Universidade do Minho
Escola de Ciências

Duarte Miguel Ferreira Moreira da Silva

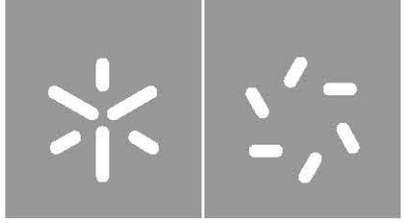
Cross-Silo Federated Learning: An Empirical Study on Learning Gains

March 2026

Cross-Silo Federated Learning: An Empirical Study on Learning Gains

Duarte Miguel Ferreira Moreira da Silva

UMinho | 2026



Universidade do Minho
Escola de Ciências

Duarte Miguel Ferreira Moreira da Silva

Cross-Silo Federated Learning: An Empirical Study on Learning Gains

Master's Dissertation Matemática e Computação

Dissertation supervised by

José Pedro Miranda Mourão Patrício
Antoine Boutet

March 2026

Copyright and Terms of Use for Third Party Work

Acknowledgments

It has been a long journey, and this work is, above all, a shared endeavor. First, I would like to thank Professor Antoine Boutet for agreeing to supervise this project, for facilitating the logistics of my stay in Lyon, and for the stimulating scientific discussions throughout the project.

I am also grateful to Helen Pouchot for her patience during this process, as well as to all the Inria members with whom I had enriching conversations during my stay, with a special note to the Inria Privatics team members. In particular, I extend a special thanks to Alix and Salvatore for the warm welcome. I now consider them friends.

I would also like to thank Professor Pedro Patrício for his attentiveness, readiness, guidance, and friendly support.

Beyond academia, I owe heartfelt thanks to my family — my parents, siblings, aunts, and grandparents — for their constant encouragement, understanding, and kind words. To my friends, companions along the journey, I thank you for making my academic years more joyful. To all others who have contributed to my current standing, I extend my gratitude to you as well.

Finally, I would like to thank Klara Stokes and Vicenç Torra, whose guidance years ago sparked my interest in the field of privacy in machine learning and motivated me to explore this area further.

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

Resumo

Federated Learning (FL) é uma abordagem essencial para o treino colaborativo de modelos de *Machine Learning* (ML) em dados distribuídos e sensíveis, sendo crucial para a proteção da privacidade em cenários de *cross-silo*. No entanto, a sua adoção é amplamente limitada pela incerteza quanto ao ganho de aprendizagem (*learning gain*), ou seja, se o modelo global oferece melhor precisão e generalização do que um modelo exclusivamente treinado localmente, o que é um fator de incentivo frequentemente negligenciado na literatura que se foca em desafios de segurança e heterogeneidade. Para maximizar o desempenho e a motivação dos participantes, este trabalho propõe um novo esquema de aprendizagem em duas fases: FedFil (*Filtered Federated Learning*). O FedFil opera através de: 1) filtragem local de dados, que emprega Deteção de Anomalias para remover dados influentes antes do treino em FL; e 2) *Fine-Tuning* pós-FL, para personalizar o modelo global às características específicas do cliente. Este projeto, "Cross-Silo Federated Learning: An Empirical Study on Learning Gains", realiza uma investigação empírica aprofundada para quantificar o ganho de aprendizagem alcançado com o FedFil, identificar quais os clientes que mais beneficiam da colaboração e avaliar se a adaptação local melhora o ganho de precisão global. Os nossos resultados indicam que o FedFil, ao filtrar localmente os pontos influentes antes dos ciclos de treino federado e ao personalizar o modelo global posteriormente, alcança um *learning gain* médio de cerca de 8% nas avaliações locais dos clientes e de 6% nas avaliações de generalização.

Palavras-Chave: *Cross-silo Federated Learning*, Filtragem de dados, Deteção de Anomalias.

Abstract

Federated Learning (FL) is an essential approach for the collaborative training of Machine Learning (ML) models on distributed and sensitive data, being crucial for privacy protection in cross-silo scenarios. However, its adoption is widely limited by the uncertainty surrounding the learning gain, i.e., whether the global model offers better performance and generalization than a model exclusively trained locally. This uncertainty represents an incentive factor often overlooked in literature focused on security and heterogeneity challenges. To improve participant performance and motivation, this work proposes a novel two-stage learning scheme: FedFil (Filtered Federated Learning). FedFil operates through: 1) local data filtering, which employs Anomaly Detection to remove influential datapoints before FL training; and 2) post-FL Fine-Tuning, to personalize the global model to the specific characteristics of the client. This project, "Cross-Silo Federated Learning: An Empirical Study on Learning Gains", conducts a thorough empirical investigation to quantify the learning gain achieved with FedFil, identify which clients benefit most from collaboration, and evaluate whether local adaptation improves global accuracy gain. Our results indicate that FedFil, by filtering influential local datapoints before the learning rounds and personalizing the global model afterwards, achieves an average learning gain of around 8% on local evaluations and 6% on generalization evaluations.

Keywords: Cross-silo Federated Learning, Data Filtering, Anomaly Detection.

Contents

List of Figures **X**

List of Tables **XIV**

List of Algorithms **XV**

1 Introduction **1**

1.1 Motivation 1

1.2 Goals 2

1.3 Structure of dissertation 3

1.4 Presentation of Inria Privatics 4

1.5 Used software 5

2 Literature review **7**

2.1 FL 7

2.2 Types of FL 8

2.3 Main challenges in FL 12

2.4 Learning gains in FL 19

2.5 Data encoding techniques 22

3 Proposal **24**

3.1 Formalization 26

3.2 Proposed intervals 26

4 Experimental setup **38**

4.1 Dataset description . . . 38
4.2 Descriptive Statistics and Feature Distributions . . . 43
4.3 Base model . . . 61
4.4 Evaluation metrics . . . 65
4.5 Comparative baselines . . . 67

5 Experimental results **68**
5.1 Initial Considerations . . . 68
5.2 IQR filtering strategy . . . 69
5.3 3SR filtering strategy . . . 74
5.4 CS filtering strategy . . . 78

6 Conclusion and future work **86**
6.1 Conclusion . . . 86
6.2 Future work . . . 87
6.3 Types of intervals and parameter estimation . . . 88
6.4 Security and privacy considerations . . . 88
6.5 Fairness evaluation . . . 89
6.6 Characterizing the nature of removed data . . . 89
6.7 Understanding mechanisms of learning gain . . . 89
6.8 Extending use cases . . . 90
6.9 Testing new models and parametrizations . . . 90

References **91**

7 Appendix **101**
7.1 US State abbreviations . . . 101
7.2 Sent140 Preliminary Assessment . . . 103
7.3 Global data removal statistics per filtering strategy . . . 105
7.4 Global mean and standard deviation from sample statistics . . . 106
7.5 Global quantile estimation from sample quantiles . . . 110
7.6 Results of global quantile estimation . . . 116

7.7 Experiment deployment on Grid'5000 118

List of Figures

1 French Inria centers. Source: Inria (2025b). 5

2 Cross-Device versus Cross-Silo FL. Source: Soudan et al. (2025). 10

3 The different data partition of horizontal FL, vertical FL, and federated transfer learning. Source: Zhang et al. (2021). 11

4 Centralized versus Decentralized FL. Source: Nguyen et al. (2021). 12

5 A taxonomy of challenges and solutions in cross-silo FL. Source: Huang et al. (2022). 13

6 All user devices share a set of base layers with same weights (colored blue) and have distinct personalization layers that can potentially adapt to individual data. The base layers are shared with the parameter server while the personalization layers are kept private by each device. Source: Arivazhagan et al. (2019). 15

7 The attacker queries the target model with a data record and receives a prediction vector of class probabilities. This vector, together with the record's label, is then fed into the attack model, which determines whether the record was part of the target model's training dataset. Source: Shokri et al. (2017). 17

8 LDP FedAvg. Source: Priya (2020). 18

9 Clients first establish encryption keys, either via a threshold protocol or a trusted authority, and use local data to compute sensitivity maps that are homomorphically aggregated into an encryption mask. In the FL stage, this mask enables homomorphic encryption of local updates, allowing the server to aggregate results without accessing sensitive models. Source: Jin et al. (2024). 19

10 Key components associated with an AD technique. Source: Chandola et al. (2009). 20

11 The benchmark model owner (MR) trains a benchmark model and derives its loss distribution (Steps 1–2), while each client computes its own loss values using this model (Step 3). The server then compares client and benchmark loss distributions to set a filtering threshold (Steps 4–5), which clients use to remove noisy samples (Step 6). Source: Tuor et al. (2020). 21

12 Illustration of Part 1 of FedFil. 25
13 Part 2 of FedFil illustrated, with connections to standard FL. 25
14 Representation of the Tukey's fences method. Source: Rabie et al. (2024) 30
15 Schematic representation of a Shewhart control chart with 3-Sigma control limits. Source: Grey (2019). 32
16 Cosine similarity illustrated through three scenarios in $\mathbb{R}^2$. Source: LearnDataSci (nd). 33
17 CD of all cosine similarity values computed between each participant's local data points and the corresponding global mean vector of other participants. 35
18 Client distribution of cosine similarity results below proposed global thresholds. 36
19 Client distribution of cosine similarity results below local 1% and 2.5% percentiles. 37

20 Dataset size per client in the Adult dataset. 41
21 CD of dataset sizes across clients in the Adult dataset. 41
22 Distribution of PINCP across clients in the Adult dataset. 44
23 Proportion of individuals above and below the $50000 income threshold across clients. 44
24 Distribution of the age of individuals, in the dataset. 45
25 Distribution of the age of individuals, per state. 46
26 Distribution of usual hours worked across individuals in the dataset. 46
27 Distribution of usual hours worked across individuals, per state. 47
28 Distribution of biological sex across individuals, per state. 48
29 Distribution of biological sex across individuals in the dataset. 48
30 Race distribution across individuals, per state. 49
31 Race distribution across individuals in the dataset. 49
32 Distribution of place of birth across individuals, per state. 50
33 Distribution of place of birth across individuals in the dataset. 50
34 Distribution of educational attainment across individuals, per state. 51
35 Distribution of educational attainment across individuals in the dataset. 51

36 Distribution of class of worker across individuals, per state. 52
37 Distribution of class of worker across individuals in the dataset. 52
38 Distribution of occupation across individuals, per state. 53
39 Distribution of occupation across individuals in the dataset. 53
40 Distribution of marital status across individuals, per state. 54
41 Distribution of marital status across individuals in the dataset. 55
42 Distribution of relationship to householder across individuals, per state. 55
43 Distribution of relationship to householder across individuals in the dataset. 56
44 Correlation analysis, with frequency encoding applied to POBP and OCCP. 57
45 Correlation analysis, with target encoding applied to POBP and OCCP. 58
46 Proportion of individuals with positive and negative sentiment label, across clients. 60
47 CD of dataset sizes across clients in the Sent140 dataset. 60

48 Data removal proportions under the global IQR interval, showing both the overall distribution (a) and the per-client results (b). 71
49 CD of client-wise filtered local gain with IQR method. Solid lines represent filtered local gain; dashed lines correspond to filtered general gain. Subfigure (a) shows baseline and FedFil with IQR interval; subfigure (b) includes fine-tuning (Ft) for different epochs (ep), applied on FedFil with IQR interval. 72
50 Subfigure (a) shows the maximum local filtered gain for each client using the IQR method; subfigure (b) shows the maximum general filtered gain for each client. The black dash indicates the proportion of each client's local data in the global dataset after IQR filtering. 73
51 Data removal proportions under the global 3SR criterion, showing both the overall distribution (a) and the per-client results (b). 75
52 CD of client-wise filtered local gain with 3SR interval. Solid lines represent filtered local gain; dashed lines correspond to filtered general gain. Subfigure (a) shows baseline and FedFil with global 3SR interval; subfigure (b) includes fine-tuning (Ft) for different epochs (ep), applied on FedFil with global 3SR interval. 76
53 Subfigure (a) shows the maximum local filtered gain for each client using the 3SR method; subfigure (b) shows the maximum general filtered gain for each client. The black dash indicates the proportion of each client's local data in the global dataset after 3SR filtering. 77

54 Data removal proportions under CS (Thr 1%), showing both the overall distribution (a) and the per-client results (b). 79
55 CD of client-wise filtered gains. Baseline and FedFil with cosine similarity (CS) at 1% and 2.5% global removal thresholds (Thr). 80
56 CD of client-wise filtered gains with fine-tuning (Ft) for different numbers of epochs (ep). FedFil with CS (Thr 1%) and CS (Thr 2.5%). 80
57 Subfigure (a) shows the maximum local filtered gain for each client using the CS (Thr 1%) method; subfigure (b) shows the maximum general filtered gain for each client. The black dash indicates the proportion of each client's local data in the global dataset after CS (Thr 1%) filtering. 81
58 Data removal proportions under the CS (Thr 2.5%) criterion, showing both the overall distribution (a) and the per-client results (b). 82
59 Subfigure (a) shows the maximum local filtered gain for each client using the CS (Thr 2.5%) method; subfigure (b) shows the maximum general filtered gain for each client. The black dash indicates the proportion of each client's local data in the global dataset after CS (Thr 2.5%) filtering. 83

60 Sent140 dataset shows poor accuracy, motivating its exclusion from the main experiments. . . . 104
61 Grid'5000 SSH access. Source: Grid'5000 Project Team (2025a). 118
62 Global access machine. 119
63 Sophia access machine. 120
64 Detailed job status. 120
65 Partial node status on the Sophia site. 120

List of Tables

1 Main software packages used in this work, covering data processing, ML, and GPU acceleration. . 6

2 Summary of ACS features used in this work. 40

3 Summary of model architectures and training parameters. 64

4 Statistical analysis of maximum filtered gains 85

5 Global data removal per filtering strategy 106

6 Global quantile estimation for IQR method 117

7 Global quantile estimation for CS method 118

List of Algorithms

1 FedFil: Filtered Federated Learning (comparison with FedAvg). Blue marks steps specific to FedFil. 27

2 IQR filtering for dataset $\mathcal{D}^i$ 30

3 3SR filtering for dataset $\mathcal{D}^i$ 32

4 CS filtering for dataset $\mathcal{D}^i$ 34

List of Acronyms and Abbreviations

3SR 3-Sigma Range

AD Anomaly Detection

AI Artificial Intelligence

CD Cumulative Distribution

CS Cosine Similarity

DP Differential Privacy

FedAvg Federated Averaging

FedFil Filtered Federated Learning

FedPer Federated Learning with Private Personalized Layers

FL Federated Learning

IID Independent and Identically Distributed

INRIA *Institut National de Recherche en Informatique et en Automatique*

IQR Interquartile Range

LDP Local Differential Privacy

ML Machine Learning

SGD Stochastic Gradient Descent

Sent140 Sentiment140

Chapter 1

Introduction

This work was carried out as part of a Master's dissertation in the Mathematics and Computation program at the University of Minho. It arose from a collaboration with Inria (*Institut National de Recherche en Informatique et en Automatique*) Privatics and it was conducted during an internship, continuing a previous project on Federated Learning (FL).

