross-entropy loss.

4.4 Overall DEFFT Procedure

The overall workflow of DEFFT integrates its main components - clustering, aggregation, and distillation mechanisms - into a unified FL process. During the initialization phase, the server establishes a hierarchical organization of clients (Sec. 4.1). In each global round e , the server broadcasts the current global model, $\mathcal{M}^{(e)}$, to all selected clients. For rounds where $e > 1$, clients whose clusters were active in the previous round ($e - 1$) also receive their cluster model, $\mathcal{M}_g^{(e)}$. The clients then perform local

Algorithm 3 DEFFT

```
1: procedure SERVER-SIDE
2:   Collect client distributions  $\{d_k\}_{k=1}^K$  and form clusters  $\{C_g\}_{g=1}^G$  (Sec. 4.1)
3:   Initialize global model  $\mathcal{M}^{(1)}$ 
4:   for  $e = 1$  to  $E$  do
5:     Sample clients  $S^{(e)}$ 
6:     for all  $k \in S^{(e)}$  in parallel do
7:       Send  $\mathcal{M}^{(e)}$  and cluster model  $\mathcal{M}_g^{(e)}$  if available
8:        $(\mu_k^{(e)}, \ell_k^{(e)}) \leftarrow \text{CLIENT-UPDATE}(k)$ 
9:     end for
10:    Compute cluster weights  $\{\rho_g^{(e)}\}$  (Sec. 4.2)
11:    Update  $\{\mathcal{M}_g^{(e+1)}\}$  via (III.5)
12:    Update  $\mathcal{M}^{(e+1)}$  via (III.6)
13:  end for
14: end procedure
15: procedure CLIENT-UPDATE(Client  $k$ )
16:   Initialize local model from  $\mathcal{M}$ 
17:   Train using CE loss or KD loss if cluster model received (Sec. 4.3)
18:   return  $(\mu_k, \ell_k)$ 
19: end procedure
```

training based on the hierarchical KD objective (Sec. 4.3), while clients in inactive clusters optimize solely using the standard cross entropy loss.

Once local training is completed, the server performs aggregation at two hierarchical levels (Sec. 4.2). At the cluster level, client updates are merged to update each $\mathcal{M}_g^{(e)}$, which is retained as the teacher model for the next round. At the global level, all client updates are aggregated with weights determined by both dataset size and cluster priority score, giving more effective clusters proportionally greater influence on the global update. The complete process is described in Algorithm 3.

4.5 Convergence Analysis of the Optimization Objective

This section analyzes the convergence behavior of DEFFT and clarifies its relationship to the conceptual fairness-regularized objective in (III.2). Since DEFFT does not explicitly optimize this objective, its cluster-aware aggregation and hierarchical KD modify the effective optimization landscape.

Assumptions. We adopt the following standard assumptions [2].

(A1) Each client loss $\ell_k(\cdot)$ is L -smooth.

(A2) The variance of stochastic gradients on each client is bounded:

$$\mathbb{E}\|\nabla\ell_k(\mathcal{M};\xi) - \nabla\ell_k(\mathcal{M})\|^2 \leq \sigma^2, \quad \forall k.$$

(A3) Each client performs a finite number of local optimization steps with a bounded step size, ensuring bounded model drift:

$$\mathbb{E}\|\mu_k^{(e)} - \mathcal{M}^{(e)}\|^2 \leq \Delta^2, \quad \forall k, e.$$

(A4) The global aggregation weights, $\tilde{w}_k^{(e)}$ (III.6), satisfy $\tilde{w}_k^{(e)} \in (0, 1)$ and $\sum_{k=1}^K \tilde{w}_k^{(e)} = 1$ for all e . Let $\tilde{\mathbf{w}}^{(e)} = (\tilde{w}_1^{(e)}, \dots, \tilde{w}_K^{(e)})$. Due to EMA smoothing of cluster losses, the weight sequence varies slowly:

$$\|\tilde{\mathbf{w}}^{(e+1)} - \tilde{\mathbf{w}}^{(e)}\|_1 \leq \epsilon_e, \quad \sum_{e=1}^{\infty} \epsilon_e < \infty.$$