1.1 Motivation

Artificial Intelligence (AI) is increasingly transforming sectors ranging from healthcare to finance and public governance, bringing both significant benefits and societal risks. Despite its potential, AI can also be misused in policy-making, sometimes with severe consequences. For instance, in the *Dutch benefits scandal*, an AI system erroneously labeled 20,000 parents as fraudsters, disproportionately affecting individuals with an immigration background (van Gestel and de Visser, 2021).

Incidents like this highlight the need for careful design and regulation to prevent discrimination and injustice, making it essential that these technologies are trustworthy, transparent, and respectful of fundamental rights (High-Level Expert Group on Artificial Intelligence, 2019).

AI systems rely on large, diverse datasets, often containing sensitive personal information. The collection and processing of such data raise privacy, security, and compliance challenges, including the protection of intellectual property rights, particularly under regulations such as the GDPR (Voigt and von dem Bussche, 2017) and the more

recent EU AI Act (Kelly et al., 2024).

The field of data privacy is well established and diverse, comprising several techniques and methods, namely privacy-enhancing technologies (PETs) (Torra, 2017). In Machine and Deep Learning, many PETs, including Differential Privacy, often sacrifice utility, struggling to preserve accuracy while protecting privacy (Senavirathne and Torra, 2020). Moreover, models can memorize training data and be exploited through attacks such as membership inference (Shokri et al., 2017), model extraction (Tramèr et al., 2016), or direct data recovery (Nasr et al., 2023).

To address these challenges, alternative approaches have emerged. Machine unlearning allows models to selectively remove information, effectively "forgetting" specific data points (Cao and Yang, 2015). Federated Learning (FL) enables collaborative model training across distributed clients (or participants) without sharing raw data, reducing exposure of sensitive information and enabling scalable, efficient training (McMahan et al., 2017). This is particularly relevant in sectors such as healthcare, where existing medical data, although abundant, remains underexploited (Rieke et al., 2020).

FL is rapidly gaining popularity and generating significant expectations from industry: the global FL market increased from USD 166.34 million in 2024 to USD 192.71 million in 2025 and is projected to continue expanding at an average annual growth rate of 15.66%, reaching USD 532.90 million by 2032 (Research and Markets, 2025). FL is intended to enable collaborative training of Machine Learning (ML) models either across different organizations (cross-silo FL, e.g., bio or medical centers) or across numerous user devices (cross-device FL, e.g., mobile phones). Despite its promise, FL still faces a variety of challenges, including privacy and security limitations (Huang et al., 2022; Kairouz et al., 2021), which pose serious barriers to broad adoption. Ensuring the confidentiality of information and the integrity of the system is crucial not only for privacy but also for protecting the competitive insights of participating organizations, particularly in cross-silo scenarios.

1.2 Goals

Building on these challenges and opportunities, this work focuses on understanding the incentives and learning gains for participants in cross-silo FL. The primary incentive for engaging in cross-silo FL is the potential for a learning gain: the global model trained collaboratively can achieve better accuracy and generalization than a model trained locally using only the participant's own data. Without such a gain, participants have little motivation to adopt FL. Notably, local training alone does not face these privacy constraints, since no data or model information is shared.

Evidence of learning gains has been observed in medical use cases involving multiple participants (hospitals) with small local datasets (Heyndrickx et al., 2024; Pati et al., 2022).

However, it remains unclear whether similar gains can be achieved in other domains. This work therefore conducts a thorough empirical investigation into the learning gains achievable in FL, a topic frequently overlooked in the state-of-the-art literature (Liu et al., 2021; Zhang et al., 2022), where the main focus is on privacy and security threats, heterogeneity challenges, and communication overhead, and is only briefly mentioned in some studies (Wen et al., 2024).

More specifically, this project aims to address the following research questions:

- What is the learning gain for participants engaged in a cross-silo FL setting?
- Which clients benefit more from using FL?
- Is it possible to change locally the learning scheme to improve global accuracy gain?

1.3 Structure of dissertation

Chapter 1 outlines the motivation and goals of this dissertation. It also presents Inria, where this work was carried out, with a focus on the Privatics team, describing its background, research areas, and relevance. The chapter concludes by introducing the software stack used throughout the experimentation, providing a detailed breakdown of its main components and their roles.

Chapter 2 presents the literature review on Federated Learning (FL), covering its variants, main challenges, and the notion of learning gain. It also discusses relevant aspects of Anomaly Detection (AD) that underpin the construction of the filtering methods and concludes with a survey of data encoding techniques occasionally required in this work.

Chapter 3 provides an initial outline of FedFil, the FL scheme proposed and analysed in this dissertation.

Chapter 4 describes the datasets used, one tabular and one text-based, enumerating their features, meanings, sizes, and modelling objectives. An exploratory statistical analysis is presented, including the distribution of features across FL participants. The chapter then introduces the local models used by each participant in each dataset, their parametrizations and full model specifications. It concludes by formally defining FedFil, its evaluation metrics, the proposed filtering methods, and other reference FL schemes used for comparison.

Chapter 4 presents the central contribution of this dissertation: the evaluation of FedFil on a tabular dataset using the proposed filtering intervals. (The text-based dataset was ultimately discarded.)

Chapter 6 summarizes the work and outlines several promising directions for future research.

Chapter 7 comprises supplementary results and the formal proofs pertaining to the proposed methods. Furthermore, it includes auxiliary material, such as the nomenclature of U.S. states and a brief description of a software routine employed in this work.

1.4 Presentation of Inria Privatics

INRIA (*Institut National de Recherche en Informatique et en Automatique*) is a French national research institute created in 1967 at Rocquencourt (Inria, 2025c). Since its foundation, it has played a central role in advancing research in computer science, applied mathematics, and related disciplines, both in France and internationally. Today, INRIA is a leading institution in AI research. Alongside ETH Zurich, it ranked as the top European institution in number of accepted publications at NeurIPS 2024 (Singh, 2025), one of the most prestigious venues in the field.

INRIA operates several research centres across France, represented in Figure 1, as well as one in Santiago, Chile. Its scientific activity is structured around specialised project teams, among which is **Privatics**.

Privatics was created in 2014, being present in the Inria Lyon and Grenoble centres, being dedicated to privacy protection in the digital world. Its work is multidisciplinary, combining computer science with legal, economic, sociological and ethical perspectives. The team's activities are structured into four axes (Inria Privatics Team, 2025):

- **AI**: analysing how ML impacts privacy, and designing privacy-enhancing techniques for ML systems;
- **Web, Mobile, IoT and Network Privacy**: studying privacy risks arising from web services, smartphones, IoT (Internet of Things) devices and wireless networks, along with mitigation strategies;
- **User Empowerment**: understanding how users can maintain control over their personal data, with a focus on consent, transparency and deceptive design practices;
- **Legal and Regulatory Analysis**: evaluating the compliance of technologies with EU data protection law and examining the legal implications of emerging privacy-invasive systems.

Privatics has contributed to several notable initiatives, including a contact-tracing protocol during the COVID-19 pandemic (Boutet et al., 2021), deployed in the most-downloaded app in France in 2021 (RFI, 2022), and a report to the European Parliament on algorithmic decision-making (Castelluccia and Le Métayer, 2019). INRIA is also highly active in Federated Learning research (Jourdan et al., 2021; Kairouz et al., 2021). Notably, the Fed-Malin project brings together eleven project teams and six research centres to address challenges in large-scale FL deployments, including privacy and fairness, energy consumption, personalization, and spatiotemporal dynamics (Inria, 2025a).

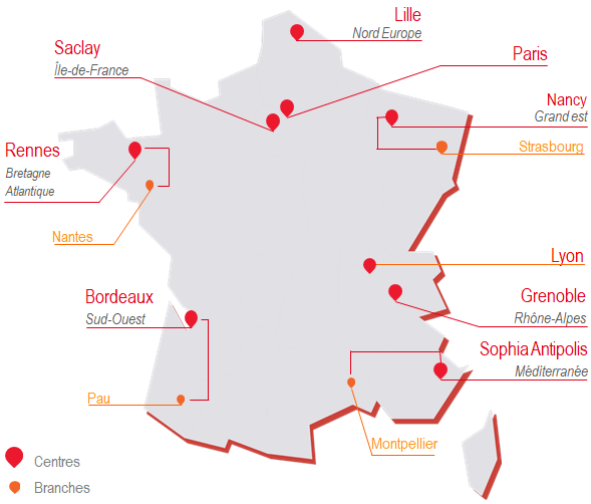


Figure 1: French Inria centers. Source: Inria (2025b).

1.5 Used software

Most of the experimentation conducted for this work was performed using Grid'5000, a large-scale and flexible testbed for experiment-driven research in all areas of computer science, with a focus on parallel and distributed computing, including AI. Grid'5000 has more than 800 compute nodes, around 25,000 cores, and more than 550 GPUs distributed across 11 sites (Grid'5000 Project Team, 2025b).

Section 7.7 provides an walkthrough of how the experiments were deployed on Grid'5000, including example job submissions and available resources.

Within Grid'5000, a virtual environment was created and Python 3.9.2 was installed. Table 1 lists the most relevant Python packages used in this dissertation, focusing on those directly involved in data processing, modelling,

experiments, and GPU execution. In addition to Python, Microsoft Excel was used for preliminary inspection of results, and RStudio for statistical analyses that were eventually not included in this document.

Package	**Version**	**Usage**
folktables	0.0.12	Importing Adult dataset
numpy	1.26.3	Numerical computing, arrays
scipy	1.12.0	Scientific computing
pandas	2.2.0	Data manipulation and analysis
matplotlib	3.8.2	Data visualization and plotting
joblib	1.3.2	Parallelization and model persistence
psutil	7.0.0	System monitoring (CPU, memory usage)
tqdm	4.67.1	Progress bars for loops and training
tensorflow	2.15.0	Deep learning framework
keras	2.15.0	High-level API for TensorFlow
scikit-learn	1.6.1	Classical ML algorithms
h5py	3.10.0	HDF5 file format support (model storage)
argparse	builtin	Command-line argument parsing
pickle	builtin	Loading and saving accuracy results in .sav files
nltk	3.8.1	Natural language processing toolkit
regex	2024.11.6	Advanced regular expressions for text processing
NVIDIA CUDA/cuDNN stack	12.x	GPU acceleration for large-scale computation

Table 1: Main software packages used in this work, covering data processing, ML, and GPU acceleration.

The relevant code developed in this project is publicly available at https://github.com/d-m-silva/dissertation_project.

Chapter 2

Literature review

2.1 FL

Machine learning (ML) is a field of AI in which models learn patterns from data to make predictions or decisions (Goodfellow et al., 2016). Federated Learning (FL) builds on this idea but trains such models collaboratively across multiple clients without centralizing their data. The term *federated learning* was introduced in 2017 by McMahan et al. (2017), describing a setting in which the learning task is solved by a loose federation of participating devices (or clients) coordinated by a central server. An unbalanced and non-IID (identically and independently distributed) data, partitioning across a massive number of unreliable devices with limited communication bandwidth was introduced as the defining set of challenges.

While the original formulation focused primarily on mobile and edge-device settings, interest in applying FL to broader scenarios has greatly increased, including applications involving only a small number of relatively reliable clients—for example, multiple organizations collaborating to train a model. A broader and now widely adopted definition of FL was later given by Kairouz et al. (2021): "Federated Learning is a machine learning setting where multiple entities (clients or participants) collaborate in solving a machine learning problem, under the coordination of a central server or service provider. Each client's raw data is stored locally and not exchanged or transferred; instead, focused updates intended for immediate aggregation are used to achieve the learning objective." Focused updates are narrowly scoped to contain the minimum information necessary for the specific learning task, with aggregation performed as early as possible to support data minimization.

FL can be formally described as an optimisation problem, in which a set of clients collaboratively minimise a

global objective function without sharing raw data. Let $\mathcal{D}_i$ denote the local dataset of client $i \in \{1, \ldots, K\}$, with $n_i = |\mathcal{D}_i|$ samples and $N = \sum_{i=1}^{K} n_i$ the total number of samples across all clients. The goal is to learn model parameters $W \in \mathbb{R}^d$ that minimise the empirical risk over all clients:

$$W^\star = \arg\min_W \; F(W) \quad \text{with} \quad F(W) = \sum_{i=1}^{K} \frac{n_i}{N} \mathcal{L}(W; \mathcal{D}_i),$$

where $\mathcal{L}$ denotes the local loss function (e.g., cross-entropy).

Training proceeds iteratively in communication rounds. At round t, the server broadcasts the current global model W^{t-1} to a subset of clients. Each selected client performs several steps of stochastic gradient descent (SGD) on its local data:

$$W_i^{t,k+1} = W_i^{t,k} - \eta \nabla \ell(W_i^{t,k}; \xi_i^{t,k}), \tag{2.1}$$

where η is the learning rate, $\xi_i^{t,k}$ is a minibatch sampled from $\mathcal{D}_i$, k indexes the local SGD steps, and $\nabla \ell(\cdot)$ denotes the gradient of the loss function with respect to the model parameters.

The server then aggregates the received updates—typically through weighted averaging, as in Federated Averaging (FedAvg) (McMahan et al., 2017):

$$W^t \; = \; \sum_{i=1}^{K} \frac{n_i}{N} W_i^t.$$

This aggregated model is redistributed to clients in the next round.

Given the diversity of deployment scenarios and data constraints, it is useful to delineate the main variants of FL, which are presented in Section 2.2.

2.2 Types of FL

FL can adapt to various data distribution scenarios and client setups to tackle distinct challenges. According to Mothukuri et al. (2021), these adaptations are categorized based on:

- data availability: how data is distributed across clients (**cross-silo**, **cross-device**);
- data partitioning: the relationship between client data in terms of users and features (**horizontal**, **vertical**,

transfer Learning);

- network topology: how clients are organized and interact during the training process (**centralized**, **decentralized**).

2.2.1 Adaptations based on data availability

While FL is often contrasted with traditional centralized machine learning, where all the data is in a centralized server or data center for training, a more nuanced comparison requires distinguishing FL from other distributed learning paradigms. In particular, three settings are commonly discussed in the literature: *distributed learning*, *cross-silo FL*, and *cross-device FL*. Although these settings share the high-level objective of training a model across multiple parties, they differ substantially in data distribution, orchestration, client availability, and system bottlenecks (Subasi et al., 2023).

Differences between distributed learning and FL. Distributed learning assumes that data is centrally stored and can be freely shuffled and balanced across clients. Any client can access any part of the dataset, and orchestration is fully centralized. Clients are typically reliable, fully addressable and computation is usually the main bottleneck due to high-speed interconnects. By contrast, FL operates on decentralized data that **cannot be shared among clients**. A central server coordinates training without **ever accessing raw data**, and communication, client availability, and heterogeneity become critical factors to consider (Kairouz et al., 2021).

Cross-silo versus cross-device FL. Within FL itself, two main deployment scenarios are distinguished based on client characteristics and scale:

- **Cross-silo FL:** Involves a relatively small number of reliable clients (typically 2–100), such as organizations or geo-distributed datacenters. Data is generated locally and remains decentralized. Partitioning can be horizontal (example-based) or vertical (feature-based). Clients participate consistently and dropout rates are low.

- **Cross-device FL:** Involves a very large number of heterogeneous and unreliable clients (up to 10^{10}) (Kairouz et al., 2021), such as mobile or IoT devices. Each client generates and stores its own data, which is non-IID and cannot be accessed by others. Partitioning is typically horizontal, clients are mostly stateless, and

dropout rates are high due to connectivity, battery, or availability constraints. Communication bandwidth is often the primary bottleneck.

Figure 2 shows a schematic representation of cross-silo and cross-device FL.

In summary, while distributed learning emphasizes centralized data and reliable clients, FL must account for heterogeneous, and often unreliable clients. Within FL, cross-silo and cross-device settings differ mainly in scale, client reliability, and the type of partitioning they support. These distinctions naturally motivate a deeper discussion on data partitioning, which is the focus of the following section 2.2.2.

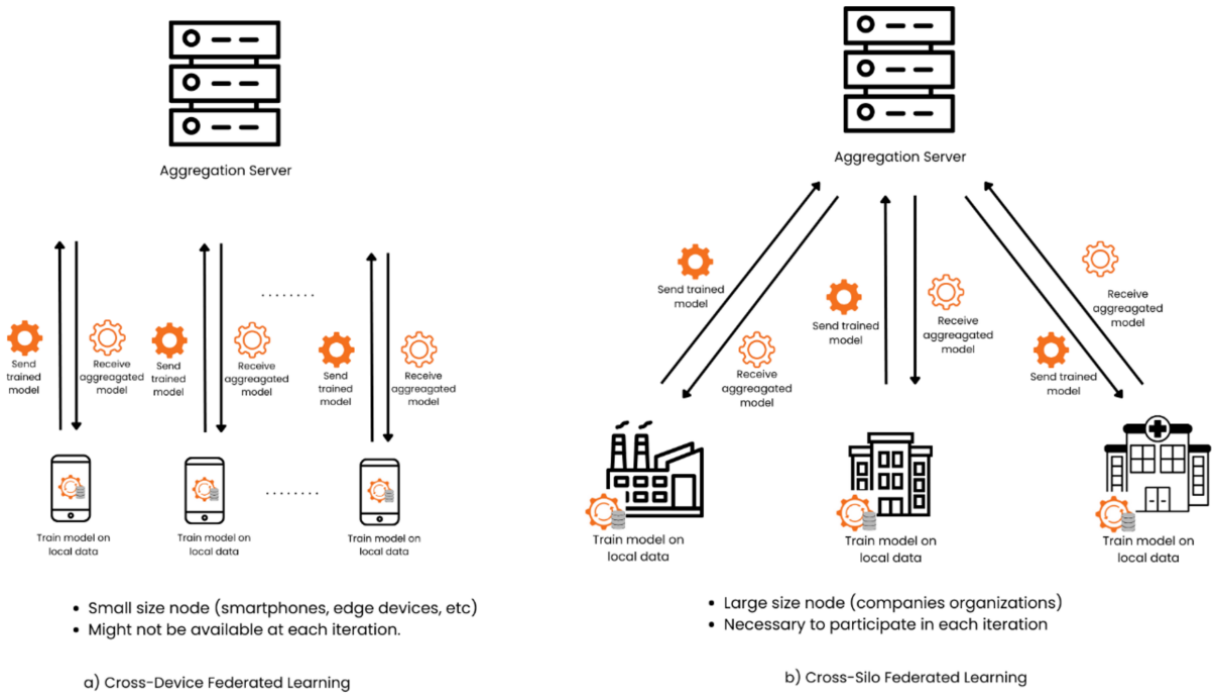


Figure 2: Cross-Device versus Cross-Silo FL. Source: Soudan et al. (2025).

2.2.2 Adaptations based on data partitioning

In the cross-device setting the data is assumed to be partitioned by examples. In the cross-silo setting, in addition to partitioning by examples, partitioning by features is of practical relevance. An example could be when two companies in different businesses have the same or overlapping set of customers, such as a local bank and a local retail company in the same city. This difference has been also referred to as horizontal and vertical FL by Zhang et al. (2021).

Federated transfer learning (Yang et al., 2019) is another concept that considers challenging scenarios in which data parties share only a partial overlap in the user space or the feature space, like labels, and leverage existing transfer learning techniques (Pan and Yang, 2010) to build models collaboratively. Figure 3 illustrates these data partitioning settings.

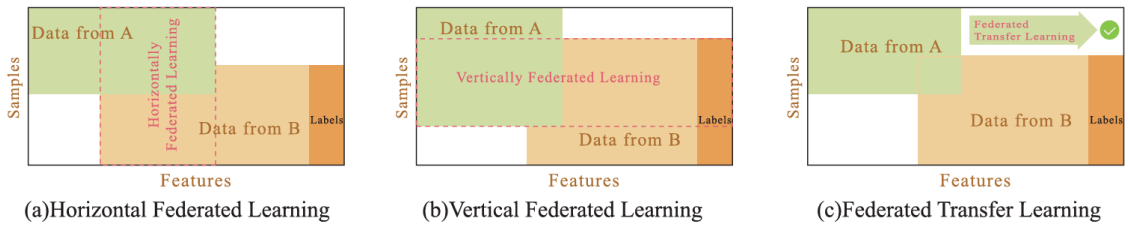


Figure 3: The different data partition of horizontal FL, vertical FL, and federated transfer learning. Source: Zhang et al. (2021).

2.2.3 Adaptations based on network topology

Having considered how FL varies with data availability and data partitioning (Sections 2.2.1 and 2.2.2), we now turn to another key dimension: the network topology governing client interactions.

In centralized FL, a central server orchestrates the training process and aggregates contributions from all clients. As the central coordinator, the server represents a potential single point of failure. While large organizations can fulfill this role in some applications, a reliable central server may not always be available or desirable in more collaborative learning scenarios (Vanhaesebrouck et al., 2017).

Fully decentralized learning addresses this limitation by replacing the central server with peer-to-peer (P2P) communication among clients. The communication topology is represented as a connected graph, where nodes correspond to clients and edges indicate communication channels between them. In each round, clients perform local updates and exchange information with their neighbors (Nguyen et al., 2021).

Unlike centralized FL, there is no single global model; instead, algorithms are designed to ensure that local models converge to a consensus solution over time (Vanhaesebrouck et al., 2017). Even in decentralized settings, a central authority may still coordinate the learning task, or it can be proposed by a client or agreed upon collaboratively through a consensus scheme (Nguyen et al., 2021).

Figure 4 presents a schematic comparison between centralized and fully decentralized FL, highlighting the role of the central server in the former and the peer-to-peer communication structure of the latter.

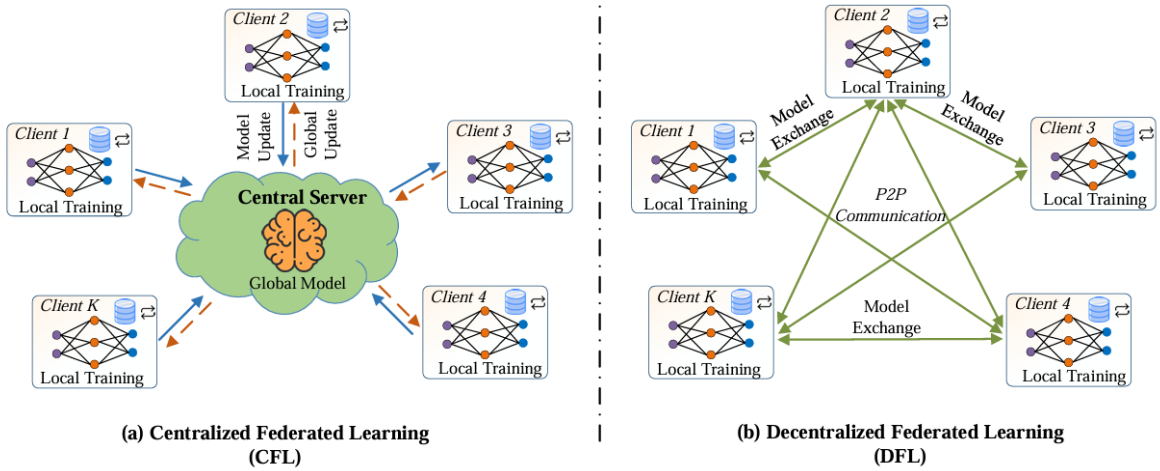


Figure 4: Centralized versus Decentralized FL. Source: Nguyen et al. (2021).

2.3 Main challenges in FL

While both cross-device and cross-silo FL share common concerns, the relative importance of each challenge differs.

In this work, we focus exclusively on cross-silo FL, and therefore its specific challenges are addressed in greater detail.

According to Huang et al. (2022), three major categories of challenges are commonly identified in cross-silo FL:

- **Effectiveness and Efficiency**: effectiveness refers to obtaining satisfactory global (and local) models that account for client heterogeneity, while efficiency concerns achieving such models rapidly and at low cost;
- **Privacy and Security**: privacy relates to protecting clients' local data from leakage, whereas security involves detecting and preventing adversaries from compromising the training process or degrading model performance;
- **Cooperation and Incentives**: unlike cross-device FL, organizations in a cross-silo setting usually pursue clear long-term strategic goals, making cooperation and incentive mechanisms crucial.

Figure 5 summarizes the taxonomy of challenges in cross-silo FL.

Among these, **effectiveness**, **privacy**, and **cooperation and incentives** are considered the major challenges. In this work, the component **cooperation and incentives** is less prevalent and will therefore be discussed in less detail.

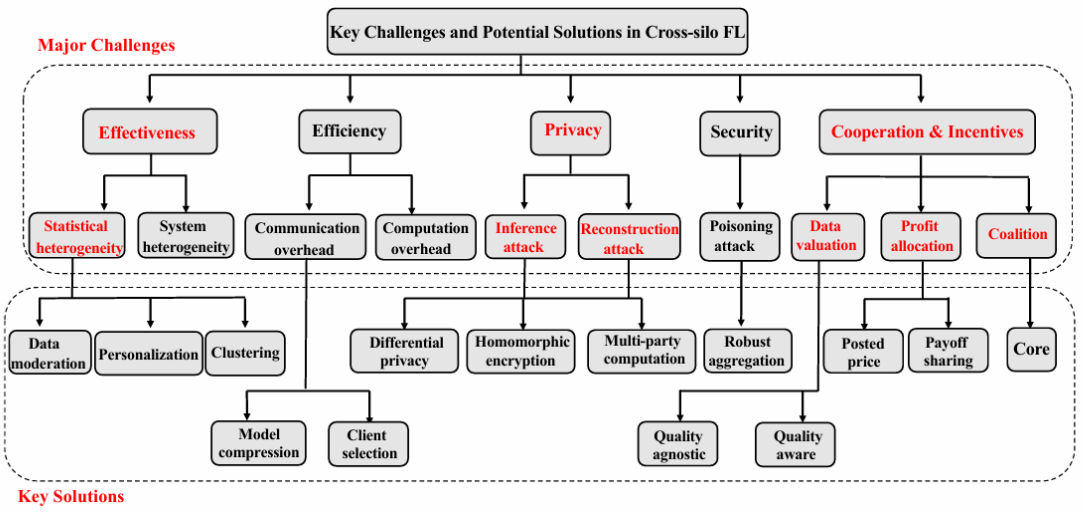


Figure 5: A taxonomy of challenges and solutions in cross-silo FL. Source: Huang et al. (2022).

2.3.1 Effectiveness

Two components crucially affect the effectiveness of FL: **statistical heterogeneity** and **system heterogeneity**.

Statistical heterogeneity

Clients' data are often non-IID (Wei et al., 2024). This can be understood in terms of violations of independence and identicalness (Hsieh et al., 2020):

- Violations of **independence**: For example, hospitals may process patient data in a time-ordered or location-ordered manner, and devices within the same region may generate correlated data.
- Violations of **identicalness**: For instance, hospitals serve patients with different age distributions and demographic profiles, and banks have customers with diverse economic backgrounds; moreover, the number

of samples can vary widely across institutions.

Existing studies have developed three main approaches (Zhu et al., 2021), including:

- **Data moderation approaches**: modify or augment local data distributions, for example by using small shared datasets or applying data augmentation informed by other clients;
- **Personalization approaches**: adapt the global model to local tasks through:
  - *local fine-tuning*, which refines local models using local data after receiving the global model;
  - *personalized layers*, in which clients' local models consist of base and personalized layers, with only the base layers uploaded for aggregation;
  - *knowledge distillation*, which transfers knowledge from the server (or other clients) to a specified client to enhance its local model accuracy on heterogeneous data.
- **Clustering approaches**: group clients into clusters with similar distributions, maintaining separate sub-models per cluster, where similarity is measured by loss values or uploaded model weights (Ouyang et al., 2021).

Among personalization approaches, notable examples include:

- **PerFedAvg (Personalized Federated Learning)** (Fallah et al., 2020): This method first finds an initial shared model that can be easily adapted locally by each participant. Each client then performs one or a few steps of gradient descent on their own data to obtain a personalized model.
- **FedPer (Private Personalized Layers)** (Arivazhagan et al., 2019): In this approach, the global aggregation process is applied exclusively to the common base layers of the model, while the upper private layers remain local to each client. During training, each client performs SGD (stochastic gradient descent) (cf. equation 2.1) updates on all layers, but only uploads the base layers to the central server. This allows clients to retain personalized components that capture individual data characteristics without sharing them.
- **FedSelect (Personalized Federated Learning with Customized Selection of Parameters for Fine-Tuning)** (Tamirisa et al., 2024): FedSelect personalizes client parameters while performing global aggregations on the remaining parameters. This method enables the personalization of both client-specific parameters and subnetwork structure during training.

Figure 6 provides a pictorial illustration of the FedPer approach.