(A5) For clients receiving a cluster teacher model ($\mathcal{M}_g^{(e)}$), the KD-augmented local objective reduces expected deviation from the cluster model compared to standard local training:

$$\mathbb{E}\|\mu_k^{(e)} - \mathcal{M}_g^{(e)}\|^2 \leq \eta \mathbb{E}\|\mu_k^{(e)} - \mathcal{M}^{(e)}\|^2, \quad \eta \in (0, 1).$$

This assumption reflects the regularizing effect of distillation toward a cluster-consensus model.

Effective Optimization Objective. Due to cluster-aware reweighting, the global update in round e corresponds to a weighted aggregation ($\tilde{w}_k^{(e)}$) of local models. This induces a time-varying weighted empirical risk:

$$F_e(\mathcal{M}) = \sum_{k=1}^K \tilde{w}_k^{(e)} \ell_k(\mathcal{M}), \quad (\text{III.8})$$

where $\tilde{w}_k^{(e)}$ reflects both dataset size and cluster priority. Unlike FEDAVG, where weights are fixed, DEFFT dynamically adjusts $\tilde{w}_k^{(e)}$ according to cluster-level training difficulty.

Convergence Behavior of the Global Model. Under Assumptions (A1)–(A5), the DEFFT global update can be interpreted as standard gradient descent on time varying weighted objective (III.8). Assumptions (A3) and (A4) bound, respectively, the client drift induced by local training and the round-to-round drift of the aggregation weights. Standard nonconvex analyses of local SGD then imply stability and ergodic convergence to a neighborhood of first-order stationary points of the effective objective

sequence $\{F_e\}$, where the radius of this neighborhood increases with the client drift level Δ and the aggregation weight drift $\{\epsilon_e\}$.

Effect of Hierarchical KD. The hierarchical KD mechanism reduces client drift by anchoring local updates to cluster-level models trained on distributionally similar data. By decreasing the expected deviation between local client models and the global model, KD improves the alignment of local updates with the global descent direction, thereby enhancing optimization stability under non-IID data. Importantly, KD does not alter the global aggregation rule; rather, it improves the quality and consistency of the local updates being aggregated.

Relation to the Fairness-Regularized Objective. Although DEFFT does not explicitly minimize the fairness-regularized objective in (III.2), its dynamic reweighting scheme emphasizes clusters with persistently higher loss. This biases optimization toward improving the worst-performing clusters and reducing tail-client risk, similar to minimizing a smoothed max-loss or Conditional Value at Risk (CVaR) objective. Under mild conditions where client performance metrics correlate monotonically with loss, reducing dispersion in cluster losses empirically leads to reduced inter-client performance variance, as confirmed by improvements in $\text{Var}(\varphi(\mathcal{M}))$ and Jain’s fairness index.

5 Experimental Setup

This section describes the experimental setup used to evaluate the performance of DEFFT under different settings.

Datasets and Models: We evaluate DEFFT on three widely used benchmark datasets: MNIST, CIFAR-10, and FEMNIST. Since MNIST and CIFAR-10 are originally homogeneous datasets, we introduce statistical heterogeneity across clients by partitioning the data using Dirichlet sampling [24] with concentration parameter $\alpha = 1$, resulting in 92 clients for MNIST and 127 clients for CIFAR-10. In contrast, FEMNIST is inherently designed for distributed and non-IID FL and is therefore used with its original client-wise data partitioning. For MNIST, we employ a lightweight neural architecture suitable for low-complexity image classification. For CIFAR-10 and FEMNIST, we adopt moderately deep convolutional models with normalization and regularization components to ensure stable optimization under heterogeneous client data distributions. Further information can be found in the supplementary materials.