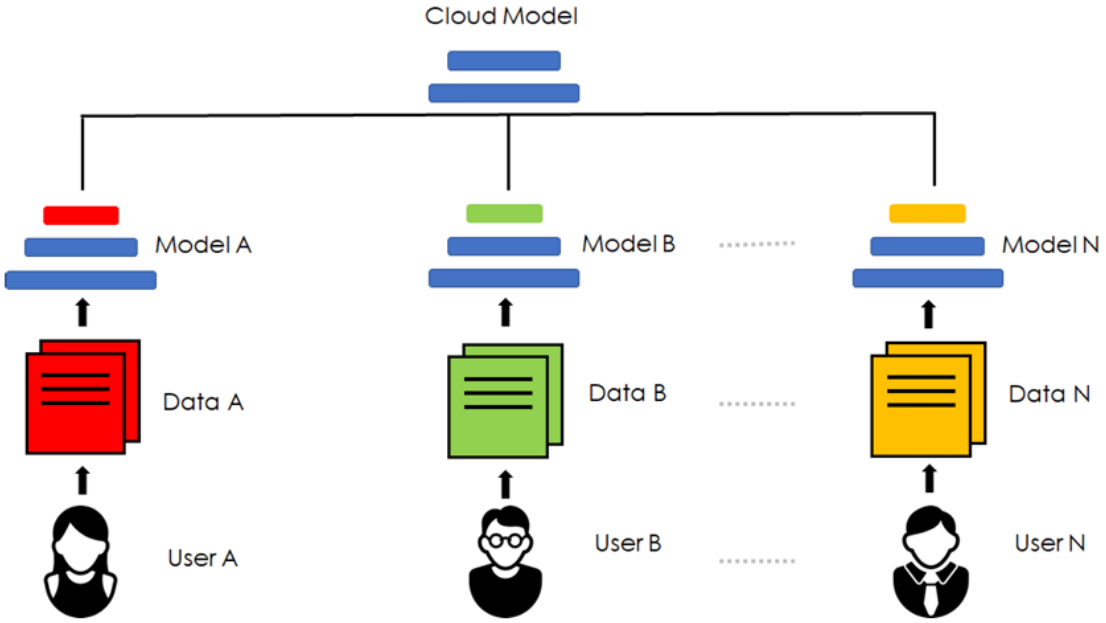


Figure 6: All user devices share a set of base layers with same weights (colored blue) and have distinct personalization layers that can potentially adapt to individual data. The base layers are shared with the parameter server while the personalization layers are kept private by each device. Source: Arivazhagan et al. (2019).

System heterogeneity

In cross-silo FL, organizations typically have stable infrastructure, sufficient computation, and reliable network connections. Therefore, system heterogeneity is usually not a major concern. This is because cross-silo FL applications can have a more stringent requirement in terms of model performance than cross-device scenarios. For example, diagnosis models trained by hospitals are expected to have a sufficiently high accuracy. In cross-device setting, a next-word prediction model can have a relatively lower accuracy requirement since the wrong predictions would cause less harm than improper medical treatment.

Several approaches have been proposed to mitigate both statistical and system heterogeneity (Li et al., 2021b). Notable examples include FedProx (Li et al., 2020b), FedNova (Wang et al., 2020), and SCAFFOLD (Karimireddy et al., 2020), all of which extend FedAvg to stabilize convergence under non-IID settings. For example, such methods may allow selected clients to perform a variable amount of local computation, thereby accommodating underlying system constraints (Li et al., 2020a).

2.3.2 Privacy

Despite not sharing raw data, FL is not immune to attacks nor the current developments in its enabling technology are mature enough to be expected to solve all privacy issues by default. It is subject to various inference attacks (Mothukuri et al., 2021):

- **Membership inference attacks**: determine whether a specific data point was part of the training set;
- **Reconstruction attacks**: attempt to infer information about training data from model updates.

Figure 7 shows a high level example of an end-to-end membership inference attack. Reconstruction attacks can happen in a scenario where updates or gradients from clients leak unintended information at the central server. The authors in Melis et al. (2019) exploit unintentional data leakage vulnerability and successfully reconstruct data of other clients through an inference attack. Malicious/curious participants in (Bhowmick et al., 2018) use the global model and parameters to reconstruct the training data of other clients.

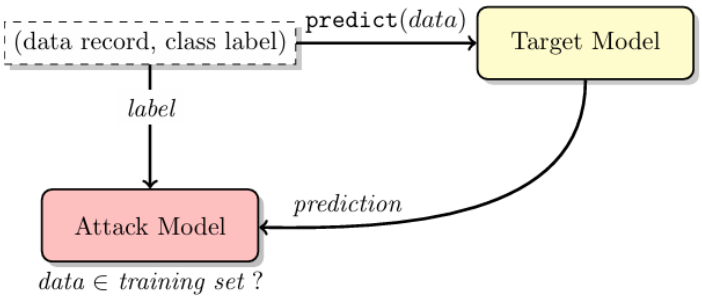


Figure 7: The attacker queries the target model with a data record and receives a prediction vector of class probabilities. This vector, together with the record's label, is then fed into the attack model, which determines whether the record was part of the target model's training dataset. Source: Shokri et al. (2017).

Privacy protection

To mitigate these risks, several privacy-preserving strategies are employed, particularly **Differential Privacy (DP)** (Dwork and Roth, 2014), a widely adopted privacy-preserving technique in both academia and industry. The basic idea is to add calibrated random noise to sensitive data, providing a mathematically rigorous guarantee that the contribution of any individual data point remains difficult to detect. Differential privacy is quantified by parameters (ϵ, δ), usually called "privacy budget", where smaller values corresponds to stronger privacy.

More formally, a randomized algorithm $\mathcal{M}$ satisfies (ϵ, δ)-DP if, for all adjacent datasets D, D' and all outputs $S \subseteq M$:

$$\Pr[\mathcal{M}(D) \in S] \leq e^{\epsilon} \Pr[\mathcal{M}(D') \in S] + \delta. \tag{2.2}$$

In the FL setting (Kairouz et al., 2021), D and D' are typically considered adjacent under a **client-level** notion: D' is obtained from D by adding or removing all data belonging to a single client (McMahan et al., 2018).

Two main variants of DP are commonly used in FL (Kasiviswanathan et al., 2011):

- **Central Differential Privacy (CDP)**: a trusted server applies the DP mechanism to the aggregated client updates it receives (e.g., adding noise after aggregation);
- **Local Differential Privacy (LDP)**: each client independently applies a DP mechanism before communicating any information to the server, thus requiring no trusted aggregator.

LDP provides protection against inference attacks by ensuring that shared updates already contain sufficient noise to hide an individual user's contribution (Sun et al., 2022). A common LDP technique in FL is DP-SGD (Naseri et al., 2020), where each client clips local gradients and adds Gaussian noise at every step before transmitting updates. Figure 8 illustrates LDP FedAvg using DP-SGD.

DP often sacrifices utility, struggling to preserve accuracy while protecting privacy (Senavirathne and Torra, 2020). Among the many privacy-preserving strategies proposed in the literature, two widely adopted approaches that typically incur a smaller utility loss are:

- **Homomorphic Encryption (HE)** (Gentry, 2021): enables certain operations (e.g., addition) to be performed directly on encrypted data without requiring decryption. In cross-silo FL, each client encrypts its local model updates before sending them to the server, which aggregates them homomorphically; the aggregated ciphertext is then returned to the clients for local decryption (Jin et al., 2024). HE preserves model accuracy but introduces substantial computational and communication overhead. Figure 9 illustrates the pipeline of FedML-HE, an HE-based privacy-preserving FL framework.

- **Secure Multi-Party Computation (MPC)** (Chaum et al., 1987): enables clients to collaboratively compute aggregated values without exposing their individual contributions. This provides strong privacy guarantees but also increases computation and communication costs (Kairouz et al., 2021). SMPAI (Mugunthan et al., 2019) proposes combining MPC with DP to further reduce the risk of sensitive information leakage.

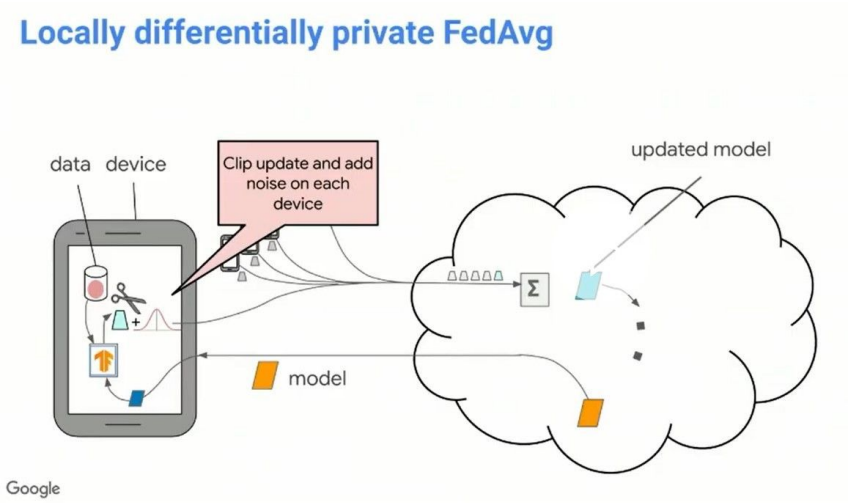


Figure 8: LDP FedAvg. Source: Priya (2020).

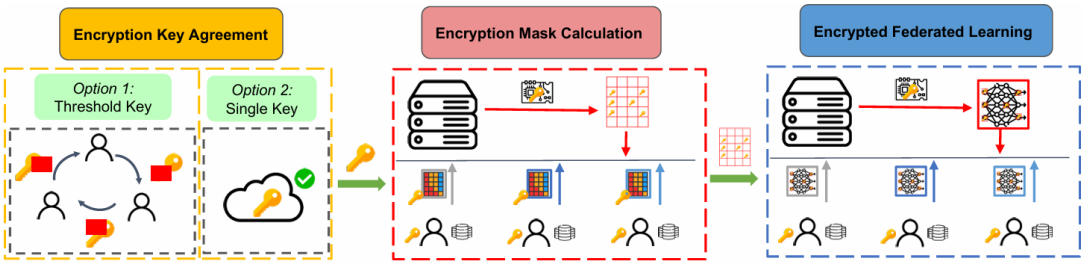


Figure 9: Clients first establish encryption keys, either via a threshold protocol or a trusted authority, and use local data to compute sensitivity maps that are homomorphically aggregated into an encryption mask. In the FL stage, this mask enables homomorphic encryption of local updates, allowing the server to aggregate results without accessing sensitive models. Source: Jin et al. (2024).

These techniques allow FL to achieve strong privacy guarantees while leveraging the stable infrastructure and computational resources of participating organizations.

2.4 Learning gains in FL

While the state-of-the-art literature heavily emphasises privacy, security, heterogeneity and communication constraints, the question of **learning gain**, whether participants actually benefit from using FL compared to training solely on their own data using classical machine learning methods, remains underexplored, as discussed in Section 1.2. Among the various metrics used to evaluate ML performance (Blagec et al., 2020), one of the most fundamental is model accuracy.

Model accuracy can be significantly hampered by the presence of discrepant data. Discrepant data, more formally referred to as outliers or anomalies, correspond to observations that deviate from the dominant characteristics of the dataset (Sikder and Batarseh, 2021). Although the concept of anomaly and outlier are sometimes distinguished, anomalies are often seen as a special case of outliers and outliers are more related with the statistical community, the terms are frequently used interchangeably (Chandola et al., 2009).

The problem of identifying such observations has a long history, dating back to statistical work in the nineteenth century (Edgeworth, 1887). Since then, a broad landscape of anomaly detection (AD) techniques has emerged across domains and applications. Figure 10 summarises the main components, challenges, and methodological families within AD.

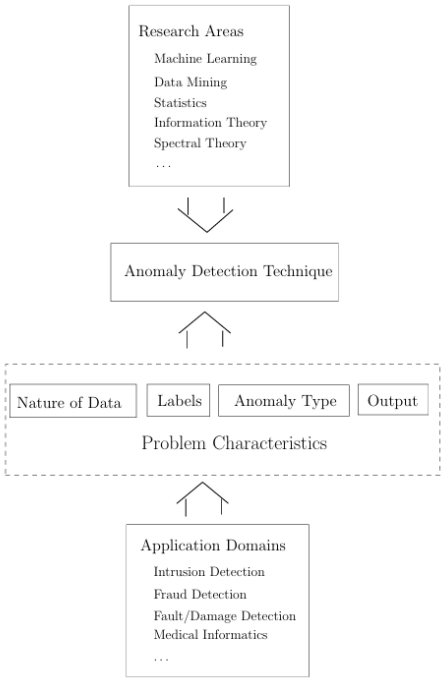


Figure 10: Key components associated with an AD technique. Source: Chandola et al. (2009).

AD techniques can be categorised according to the type of anomaly they target, the availability of data labels, and the nature of their output (Chandola et al., 2009):

- **Type of anomaly**:
  - **point anomaly**: an individual instance deviates strongly from the global data distribution;
  - **context anomaly**: an instance is anomalous only under specific contextual conditions;
  - **collective anomaly**: a group of individually normal instances forms an anomalous pattern when considered jointly (e.g., unusual time-series segments).
- **Data labels**:
  - **supervised**: AD relies on labelled normal and anomalous instances – effective but limited by class imbalance and scarcity of anomaly labels;
  - **unsupervised**: AD finds deviations without labels, assuming anomalies are rare, being sensitive to violations of this assumption.
- **Output**:

- **scores**: each instance receives a continuous anomaly score, enabling ranking or threshold selection;
- **labels**: each instance is classified directly as normal or anomalous.

In this work, **unsupervised, point-anomaly, score-based** detection methods are employed to filter the client datasets, being discussed with more detail in Section 3.2, which are then tested with results coming from centralized settings, on Chapter 5. In FL, to the best of our knowledge, Tuor et al. (2020) and Li et al. (2021a) are the only papers addressing data filtering on clients' datasets, but from different perspectives. Tuor et al. (2020) proposes that each client filters and selects relevant data before training using a small benchmark dataset. Each client evaluates the loss of each data sample using the benchmark model, and by comparing the distribution of these losses with that of the benchmark dataset, a threshold is determined. Samples with a loss above this threshold are considered noisy and excluded from training. Figure 11 summarizes the overall data selection process.

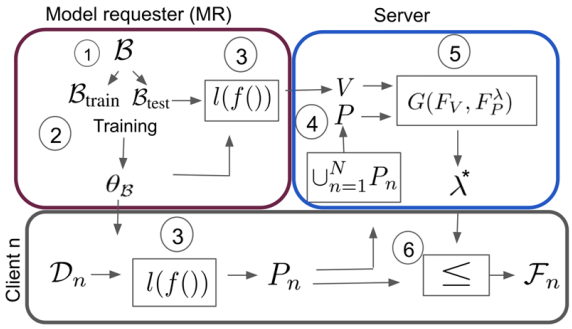


Figure 11: The benchmark model owner (MR) trains a benchmark model and derives its loss distribution (Steps 1–2), while each client computes its own loss values using this model (Step 3). The server then compares client and benchmark loss distributions to set a filtering threshold (Steps 4–5), which clients use to remove noisy samples (Step 6). Source: Tuor et al. (2020).

They report learning gains compared to a baseline that uses all data. Their method seems better suited for cross-device FL, as they perform experiments with more than 100 participants, whereas cross-silo settings typically involve fewer clients (cf. Section 2.2.1). The approach by Li et al. (2021a) is more comprehensive, evaluating data quality both before and during training to select high-quality samples for the target task, incentivizing clients to participate under a budget in a privacy-preserving manner. Like the previous method, it does not compare to classical learning.

2.5 Data encoding techniques

Many mathematical methods require numerical inputs, emphasizing the need for effective preprocessing of categorical variables. Since some of the experimental results rely on methods requiring numerical inputs, and the dataset has mostly categorical features, a review of commonly used techniques is included. According to (Ouahi et al., 2024), encoding techniques can be divided into simple and advanced; in this work, only simple techniques are used.

Following (Ouahi et al., 2024), the main simple encoding techniques are:

- **Binary encoding**: converts categories into binary digits.
  - Example: for categories {A, B, C, D}, map them to {1, 2, 3, 4} and convert to binary: A $\rightarrow$ 01, B $\rightarrow$ 10, C $\rightarrow$ 11, D $\rightarrow$ 100.
  - It reduces dimensionality compared to one-hot encoding (number of columns grows as $\log_2(n)$ versus n), but may reduce interpretability.
- **Frequency (or Count) encoding**: replaces each category with its relative or absolute frequency.
  - Example: for categories {A, B, B, A, A}, the relative frequencies are: A $\rightarrow$ 3/5, B $\rightarrow$ 2/5.
  - It's simple and low-dimensional, but can leak target information if not used carefully.
- **Target (or Mean) encoding**: replaces each category with the mean of the target variable for that category.
  - Example: consider categories {A, B, B, A, A} with a binary target {1, 0, 1, 1, 0}, then A $\rightarrow 2/3$, B $\rightarrow 1/2$.
  - It can improve predictive power, but risks overfitting if categories have few samples.
- **One-hot encoding**: creates a binary indicator column for each category.
  - Example: for categories {Low, Medium, High}, Low becomes (1,0,0), Medium becomes (0,1,0), High becomes (0,0,1).
  - It's interpretable and widely used, but dimensionality grows linearly with the number of categories (n), which can become large for high-cardinality features.

- **Label encoding**: assigns a unique integer to each category.
  - Example: {Low, Medium, High} is assigned {1, 3, 2}.
  - It's compact and simple, but may introduce unintended ordinal relationships.
- **Ordinal encoding**: maps ordered categories to ordered integers.
  - Example: {Low, Medium, High} is assigned {1, 2, 3}.
  - It preserves order information, but assumes equal spacing between categories which may not be accurate.

These methods were employed in Sections 4.2.4 and 3.2.4, where the first analyses correlations between features, and the second computes cosine similarity.

Chapter 3

Proposal

As stated in Section 1.2, one of the main objectives of this work is to study how the removal of a particular set of discrepant data (namely outliers) affects performance.

To address the challenges in achieving significant learning gains for individual participants in cross-silo FL, this work proposes a novel two-stage learning scheme: **FedFil** (Filtered Federated Learning). FedFil is designed to improve the accuracy of the final personalized model for each client, by addressing issues prior to collaborative training and improving global accuracy gain. On the other hand, since removing datapoints may reduce accuracy, FedFil also ensures that all the data at each client is eventually used again. It can be divided into two parts:

- **Part 1: Local Data Filtering.** Before participating in the collaborative FL training rounds, each client applies an unsupervised, score-based Anomaly Detection (AD) technique to their local dataset. This process identifies and removes influential and anomalous datapoints that might otherwise degrade the quality of the local updates and, consequently, the final global model. The subsequent FL training proceeds only on these curated local datasets.

- **Part 2: Post-FL Fine-Tuning.** Upon receiving the converged global model from the FL server, each client performs a short round of local fine-tuning. This personalization step adapts the general knowledge embedded in the global model to the specific characteristics of the client's already filtered local data, aiming to further improve the client-specific learning gain.

Figures 12 and 13 show schematic views of what constitutes FedFil, using two clients. In Figure 12, the first part of FedFil is shown. Each client has a local dataset, represented by a cylinder. The most influential, anomalous datapoints, marked in red, are filtered using bounds calculated in coordination with the central server. Figure 13a

shows the standard FL, using the filtered dataset: each client runs a local model on their own data, sharing with the server some parameters that are aggregated by the server and sent back to each client. After the last learning round, each client runs a set of epochs locally, using the model parameters obtained after all learning rounds. In this step, all of their data is used, including the datapoints that were previously removed during filtering. This is the second part of FedFil, shown in Figure 13b.

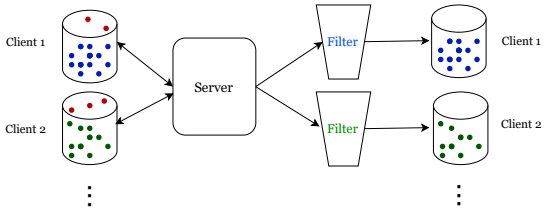
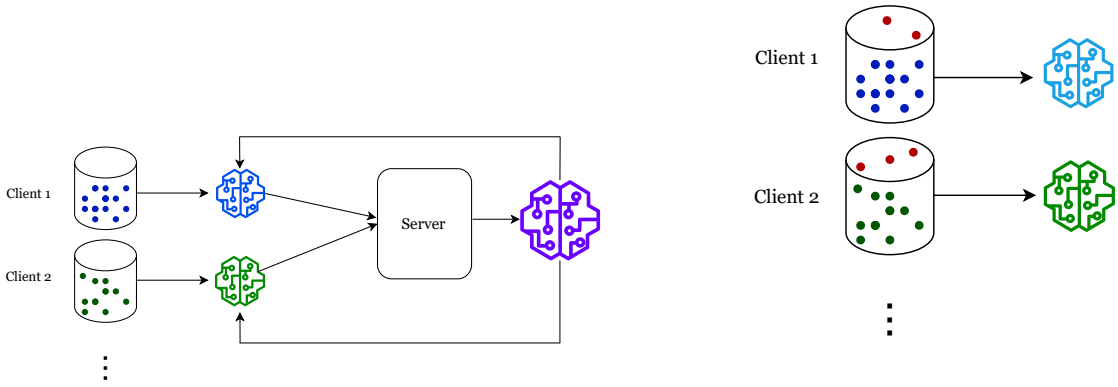


Figure 12: Illustration of Part 1 of FedFil.



(a) Standard FL experiment.

(b) Illustration of Part 2 of FedFil.

Figure 13: Part 2 of FedFil illustrated, with connections to standard FL.

The FedFil scheme provides the foundational framework for addressing the research questions posed in Section 1.2: by analyzing the accuracy of clients under various filtering mechanisms (Part 1), we empirically determine the achievable learning gain and identify the clients who benefit most. The personalized fine-tuning (Part 2) is used to investigate whether local adaptation improves the final accuracy gain.

3.1 Formalization

Let $\mathcal{D} = (x, y)$ denote the global dataset. Suppose there are K clients, denoted as $C_1, \ldots, C_K$. The local dataset of client C_i is denoted as $\mathcal{D}^i = (x_i, y_i)$. w^t and w_i^t are used to denote the global model and the local model of participant C_i in round t, respectively. Thus, w^t is the output model of the FL process.

In the standard **FedAvg** algorithm, each client trains locally on its entire dataset $\mathcal{D}^i$ for a fixed number of epochs, and the resulting updates are sent to the server, which aggregates them into a new global model w^{t+1}.

In contrast, **FedFil** modifies the local training phase by introducing a preprocessing filtering step that removes data points considered outliers or anomalous according to a predefined criterion. This reduced dataset $\mathcal{D}^i_{\text{filt}} \subseteq \mathcal{D}^i$ is used during the FL experiment.

FedFil also introduces an additional personalization step which aims to mitigate the negative impact of the data removal on the model performance during FL, by adding a second local training stage using the full local dataset $\mathcal{D}^i$. Algorithm 1 presents a comparison between **FedFil** and **FedAvg** in more detail.

3.2 Proposed intervals

3.2.1 Parameter estimation in a federated setting

In this work, the preprocessing filtering step that removes influential datapoints relies on interval-based methods, which theoretically require the calculation or estimation of global statistics from the full dataset. This is challenging in FL, as each client only has access to its own local data. This constraint requires the use of **estimation procedures** to approximate global statistics given local statistics.

For clarity of evaluation, the experiments use the **true global statistics** rather than estimated ones, avoiding confounding errors from the estimation process. The estimation methodology is still retained to remain faithful to the FL setting and to highlight the theoretical difference between statistics that allow exact distributed computation (e.g., the mean) and those that require approximation (e.g., quantiles).

Algorithm 1 FedFil: Filtered Federated Learning (comparison with FedAvg). Blue marks steps specific to FedFil.

Input: local datasets $\mathcal{D}^i$, number of clients K, number of communication rounds T, number of local epochs E, learning rate η, number of local refinement epochs $\tilde{E}$
Output: final model w^T $\tilde{w}^T$

Pre-FL Filtering (FedFil only):
for $i = 1$ **to** K **in parallel do**
  client C_i applies **FilterClient**$(\mathcal{D}^i)$

Server executes:
for $t = 0$ to $T - 1$ **do**
  Sample a set of clients S_t ▷ In this work, S_t always includes all clients
  $n \leftarrow \sum_{i \in S_t} |\mathcal{D}^i|$ ▷ FedAvg
  $n \leftarrow \sum_{i \in S_t} |\mathcal{D}^i_{\text{filt}}|$ ▷ FedFil
  for $i \in S_t$ **in parallel do**
    send global model w^t to client C_i
    $\Delta w_i^t \leftarrow$ **LocalTraining**(i, w^t)
  $w^{t+1} \leftarrow w^t - \eta \sum_{i \in S_t} \frac{|\mathcal{D}^i|}{n} \Delta w_i^t$ ▷ FedAvg
  $w^{t+1} \leftarrow w^t - \eta \sum_{i \in S_t} \frac{|\mathcal{D}^i_{\text{filt}}|}{n} \Delta w_i^t$ ▷ FedFil
return w^T to all clients

Client executes:
$L(w; \mathbf{b}) = \sum_{(x,y) \in \mathbf{b}} \ell(w; x; y)$

Local personalization on each client (FedFil only):
for $i = 1$ to K **in parallel do**
  $\tilde{w}_i^T \leftarrow$ **LocalPer**$(w^T, \mathcal{D}^i, \tilde{E})$

function LocalTraining(i, w^t)
  $w_i^t \leftarrow w^t$
  for each batch $\mathbf{b} = \{x, y\}$ of $\mathcal{D}^i$ **do** ▷ FedAvg
    $w_i^t \leftarrow w_i^t - \eta \nabla L(w_i^t; \mathbf{b})$
  for each batch $\mathbf{b} = \{x, y\}$ of $\mathcal{D}^i_{\text{filt}}$ **do** ▷ FedFil
    $w_i^t \leftarrow w_i^t - \eta \nabla L(w_i^t; \mathbf{b})$
  $\Delta w_i^t \leftarrow w^t - w_i^t$
  return Δw_i^t to server

function FilterClient$(\mathcal{D}^i)$
  return filtered dataset $\mathcal{D}^i_{\text{filt}}$ ▷ Filtering strategies are explained by Algorithms 2, 3, and 4

function LocalPer$(w^T, \mathcal{D}^i, \tilde{E})$
  $w_i \leftarrow w^T$
  for $k = 1$ to $\tilde{E}$ **do**
    for each batch $\mathbf{b}$ of $\mathcal{D}^i_{\text{filt}}$ **do**
      $w_i \leftarrow w_i - \eta \nabla L(w_i; \mathbf{b})$
  $\tilde{w}_i^T \leftarrow w_i$
  return $\tilde{w}^T$

Also, for simplicity, this work assumes that the central server acting as the aggregator is a trusted party, that there are no attackers and that there are no sizeable communication differences between participants. In many practical scenarios, however, these assumptions may not hold. In such cases, secure aggregation protocols (Segal et al., 2017) or mechanisms relying on carefully calibrated noise (Chen et al., 2022) can be employed to protect client-level information.

Three methods are proposed: **IQR**, **3SR**, and **CS**. Beyond knowing each client's dataset size n_i, the server must request additional local statistics from the clients, depending on the method employed:

- For the IQR method (Section 3.2.2):
  - The first and third global quartiles of `PINCP`, estimated as described in Section 7.5.
- For the 3SR method (Section 3.2.3):
  - The global mean and standard deviation of `PINCP`, computed using results from Sections 7.4.1 and 7.4.2.
- For the CS method (Section 3.2.4):
  - The vector mean of each client, used to compute the mean of all clients except the client under consideration, was computed using results from Section 7.4.3;
  - Global quantile estimates were derived as in Section 7.5.

The **estimation** workflow used for the IQR method and partially for CS proceed as follows:

1. each participant C_i computes the required local statistics;
2. the server aggregates the received local values to **estimate** the corresponding global quantity, from which the interval is obtained;
3. the resulting interval is then returned to all participants for comparison.

For the remaining methods, the statistic can be **computed exactly** in a distributed manner, with the procedural structure being similar:

1. each C_i computes their local statistics;
2. the server aggregates these contributions to obtain the **exact global statistic** and builds the interval;
3. the interval is disseminated back to the participants for comparison.

3.2.2 Interquartile Range (IQR)

The interquartile range (IQR) has been used since at least the nineteenth century to study variability in observed phenomena (Stanton, 2001). It remains one of the most widely known statistical measures for describing data dispersion. Its theoretical properties have been examined in detail, notably by Whaley (2005). The IQR is particularly useful when data distributions are skewed or contain outliers, as it provides a robust measure of spread less sensitive to extreme values.

John Tukey formally introduced the IQR within the framework of exploratory data analysis (Tukey, 1977), proposing it as a simple yet powerful tool for data summarisation. In this seminal work, Tukey defined the concept of *fences* based on the first quartile (Q_1) and the third quartile (Q_3):

- Q_1 (25th percentile): the median of the lower half of the dataset (values below the median).
- Q_3 (75th percentile): the median of the upper half of the dataset (values above the median).