Baselines: We compare DEFFT with FEDAVG [1] and Q-FEDAVG [6]. FEDAVG serves as the baseline for federated optimization, while Q-FEDAVG is a fairness-aware method that reweights client updates based on the current global model’s loss. This comparison helps us evaluate if the hierarchical aggregation and distillation mechanisms enhance average performance and inter-client fairness beyond loss-based reweighting. All methods are implemented in PyTorch, and the source code is publicly available on Anonymous GitHub.

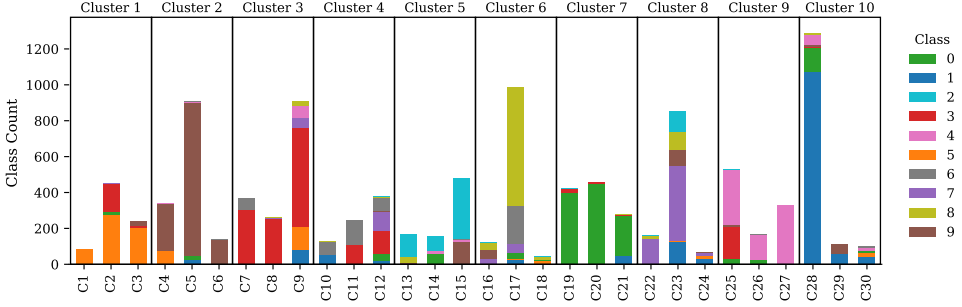


Figure 1: Class count distributions for the sampled clients in the MNIST dataset, using hierarchical agglomerative clustering in Ctrl St., with 3 samples per class.

Clustering Details. The MNIST and CIFAR-10 datasets were clustered using HAC, yielding 10 and 11 clusters, respectively. In contrast, the FEMNIST dataset demonstrates significantly different heterogeneity characteristics compared to MNIST and CIFAR-10, causing the direct application of label distribution based clustering to be inadequate. FEMNIST includes 62 character classes (digits, lowercase, and uppercase letters) with significant label sparsity per client and imbalanced local sample sizes. Given these conditions, JSD over the entire label simplex becomes unstable and fails to capture the main optimization challenges across clients. To address these challenges, we utilize a feature based client embedding specifically designed for FEMNIST, followed by model-based clustering. After receiving the label distributions from each client k , the server generates a compact embedding vector that captures three complementary aspects of its local data: (i) *semantic mass*: the total probability mass attributed to digits, lowercase letters, and uppercase letters; (ii) *distributional concentration*: represented by the entropy of the normalized label distribution, along with the maximum mass of a single label and the cumulative mass of the top five labels; and (iii) *data scale*: indicated by the logarithm of the client’s local sample size. These client embeddings are standardized and projected into a low-dimensional space using principal component analysis, then clustered using a full-covariance Gaussian Mixture Model [25]. The number of clusters is selected automatically via the Bayesian Information Criterion, resulting in 9 clusters.

Choice of Hyperparameters. For all algorithms, we maintained consistent optimization hyperparameters (learning rate, number of local and global rounds) within each dataset to ensure fair comparison. Detailed configurations are provided in the appendix. DEFFT introduces three additional hyperparameters: the distillation weight λ , the distillation temperature τ (III.7), and the EMA smoothing coefficient β (III.4). The weight λ balances local cross-entropy loss with the cluster-level knowledge distillation, where small values favor local training and large values promote alignment to the cluster model, potentially reducing personalization. τ adjusts the softness of the teacher distribution, with larger values yielding smoother predictions. Lastly,

β influences the stability of cluster-level loss estimates: larger values stabilize the aggregation weights, while smaller values may cause fluctuations. These hyperparameters were selected empirically based on validation performance and stability. For MNIST, milder heterogeneity required weaker regularization ($\lambda = 0.1$, $\tau = 2.0$) and smoother weighting ($\beta = 0.5$). For CIFAR-10, stronger distillation ($\lambda = 0.4$, $\tau = 3.0$) improved stability under higher task complexity, with $\beta = 0.5$ providing sufficient smoothing. For FEMNIST, which exhibits strong client heterogeneity, we used a small $\beta = 0.1$ for responsive cluster weighting and a higher $\tau = 3.5$ to stabilize distillation; the λ was adapted per client based on its relative cluster performance.

Controlled Study (Ctrl St). To evaluate the proposed method under controlled and balanced clustered participation, we consider a controlled setting on MNIST using the same pool of 92 clients obtained in the general experimental setup. We compute the pairwise JSD between client label distributions and cluster the clients into ten clusters $\{\mathcal{C}_g\}_{g=1}^{10}$. To ensure fair representation across clusters, we select a fixed number of participating clients from each cluster by sampling 3 clients per cluster, resulting in a total of $K = 30$ participating clients. Fig. 1 illustrates the class count distributions for the selected clients. Although clustering is performed based on label-distribution similarity, substantial heterogeneity remains both across clusters and among clients within the same cluster. Several clients exhibit extreme label skew, with dominance of one or two classes, while others have limited sample sizes. This confirms that the controlled setting preserves meaningful non-IID characteristics while ensuring proportional cluster representation, making it suitable for isolating the effects of hierarchical aggregation and KD. The selected client set is fixed throughout the training, and all selected clients participate in every global round. For fair comparison, baseline methods (FEDAVG and Q-FEDAVG) are also trained using the same set of participating clients. All methods used identical hyperparameter settings.

6 Results and Discussion

In this section, we present the results of the control study along with the benchmark dataset results for the three studied algorithms.

6.1 Control Study Results Analysis

Table 1 presents the per-client test accuracies in a controlled participation setting. The DEFFT consistently enhances the performance of low and mid performing clients (e.g., see C1, C2, C3, C13, and C15) compared to both FEDAVG and Q-FEDAVG. These clients belong to clusters that experience higher average training loss (Cluster 1 and Cluster 5), resulting in their assignment of elevated priority scores (Figure 2b). This higher priority increases the impact of their updates on the global model during the global aggregation process (Sec. 4.2).

Table 1: Results from the **MNIST Ctrl St.** Show per-client test accuracies. Note that the horizontal lines separate the different client clusters.

Client	FedAvg	q-FedAvg	DEFFT	Client	FedAvg	q-FedAvg	DEFFT
C1	0.250	0.438	0.500	C16	0.938	0.812	0.969
C2	0.500	0.554	0.696	C17	0.796	0.700	0.854
C3	0.321	0.482	0.607	C18	0.500	0.625	0.625
C4	0.788	0.825	0.862	C19	0.952	0.923	0.894
C5	0.888	0.906	0.902	C20	0.991	0.991	0.929
C6	0.781	0.844	0.812	C21	0.938	0.906	0.906
C7	0.841	0.864	0.864	C22	0.800	0.675	0.850
C8	0.812	0.875	0.750	C23	0.822	0.798	0.880
C9	0.821	0.812	0.830	C24	0.875	0.812	0.875
C10	0.875	0.875	0.875	C25	0.828	0.758	0.789
C11	0.875	0.875	0.875	C26	0.725	0.575	0.675
C12	0.841	0.761	0.795	C27	0.825	0.700	0.762
C13	0.525	0.575	0.725	C28	0.969	0.953	0.953
C14	0.725	0.725	0.775	C29	0.875	0.875	0.875
C15	0.558	0.617	0.767	C30	0.833	0.792	0.875