The fences are then defined as:

$$\begin{aligned} \text{Lower fence} &= Q_1 - k\,(Q_3 - Q_1), \\ \text{Upper fence} &= Q_3 + k\,(Q_3 - Q_1), \\ \text{where } IQR &= Q_3 - Q_1 \end{aligned} \tag{3.1}$$

Typically $k = 1.5$ and $k = 3$, corresponding to the *inner* and *outer* fences, respectively. Observations lying beyond the inner fences are classified as *outside*, and those beyond the outer fences as *far out*. The most common bound is the inner fence ($k = 1.5$), which is also adopted in this work. A visual representation of this method is shown in Figure 14.

As in many applications of this measure, this work will refer to this interval simply as the IQR, with Algorithm 2 explaining its implementation in this work.

Recent applications of the IQR in anomaly detection include its use in data mining (Dash et al., 2023) and intrusion detection (Vinutha et al., 2018).

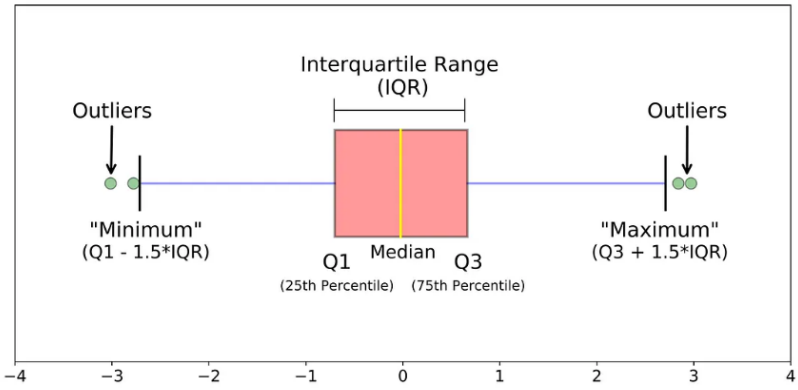


Figure 14: Representation of the Tukey's fences method. Source: Rabie et al. (2024)

Algorithm 2 IQR filtering for dataset $\mathcal{D}^i$

Require: Local datasets $\{\mathcal{D}^1, \ldots, \mathcal{D}^N\}$ and respective sizes $\{n_1, \ldots, n_N\}$, $k = 1.5$

Note: "resp." stands for "respectively", "Eq." stands for "Equation".

Each C_j computes their PINCP first and third quartile, resp. q_1^j and q_3^j

($j = 1, \ldots, N$; C_j denotes participant j, see Section 3.1 for notation.)

Clients send $\mathbf{q_1} = \{q_1^1, \ldots, q_1^N\}$ and $\mathbf{q_3} = \{q_3^1, \ldots, q_3^N\}$ to server

Server estimates the global first and third quartiles, Q_1 and Q_3, using $\mathbf{q_1}$ and $\mathbf{q_3}$

(cf. Section 3.2.1 and Eq. (3.1)*)*

Server calculates and sends **IQR** to C_i

C_i initializes filtered dataset: $\mathcal{D}^i_{\text{filt}} \leftarrow \emptyset$

for each row $d \in \mathcal{D}^i$ **do**

  if $d[\texttt{PINCP}] \in$ **IQR then** ▷ $d[\texttt{PINCP}]$: value of variable PINCP in row d

    Add d to $\mathcal{D}^i_{\text{filt}}$

return $\mathcal{D}^i_{\text{filt}}$

3.2.3 3-Sigma Rule (3SR)

The 3-Sigma rule (3SR) is widely used in conjunction with control charts, which are graphical tools for monitoring variables to support quality management in goods and services. Introduced by Shewhart at Bell Labs in his seminal work on quality control (Shewhart, 1930), control charts remain pivotal in statistical process control today (Colosimo

et al., 2024).

Shewhart control charts are a heuristic method used to distinguish between expected (common cause) variation and variation due to assignable (special) causes. They plot over time the statistics of key variables reflecting process or product quality. If the plotted statistic remains within control limits, the process is considered in control, with observed variation attributed to common causes. Conversely, values outside the control limits indicate a potential change in the process, likely due to an assignable cause requiring attention.

Figure 15 illustrates a typical Shewhart control chart with 3-Sigma control limits, where the limits are conventionally set at three standard deviations from the mean. This choice is grounded in several probabilistic considerations, summarized by Pukelsheim (1994) and Shewhart and Deming (1986):

- Chebyshev's inequality: for any probability distribution, the probability that an outcome deviates more than k standard deviations from the mean is at most $1/k^2$;
- Vysochanskii–Petunin inequality: for any unimodal distribution, the probability of an outcome exceeding k standard deviations from the mean is at most $4/9k^2$;
- Normal distribution: a commonly observed distribution where approximately 99.7% of observations fall within three standard deviations of the mean.

In the present work, the implementation of 3SR is adapted from (Wheeler and Chambers, 1992), with simplifications appropriate to the current setting: only a single group of observations is considered, and given the large sample size, correction factors are not applied (Giles, 2022; Law and Kelton, 1991).

Control limits are then defined as

$$\begin{aligned} \text{LCL} &= \overline{X} - 3S, \\ \text{UCL} &= \overline{X} + 3S, \\ \text{with 3SR} &= [\text{LCL, UCL}] \end{aligned} \tag{3.2}$$

$\overline{X}$ denotes the dataset average and S the dataset standard deviation, both serving as consistent estimators (Cramér, 1946) of the population mean μ and standard deviation σ.

Algorithm 3 presents this work's implementation of 3SR.

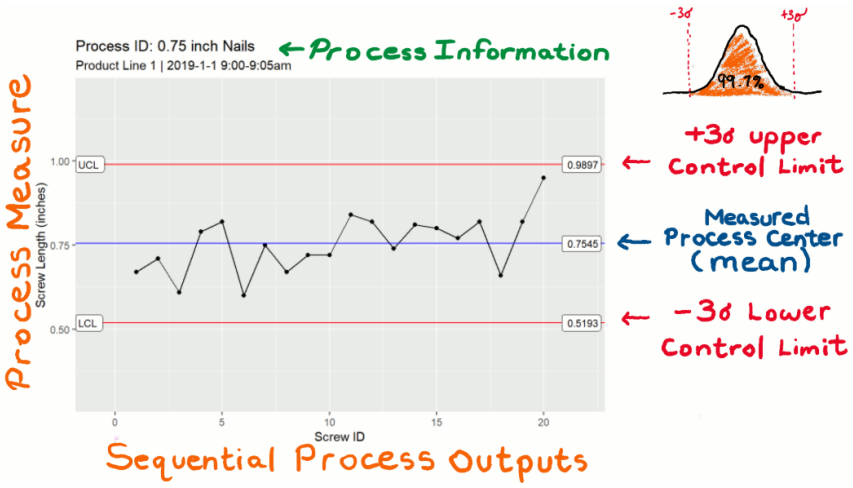


Figure 15: Schematic representation of a Shewhart control chart with 3-Sigma control limits. Source: Grey (2019).

Algorithm 3 3SR filtering for dataset $\mathcal{D}^i$

Require: Local datasets $\{\mathcal{D}^1, \ldots, \mathcal{D}^N\}$ and respective sizes $\{n_1, \ldots, n_N\}$

Output: Filtered local dataset $\mathcal{D}^i_{\text{filt}}$

Note: "resp." stands for "respectively", "Eq." stands for "Equation".

Each C_j calculates their PINCP mean and standard deviation, resp. $\bar{x_j}$ and s_j

($j = 1, \ldots, N$; C_j denotes participant j, see Section 3.1 for notation.)

Clients send $\bar{\mathbf{x}} = \{\bar{x_1}, \ldots, \bar{x_N}\}$ and $\mathbf{s} = \{s_1, \ldots, s_N\}$ to server

Server calculates the global mean $\overline{X}$ and standard deviation S, using $\bar{\mathbf{x}}$ and $\mathbf{s}$

(cf. Section 3.2.1 and Eq. (3.2))

Server sends **3SR** to C_i

C_i initializes filtered dataset: $\mathcal{D}^i_{\text{filt}} \leftarrow \emptyset$

for each row $d \in \mathcal{D}^i$ **do**

  if $d[\texttt{PINCP}] \in$ **3SR then** ▷ $d[\texttt{PINCP}]$: value of variable PINCP in row d

    Add d to $\mathcal{D}^i_{\text{filt}}$

return $\mathcal{D}^i_{\text{filt}}$

Beyond quality control, the 3-Sigma rule has also been applied to anomaly detection in survey engineering (Lehmann, 2013).

3.2.4 Cosine Similarity (CS)

The cosine similarity between two vectors **A** and **B** in $\mathbb{R}^n$ is defined as the cosine of the angle θ between them:

$$\text{cosine\_similarity}(\mathbf{A}, \mathbf{B}) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \, \|\mathbf{B}\|} \tag{3.3}$$

where

$$\mathbf{A} \cdot \mathbf{B} = \sum_{i=1}^{n} A_i B_i, \quad \|\mathbf{A}\| = \sqrt{\sum_{i=1}^{n} A_i^2}, \quad \|\mathbf{B}\| = \sqrt{\sum_{i=1}^{n} B_i^2},$$

The cosine similarity ranges from -1 to 1, where values close to:

- 1 indicates vectors pointing in the same direction **(similar vectors)**;
- 0 indicates orthogonality **(no association)**;
- -1 indicates vectors pointing in opposite directions **(dissimilar vectors)**.

Figure 16 provides a visual representation of these cases.

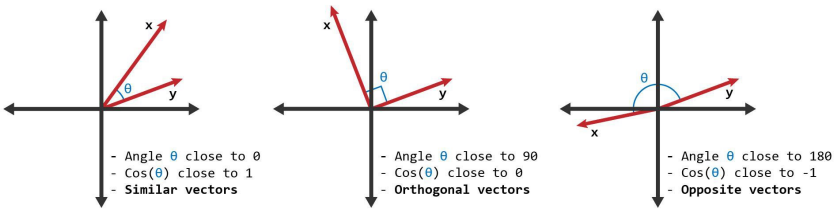


Figure 16: Cosine similarity illustrated through three scenarios in $\mathbb{R}^2$. Source: LearnDataSci (nd).

In this work, the idea is to use cosine similarity to help detect discrepant, anomalous data points that certain clients have in comparison with the rest of the clients. Anomaly detection applications using cosine similarity include defects in aircraft assembly (Shete et al., 2024) and in-vehicle networks (Kwak et al., 2021). Cosine similarity takes in account the multidimensionality nature of the dataset, unlike the methods in Sections 3.2.2 and 3.2.3, which can be applied solely on a single variable.

To achieve this, the cosine similarity of each data point of client i, C_i, will be compared to the mean of the set of all clients except C_i. This differs from the methods mentioned before, which included data from all participants to design the intervals.

Given that much of the dataset is composed of categorical features (Table 2), it was necessary to transform these features in order to calculate cosine similarity. The transformations are listed in 4. The whole procedure used is summarized in Algorithm 4.

Algorithm 4 CS filtering for dataset $\mathcal{D}^i$

Require: Local datasets $\{\mathcal{D}^1, \ldots, \mathcal{D}^N\}$ and respective sizes $\{n_1, \ldots, n_N\}$

Output: Filtered local dataset $\mathcal{D}^i_{\text{filt}}$

Each participant j creates a *preprocessed copy* $\widetilde{\mathcal{D}}^j$ of $\mathcal{D}^j$ ▷ see Section 4

Each client computes their local mean vector u^j

Server receives local mean vectors u^j

for $k = 1, \ldots, N$ **do**

Server computes global mean without u^k: u^k_{others}

Server sends u^k_{others} to C_k ▷ C_k denotes client k; see Section 3.1 for notation.

C_k initializes vector $s^k \in \mathbb{R}^{n_k}$

for each preprocessed row $\tilde{d} \in \widetilde{\mathcal{D}}^k$ **do**

Compute cosine similarity: $s^k(\tilde{d}) \leftarrow \text{cosine\_similarity}(\tilde{d}, u^k_{\text{others}})$

Clients send $\mathbf{s} = \{s^1, \ldots, s^N\}$ to the server ▷ See Figure 17 for a plot of **s**

Server estimates the global quantile τ, using **s**, sending it to C_i

C_i initializes filtered dataset: $\mathcal{D}^i_{\text{filt}} \leftarrow \emptyset$

for each row $d \in \mathcal{D}^i$ **do**

if $s^i(d) \geq \tau$ **then** ▷ Optionally, $s^i(d) \in [\tau, 1]$

Add d to $\mathcal{D}^i_{\text{filt}}$

return $\mathcal{D}^i_{\text{filt}}$

Figure 17 shows the cumulative distribution of all pairs of cosine similarity computed between each participant and an aggregated mean of the other participants. From there, since most cosine similarity pairs are close to 1, it follows that the data points are highly similar to the rest of the dataset. However, this hides considerable variability seen in each client.

The chosen thresholds correspond to the global 1% and 2.5% percentiles of the cosine similarity distribution, derived from the application of IQR (Section 5.2) and 3SR (Section 5.3) to FedFil, where IQR higher removal rates

compared with those of 3SR (Section 7.3) proved too stringent.

Figures 18a and 18b show the distribution of cosine similarity results below the global 1% and 2.5% percentile thresholds, respectively. Although both figures illustrate variability within clients, the figure for the 2.5% threshold shows medians (colored in green) ranging from below 0.2 to above 0.8. When the same percentiles are applied locally (Figure 19), even greater variation is observed.

Figure 19a is particularly illustrative: many clients have maximum cosine similarity values around 0.2, while another large set of participants have values exceeding 0.8, highlighting the variation across clients.