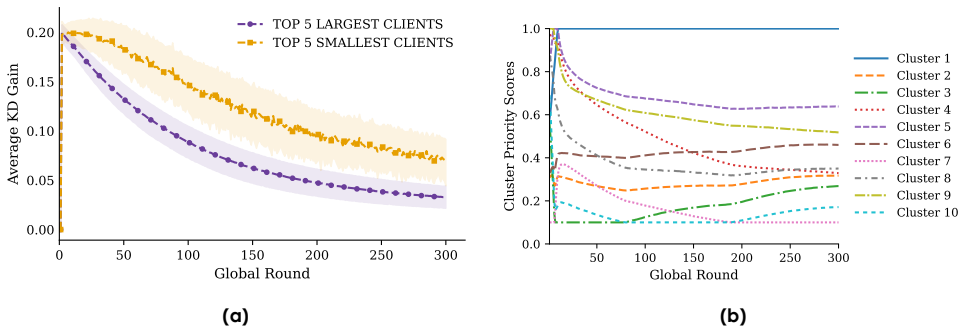


Figure 2: MNIST Ctrl St. (a) Average KD gain, defined as the validation loss improvement from hierarchical KD over standard local training, for the five largest and five smallest clients (by local dataset size). (b) Evolution of cluster priority scores across global rounds.

As illustrated in Figure 2a, KD gains are most highest in the early stages of training, gradually diminishing as the training progresses. This trend reflects the increasing alignment between the student and teacher models performance as they converge. Notably, clients with smaller datasets consistently demonstrate higher KD gains than those with larger datasets throughout the training process. This suggests that hierarchical KD provides a significant advantage to data-scarce clients by offering a robust regularization signal from the cluster level teacher, which helps mitigate local optimization instability and client drift in the presence of heterogeneous data. In contrast, clients with larger datasets, which benefit from richer local supervision, tend to rely less on the distillation signal, resulting in smaller yet consistently positive performance improvements.

Figure 2b illustrates that cluster priorities adapt rapidly in the early rounds, highlighting the differences in optimization challenges faced by different clusters. As training progresses, these priorities stabilize rather than collapsing to uniform weights, indicating convergence toward a persistent hierarchy that continues to shape aggrega-

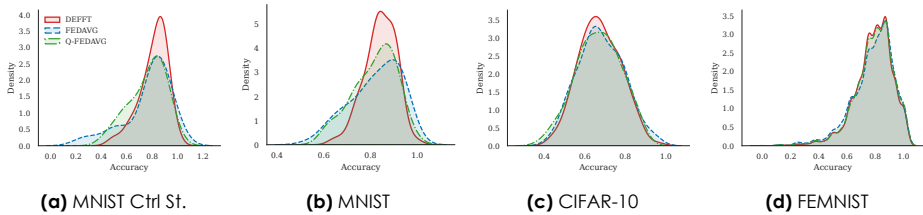


Figure 3: Kernel density estimates of per-client test accuracies at convergence. Results are averaged over five independent runs with different random client participation.

tion. This confirms that DEFFT does not reduce to FEDAVG at the later stages, even under fully controlled participation.

The aggregated results of the controlled study are presented in Table 2. Under identical participation and training conditions, DEFFT significantly enhances the mean accuracy of the global model (0.763 compared to 0.701 for FEDAVG and 0.686 for Q-FEDAVG), while also decreasing inter-client performance variance (0.024) by over 50% relative to FEDAVG (0.050). This reduction in dispersion is further evidenced by the tighter accuracy distribution observed in Figure 3a, which shows a clear left tail contraction and reduced spread under DEFFT. These trends are consistent with the higher Jain’s fairness index and consistent enhancements in tail performance metrics, including the 10th-percentile accuracy, worst-10% mean accuracy, and minimum client accuracy (see Table 1). In contrast, while Q-FEDAVG improves certain worst-case metrics through loss-based reweighting, it does not achieve comparable gains in average performance and displays higher residual variance. The results and dynamics of KD gain and cluster priority scores suggest that the improvements seen with DEFFT arise from hierarchical KD and adaptive cluster-aware aggregation, not just favorable sampling effects.

6.2 Evaluation on Benchmark Federated Datasets

Table 2 summarizes the benchmark results of the three FL algorithms on the studied datasets. Notably, DEFFT achieves the highest mean accuracy on MNIST while substantially reducing inter client variance compared to both FEDAVG and Q-FEDAVG. These gains indicate that the proposed hierarchical design effectively improves global performance despite severe client heterogeneity induced by Dirichlet sampling with $\alpha = 0.1$. On CIFAR-10, while FEDAVG attains the highest mean accuracy, DEFFT remains competitive and outperforms Q-FEDAVG. This behavior is expected given the increased task complexity and representation mismatch across clients in CIFAR-10, where aggressive reweighting can negatively affect global optimization.

Beyond average accuracy, DEFFT consistently improves fairness related metrics across both datasets. On MNIST, DEFFT records the lowest inter-client variance (0.005 vs. 0.014 for FEDAVG) and attains the highest Jain’s fairness index (0.993).

Additionally, it shows substantial gains in tail-performance metrics, including the 10th-percentile accuracy (0.741), worst-10% mean accuracy (0.695), and minimum client accuracy (0.631). On CIFAR-10, DEFFT achieves the lowest variance (0.0125) and improves tail metrics such as worst-10% mean accuracy (0.4706 vs. 0.4468 for Q-FEDAVG). In contrast, Q-FEDAVG exhibits less consistent fairness gains and degrades minimum accuracy (0.355 vs. 0.404 for FEDAVG). These results show that DEFFT provides a favorable balance between average accuracy and fairness. It enhances both performance and fairness when improvements in accuracy are possible (MNIST), and focuses on robustness and tail performance in more challenging scenarios (CIFAR-10). On FEMNIST, all methods achieve similar mean accuracy (0.788–0.792), with DEFFT showing the lowest variance (0.017) and the highest Jain’s index (0.975), indicating uniform performance across clients. Additionally, DEFFT improves the worst-10% mean accuracy (0.528) to match the best minimum accuracy (0.125), suggesting better robustness for weaker clients.

In addition to the quantitative metrics, the Kernel Density Estimation (KDE) plots (Figure 3) of client accuracies provide further evidence of the fairness improvements discussed above. For the MNIST dataset (Figure 3b), DEFFT achieves a significantly tighter distribution with a noticeable contraction of the left tail. Similarly, for CIFAR-10 (Figure 3c), DEFFT reduces the spread of the distribution without collapsing the right tail. In contrast, Q-FEDAVG results in broader or skewed distributions, which suggest uneven improvements among clients. These visual patterns are consistent with the reported variance and tail-performance metrics, highlighting that DEFFT enhances robustness and inter-client fairness through structured aggregation rather than through isolated client reweighting. For FEMNIST, the KDE curves similarly indicate a modest tightening of the accuracy distribution under DEFFT, consistent with its lower variance and improved tail metrics.

7 Conclusion and Future Directions

This study introduced DEFFT, a fairness-aware FL framework that organizes the clients based on distributional similarity and leverages hierarchical knowledge transfer to promote fair optimization. Through extensive evaluation against FEDAVG and Q-FEDAVG on MNIST, CIFAR-10, and FEMNIST, we demonstrated that DEFFT achieves competitive global performance while consistently improving fairness-related metrics. We have also studied the behavior of DEFFT in a controlled participation setting and found that it consistently enhanced the performance of low- and mid-performing clients throughout the training process. Future work will explore multi-level and soft clustering strategies that relax hard client assignments, allowing clients to share information across overlapping similarity groups. Such extensions may further reduce cross-client interference and improve fairness in highly heterogeneous federated environments

Table 2: The converged global model is evaluated on each client’s test set. Columns report mean accuracy (Acc), variance of client accuracies, Jain’s fairness index, 10th-percentile accuracy (P_{10}), mean accuracy of the worst 10% of clients (Worst-10%), and minimum client accuracy (Worst Acc). Arrows indicate whether higher (\uparrow) or lower (\downarrow) values are better. Results are reported as mean values over five independent runs.