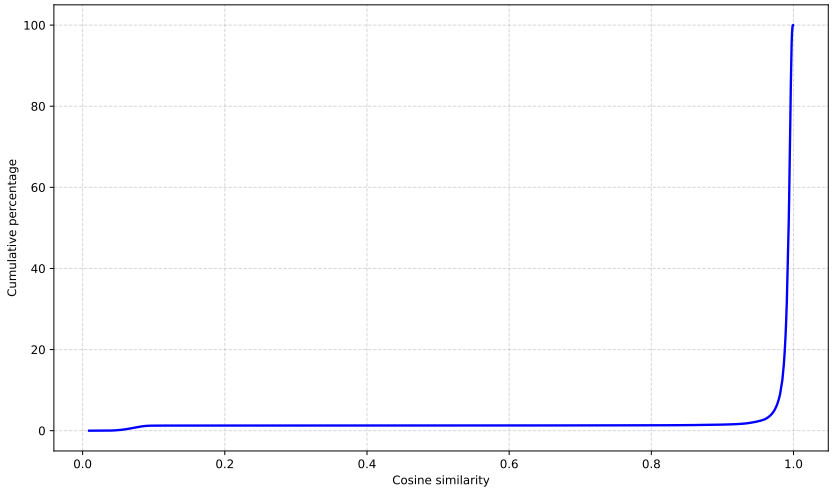
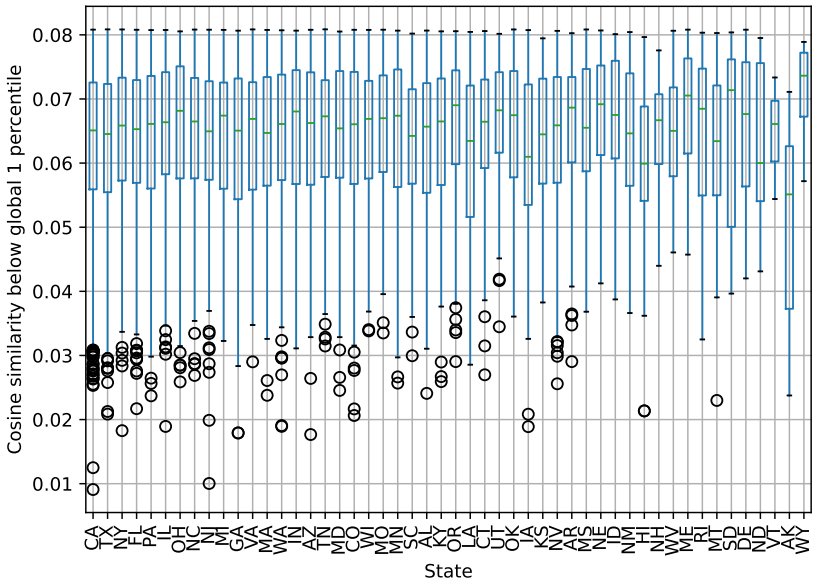
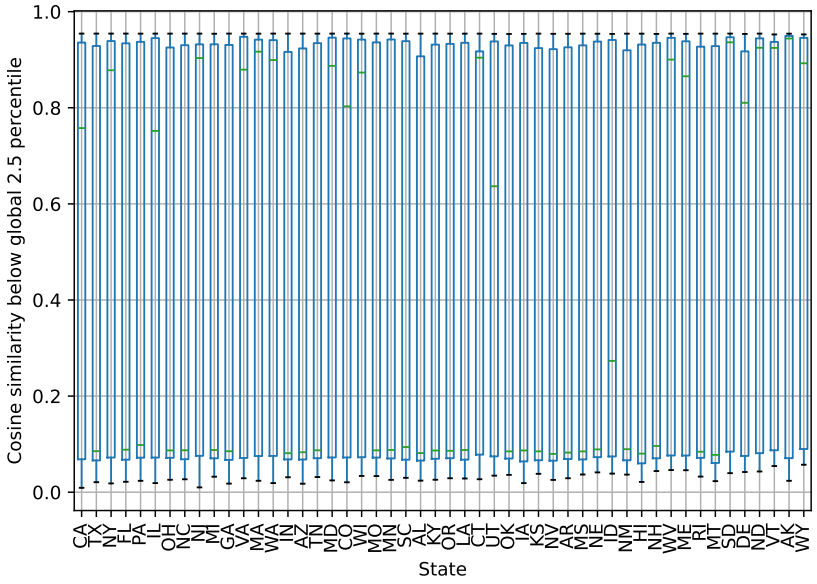


Figure 17: CD of all cosine similarity values computed between each participant's local data points and the corresponding global mean vector of other participants.

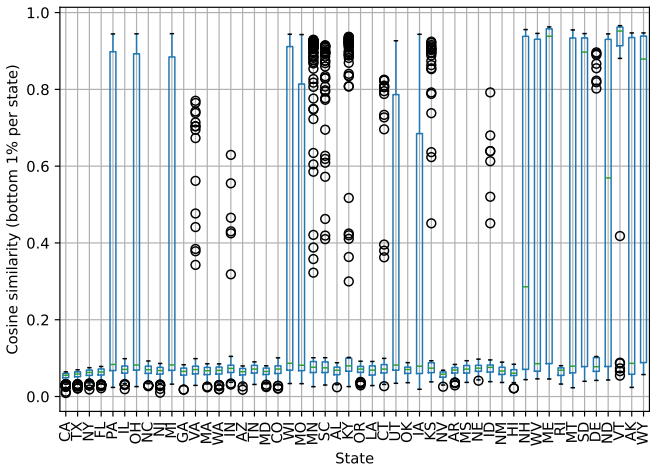


(a) Client distribution of cosine similarity results below global 1% percentile.

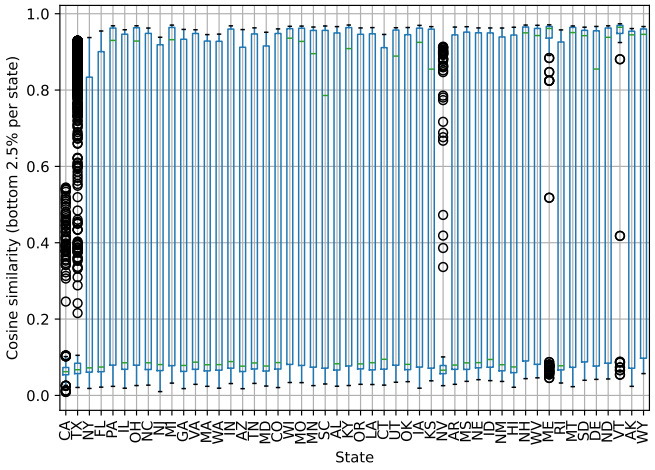


(b) Client distribution of cosine similarity results below global 2.5% percentile.

Figure 18: Client distribution of cosine similarity results below proposed global thresholds.



(a) Client distribution of cosine similarity results below local 1% percentile.



(b) Client distribution of cosine similarity results below local 2.5% percentile.

Figure 19: Client distribution of cosine similarity results below local 1% and 2.5% percentiles.

Chapter 4

Experimental setup

4.1 Dataset description

For the development of the experimental work presented in this dissertation, two datasets with distinct characteristics were considered: the Adult dataset, constructed from the American Community Survey (ACS) Public Use Microdata Sample (PUMS) (U.S. Census Bureau, 2024), and the Sentiment140 dataset, from now on shortened to Sent140. Both were initially assessed as potential candidates for implementing FL schemes focused on evaluating performance under non-IID conditions and exploring data selection strategies. After preliminary analysis, only the first was retained for the main experiments, while the second was discarded due to unsatisfactory performance resulting from artificial construction (Section 4.1.2).

4.1.1 Adult

The Adult dataset is based on the ACS PUMS, which contains anonymized microdata about individual people or housing units representative of the US population. It was filtered for the year 2021 and for the one-year horizon. It offers a modern, large-scale alternative to the UCI Adult dataset (Ding et al., 2022), with richer demographic and socioeconomic coverage. Data extraction and preprocessing were carried out using the `Folktables` Python package (Ding et al., 2021), which facilitates the creation of predictive tasks from ACS microdata. This tabular dataset includes approximately **14.34 million** observations, divided between 1.303 million rows and 11 columns,

filtered according to the following criteria:

- Individuals aged over 16;
- Personal Income (PINCP) greater than $100;
- At least one hour of work per week in the past year (WKHP > 0).

The target variable, **PINCP_bin**, for this classification task is derived from the continuous variable **PINCP**, which reports an individual's total personal income. **PINCP_bin** was created using the `ACSIncome` module from the `Folktables` Python package, income values from **PINCP** were thresholded at $50,000 to define a binary label: individuals earning more than $50000 were assigned label 1, and the remaining individuals were assigned label 0. This preprocessing step was applied consistently across all selected U.S. states. The predictive features consist of a mix of demographic, educational, occupational, and relational variables, including:

- **Demographic attributes**: age (AGEP), sex (SEX), race (RAC1P), place of birth (POBP);
- **Educational and employment attributes**: educational attainment (SCHL), class of worker (COW), occupation (OCCP), usual hours worked per week (WKHP);
- **Household attributes**: marital status (MAR), relationship to reference person (RELP).

More details about each feature, including type and value range, can be found in Table 2.

In this work, the dataset was partitioned by US state, with each state treated as an independent client in the FL framework.

Figure 20 depicts the distribution of dataset size across clients. The imbalance is stark: while states like California (CA) and Texas (TX) have over 100,000 data points, others such as Delaware (DE), North Dakota (ND), and Vermont (VT) have fewer than 20,000. This reinforces the notion of unbalanced participation in the federated setting.

Another way to illustrate this discrepancy is through a cumulative distribution (CD), as shown in Figure 21. More than 40% of the clients, over 20 states, have a dataset size below 20,000. Approximately 80% have fewer than 40,000 data points, whereas the remaining 20% concentrate disproportionately large datasets, with sizes ranging from 40,000 to nearly 160,000.

Table 2: Summary of ACS features used in this work.

Variable	Type	Value Range	Description
AGEP	Numerical discrete	16–99	Age in years;
SEX	Categorical nominal	1: Male, 2: Female	Biological sex.
RAC1P	Categorical nominal	1–9	Detailed race code; includes multiracial categories such as "Two or More Races" encoded by 9.
POBP	Categorical nominal	0–554	Place of birth; includes U.S. states (e.g., Alabama = 1) and foreign countries.
SCHL	Categorical ordinal	1–24	Educational attainment; from no schooling (1) to doctorate degree (24).
COW	Categorical nominal	1–9	Class of worker; includes private wage/salary (1), federal government (5), and unemployed or not in labor force (9).
OCCP	Categorical nominal	ACS codes	Occupation; includes professions such as Chief Executives and Legislators (code 10); full list in U.S. Census Bureau (2025).
WKHP	Numerical discrete	1–99	Usual hours worked per week; 99 indicates 99 or more hours.
MAR	Categorical nominal	1–5	Marital status; includes married (1), widowed (2), divorced (3), separated (4), and never married (5).
RELP	Categorical nominal	20–38	Relationship to reference person in household; includes Father or mother (29), Grandchild (30), and other household relationships (codes 20–38).
PINCP	Numerical continuous	$(100, +\infty)$	Total personal income in dollars.

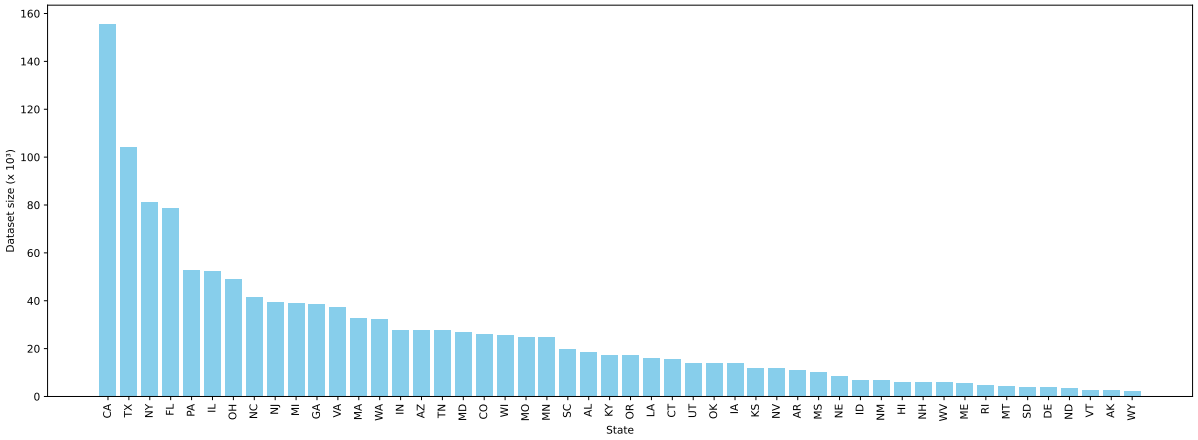


Figure 20: Dataset size per client in the Adult dataset.

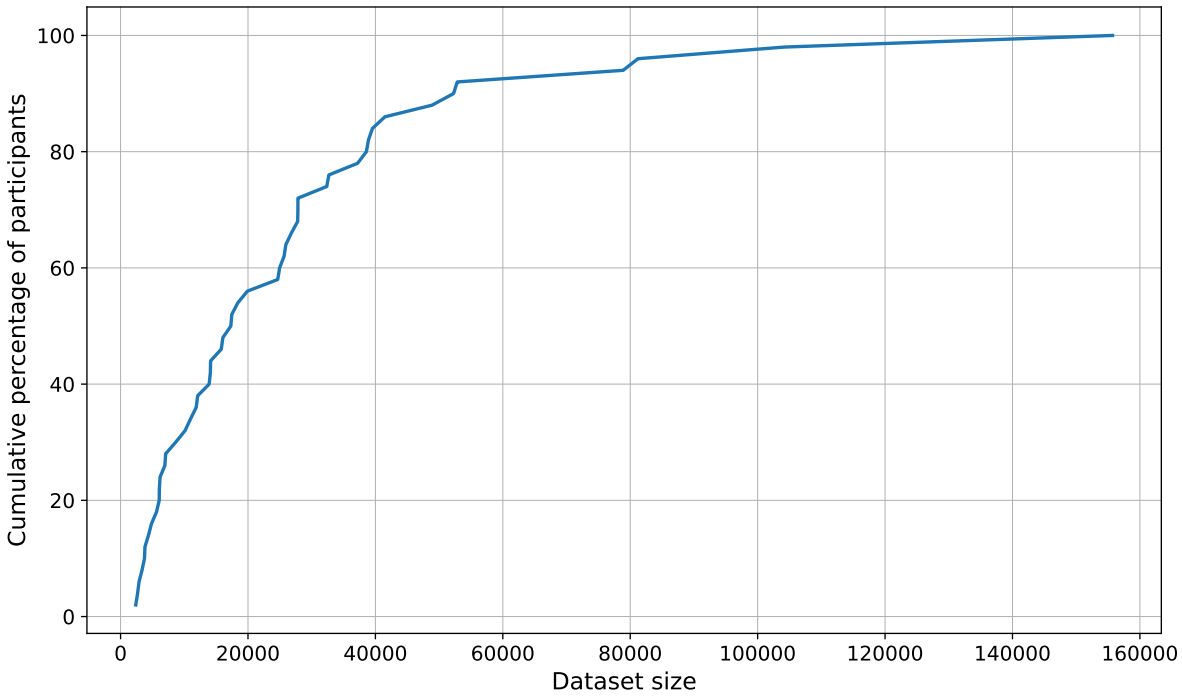


Figure 21: CD of dataset sizes across clients in the Adult dataset.

4.1.2 Sent140

The Sent140 dataset is a large-scale corpus of tweets labeled for sentiment classification, constructed using distant supervision techniques. Originally introduced by Go et al. (2009), it consists of **1.6 million tweets** collected between 2008 and 2009 from the Twitter platform. The dataset was created by leveraging emoticons as weak supervision signals to assign sentiment labels automatically, without the need for manual annotation.