Algorithm	Acc \uparrow	Var \downarrow	Jain’s \uparrow	P_{10} Acc \uparrow	W10% \uparrow	Min Acc \uparrow
MNIST CTRL St.						
FedAvg	0.701	0.050	0.907	0.360	0.197	0.179
q-FedAvg	0.686	0.045	0.913	0.373	0.304	0.250
DEFFT	0.763	0.024	0.961	0.548	0.405	0.313
MNIST Dataset						
FedAvg	0.821	0.014	0.980	0.663	0.578	0.480
q-FedAvg	0.816	0.010	0.985	0.667	0.611	0.518
DEFFT	0.841	0.005	0.993	0.741	0.695	0.631
CIFAR-10 Dataset						
FedAvg	0.678	0.015	0.969	0.524	0.468	0.404
q-FedAvg	0.667	0.014	0.969	0.509	0.447	0.355
DEFFT	0.671	0.013	0.973	0.534	0.471	0.354
FEMNIST Dataset						
FedAvg	0.788	0.020	0.968	0.625	0.450	0.100
q-FedAvg	0.792	0.018	0.972	0.625	0.520	0.125
DEFFT	0.791	0.017	0.975	0.625	0.528	0.125

References

- [1] B. McMahan et al. “Communication-efficient learning of deep networks from decentralized data.” In: *AI and Statistics*. PMLR. 2017, pp. 1273–1282.
- [2] T. Li et al. *Federated Optimization in Heterogeneous Networks*. 2020. arXiv: 1812.06127 [cs.LG].
- [3] J. Wang et al. “Tackling the objective inconsistency problem in heterogeneous federated optimization.” In: *Advances in neural infor. proc. syst.* 33 (2020), pp. 7611–7623.
- [4] S. P. Karimireddy et al. “Scaffold: Stochastic controlled averaging for federated learning.” In: *International conference on ML*. PMLR. 2020, pp. 5132–5143.
- [5] M. Mohri and et al. *Agnostic Federated Learning*. 2019. arXiv: 1902.00146 [cs.LG].
- [6] T. Li, M. Sanjabi, A. Beirami, and V. Smith. “Fair resource allocation in federated learning.” In: *International Conference on Learning Representations*. 2019.
- [7] F. Sattler, K.-R. Müller, and W. Samek. “Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints.” In: *IEEE transactions on neural networks and learning systems* 32.8 (2020), pp. 3710–3722.

- [8] R. K. Jain, D.-M. W. Chiu, W. R. Hawe, et al. “A quantitative measure of fairness and discrimination.” In: *Eastern Research Laboratory, Digital Equipment Corporation, Hudson, MA 21.1* (1984), pp. 2022–2023.
- [9] D. M. Chiu. *A quantitative measure of fairness and discrimination for resource allocation in shared computer systems*. Tech. rep. Digital Equipm. Corporat., 1984.
- [10] T. H. Rafi et al. “Fairness and privacy preserving in federated learning: A survey.” In: *Information Fusion* 105 (2024), p. 102198.
- [11] Q. Li, B. He, and D. Song. “Model-contrastive federated learning.” In: *Proc. of the IEEE/CVF conf. on comp. vision and pattern recognition*. 2021, pp. 10713–10722.
- [12] D. Y. Zhang, Z. Kou, and D. Wang. “Fairfl: A fair federated learning approach to reducing demographic bias in privacy-sensitive classification models.” In: *IEEE Int. Conference on Big Data*. 2020, pp. 1051–1060.
- [13] S. Cui, W. Pan, J. Liang, C. Zhang, and F. Wang. “Addressing algorithmic disparity and performance inconsistency in federated learning.” In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 26091–26102.
- [14] T. Li and et al. *Ditto: Fair and Robust Federated Learning Through Personalization*. 2021. arXiv: 2012.04221 [cs.LG].
- [15] J. Pei. “F3: Fair Federated Learning Framework with adaptive regularization.” In: *Knowledge-Based Systems* 316 (2025), p. 113392.
- [16] T. Zhu, Y. Lin, Y. Qu, Z. Liu, Y. Luo, T. Mao, and Z. Chen. “Federated learning with empirical insights: Leveraging gradient historical experiences for performance fairness.” In: *Pervasive and Mobile Computing* (2025), p. 102061.
- [17] Z. Shi, L. Zhang, Z. Yao, L. Lyu, C. Chen, L. Wang, J. Wang, and X.-Y. Li. “Fedfaim: A model performance-based fair incentive mechanism for federated learning.” In: *IEEE Transactions on Big Data* 10.6 (2022), pp. 1038–1050.
- [18] A.-A. Bendoukha, H. H. Arcolezi, N. Kaaniche, A. Boudguiga, R. Sirdey, and P.-E. Clet. “FADE: Federated Aggregation with Discrimination Elimination.” In: *Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency*. 2025, pp. 3182–3195.
- [19] J. Zheng, S. Zhao, P. Hu, and X. Shen. “CFLKD: Clustered Federated Learning via Cross-Group Knowledge Distillation.” In: *Future Generation Computer Syst.* (2025), p. 108253.
- [20] Z. Leng, C. Zhang, G. Long, R. Xia, and B. Yang. “Distilling A Universal Expert from Clustered Federated Learning.” In: *arXiv e-prints* (2025), arXiv–2506.

- [21] M. L. Menéndez, J. A. Pardo, L. Pardo, and M. d. C. Pardo. “The jensen-shannon divergence.” In: *Journal of the Franklin Institute* 334.2 (1997), pp. 307–318.
- [22] F. Murtagh and P. Contreras. “Algorithms for hierarchical clustering: an overview.” In: *Wiley interdiscipl. reviews: data mining and knowledge discovery* 2.1 (2012), pp. 86–97.
- [23] A. Mora, I. Tenison, P. Bellavista, and I. Rish. “Knowledge distillation for federated learning: a practical guide.” In: *arXiv preprint arXiv:2211.04742* (2022).
- [24] T. Lin et al. “Ensemble distillation for robust model fusion in federated learning.” In: *Advances in neural information processing systems* 33 (2020), pp. 2351–2363.
- [25] J. D. Banfield and A. E. Raftery. “Model-based Gaussian and non-Gaussian clustering.” In: *Biometrics* (1993), pp. 803–821.

..

Federated Learning (FL) is a distributed learning paradigm that enables multiple clients to collaboratively train a shared model without centralizing their data. This design supports learning in decentralized, heterogeneous, and data-constrained settings while providing privacy benefits by keeping raw data local. However, in practical implementations, client data are typically non-independent and identically distributed (non-IID). This resulting in heterogeneous learning dynamics and unequal benefits across participants. Improvements in average global performance can mask performance degradation for disadvantaged clients, highlighting a structural fairness challenge in FL. This thesis argues that achieving fairness under non-IID FL requires explicit understanding and modeling of client behavioral heterogeneity rather than uniform aggregation of client updates.

In addressing the issue of fairness in FL under data heterogeneity, the thesis first studies and analyzes clients' deviating behavior during the federated training process. An eccentricity-based approach is introduced to quantify deviations in local models and data representations within the global model, enabling systematic identification of atypical contribution and benefit patterns. The insights gained lay the foundation for our further research into developing novel, fairness-aware FL solutions for heterogeneous, distributed learning setups.

Then it proposes a fairness-aware aggregation framework called FEDABOOST that adapts client influence based on local performance signals. By dynamically weighting client updates and adjusting local optimization to emphasize hard examples, the method reduces disparities across heterogeneous clients while maintaining competitive global performance. Later, the thesis introduces DEFFT, a clients distribution-aware framework that models latent similarities among clients through persistent grouping based on label distributions. Cluster-level models and hierarchical knowledge distillation integrate inter-client structure into the learning process, enhancing fairness metrics along with overall accuracy.

Across multiple benchmark datasets, the proposed approaches demonstrate that a principled way to modeling heterogeneity can lead to measurable improvements in fairness without compromising global performance. The three discussed studies together establish a structured framework for mitigating unequal benefits in FL under non-IID data distributions.