Each tweet in the dataset is originally labeled with one of three sentiment classes: 0 (negative), 2 (neutral), and 4 (positive). Following the setup proposed by Cho et al. (2024), the neutral class is discarded, and the negative and positive labels are remapped to 0 and 1, resulting in a binary classification task. The input features consist of the raw tweet text, which is preprocessed after label assignment by removing usernames, URLs, and emoticons.

The dataset includes the following fields:

- **Tweet ID** – a unique identifier;
- **Date and time** – timestamp of publication;
- **Query** – the keyword used to retrieve the tweet (if any);
- **User** – the username of the Twitter account;
- **Text** – the tweet content, stripped of emoticons used for labeling;
- **Label** – sentiment: 0 for negative and 4 for positive (binary classification).

In the FL setting used in this study, the dataset is partitioned into 31 clients, each corresponding to a distinct Twitter user with sufficient message volume.

Despite its scale, its construction is artificial and greatly impacts the performance of the model when used outside the context introduced in Cho et al. (2024), it was ultimately excluded from the main experiments.

4.2 Descriptive Statistics and Feature Distributions

4.2.1 Adult

Before implementing and evaluating FL methods, it is essential to gain a comprehensive understanding of the underlying data structure. This section presents descriptive statistics and feature distributions for the datasets under consideration, with a particular focus on the ACS-based Adult dataset, which was selected for the main experiments.

The ACS dataset contains a rich set of demographic and occupational features. The descriptive statistics and visualizations included here focus both on the distribution of the target variable **PINCP_bin** across clients, its association with other features, and the distributions of the other features. Note that both PINCP and PINCP_bin are examined in this chapter: the former serves as the feature from which the target variable is derived, while the latter corresponds to the target variable itself. However, neither is **directly** incorporated into the modelling stage. In particular, **PINCP is not used as a predictive feature**, and therefore is not part of the model architecture described in Section 4.3.2.

Although the Sent140 dataset was ultimately left out of the main experimental phase, a short description of its user-level distribution is provided here for completeness.

4.2.2 Numerical features

PINCP

We begin with a focused analysis of PINCP, personal income. Figure 22 shows the distribution of the target variable (PINCP) across clients. The plot reveals a high number of outliers and substantial dispersion in income levels across states. For instance, Alaska (AK) has a maximum personal income close to $600,000, whereas states like California (CA) and New Jersey (NJ) exhibit maximum values exceeding $1,400,000, more than 2.5 times higher. This disparity is expected given the socioeconomic heterogeneity captured by the dataset.

Figure 23 presents the distribution of **PINCP_bin**, obtained by thresholding PINCP at $50,000. As shown in Figure 23, this binarization reduces some of the variability, but the balance between the two income classes still varies substantially among clients.

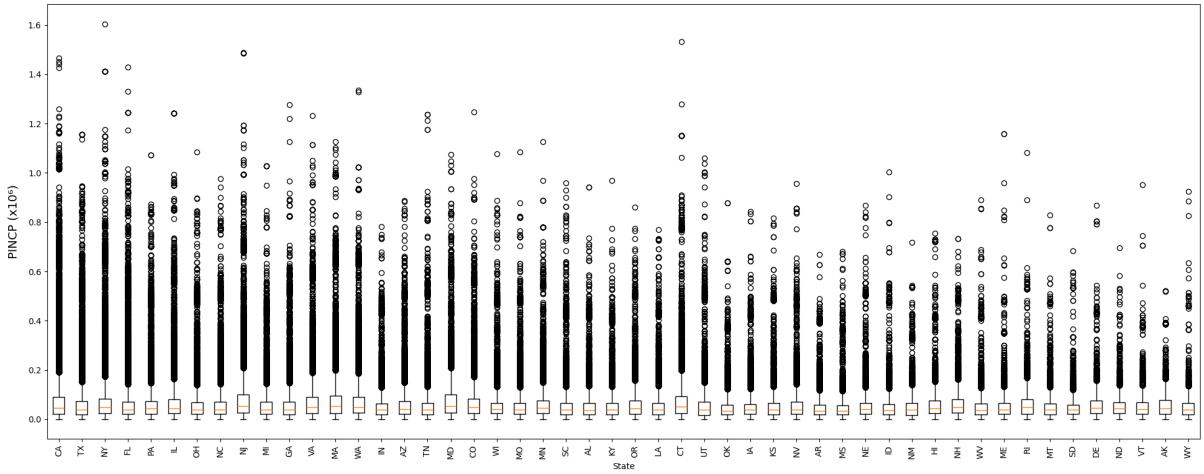


Figure 22: Distribution of PINCP across clients in the Adult dataset.

Most clients exhibit a higher proportion of individuals in the lower income class (≤ $50,000), but the proportion of high-income individuals (> $50,000) ranges from approximately 20% to nearly 50%. In both Figures 22 and 23, the states are ordered by dataset size. A visual inspection suggests no apparent relationship between the number of data points per client and the dispersion of income values, nor between dataset size and the proportion of individuals above or below the $50,000 threshold.

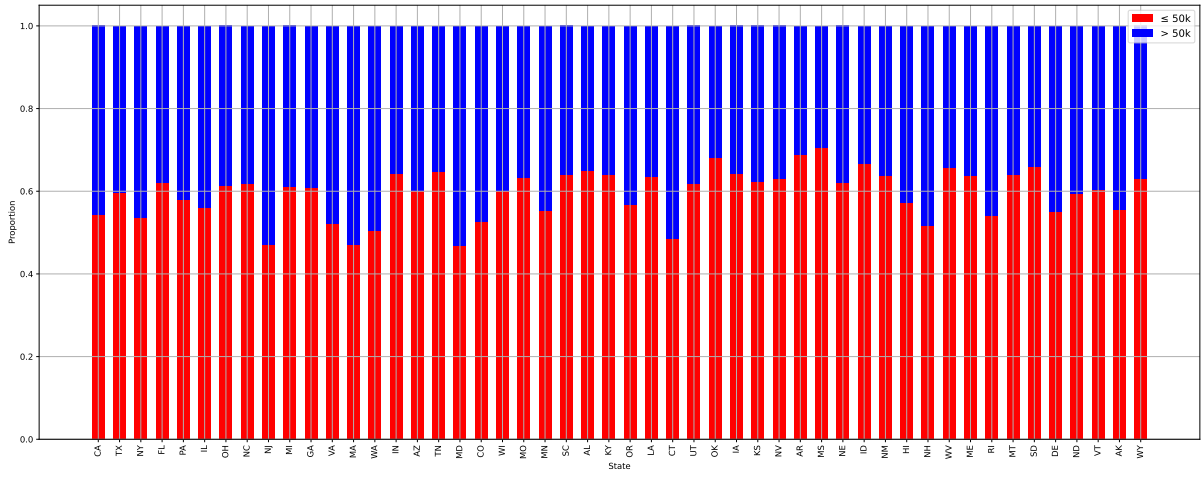


Figure 23: Proportion of individuals above and below the $50000 income threshold across clients.

Taken together, Figures 23, 22, 20 and 21 highlight the non-IID nature of the dataset, reflected in the diversity of value distributions and the heterogeneity in dataset size between clients.

AGEP

Figure 24 presents the relative frequency distribution of the variable AGEP, which corresponds to the age of individuals in the dataset. The histogram reveals a right-skewness distribution, with the highest concentration of individuals situated between the ages of 18 and 60, indicating a demographic predominance of working-aged adults. Frequencies form a long tail beyond 60, approaching negligible levels past 80.

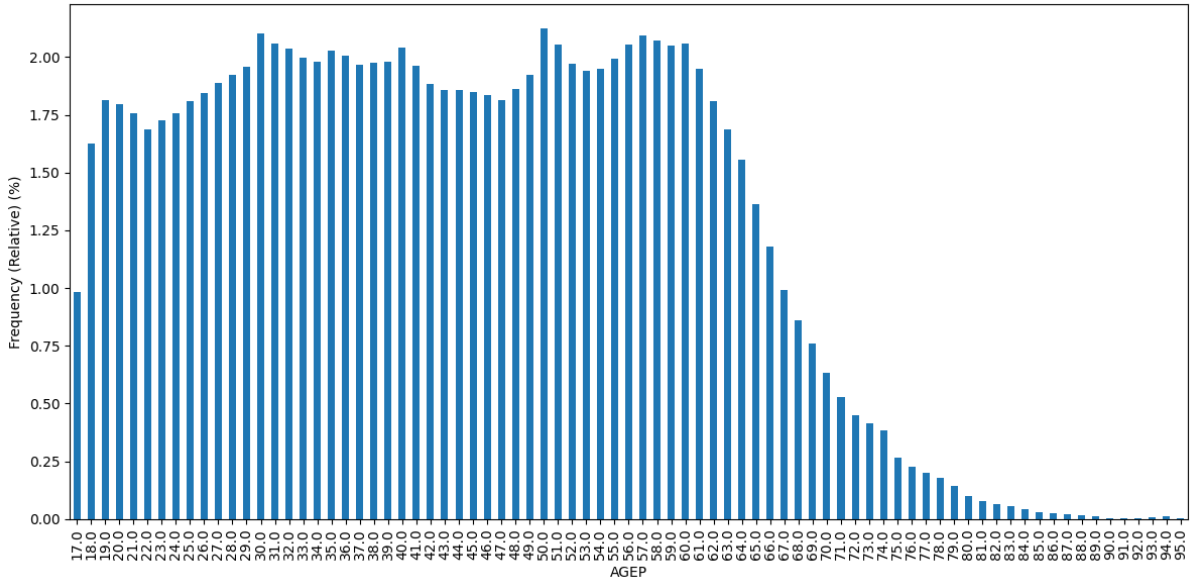


Figure 24: Distribution of the age of individuals, in the dataset.

The histogram also suggests a relatively smooth distribution without abrupt discontinuities, implying that the dataset captures age data with sufficient granularity. The tailing off of frequencies in older age brackets may reflect both natural demographic attrition and potential underrepresentation in data collection processes, particularly among the elderly.

Figure 25 complements this analysis by presenting a boxplot of AGEP across U.S. states, enabling a comparative assessment of age distributions at the state level. Each boxplot encapsulates the interquartile range (IQR), a statistical tool used to detect outliers, median age (indicated by the green line), and outliers (represented by black circles). The visualization shows some heterogeneity: states such as Alaska (AK) and Utah (UT) display lower medians and narrower IQRs.

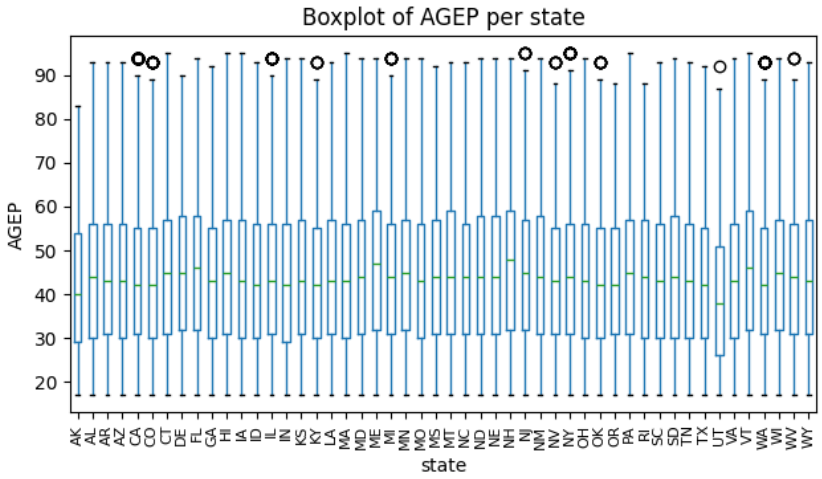


Figure 25: Distribution of the age of individuals, per state.

WKHP

Figure 26 presents the relative frequency distribution of usual hours worked per week (WKHP). To enhance interpretability for this visualization, extreme values were grouped using the IQR of WKHP: values ≤27 (lower bound) and ≥47 (upper bound) were aggregated. This preserves the central structure while mitigating visual clutter from low-frequency extremes. Observations are highly concentrated at WKHP = 40, which accounts for more than 40% of the dataset. The aggregated classes of 27 hours or less and 47 hours or more each represent over 10% of the observations, indicating some dispersion but maintaining a strong central tendency around the standard full-time work week.

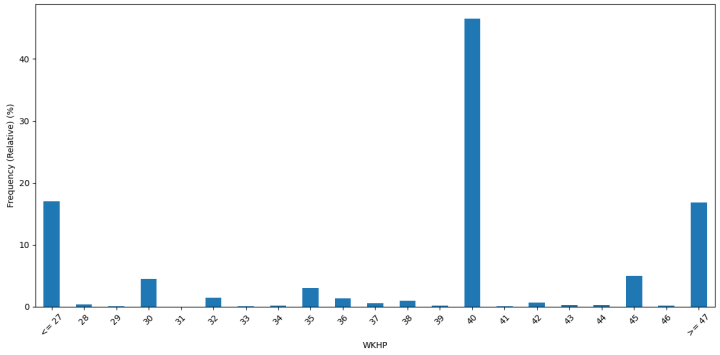


Figure 26: Distribution of usual hours worked across individuals in the dataset.

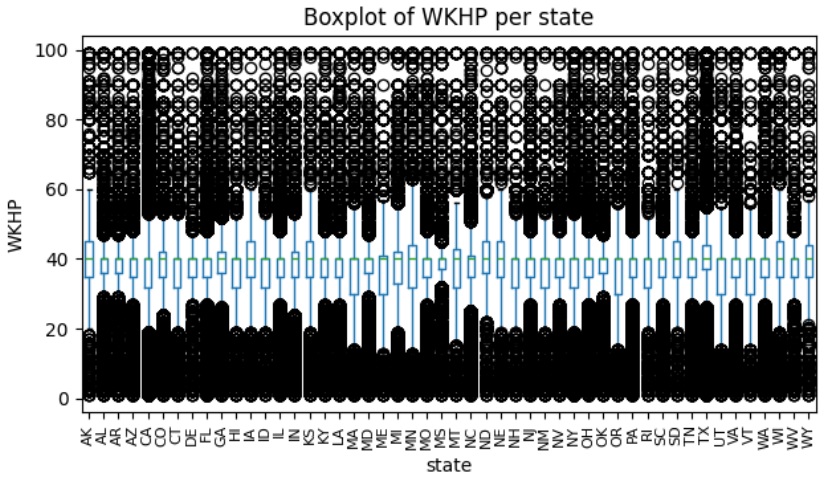


Figure 27: Distribution of usual hours worked across individuals, per state.

Figure 27 complements this analysis with a boxplot for each state. Medians cluster around 40 hours, consistent with the dominant peak observed in Figure 26, but variations across boxplots indicate differences in work-hour distribution between states.

4.2.3 Categorical features

Having concluded the analysis of continuous variables, we now turn to the categorical features. For improved readability and interpretability, all visualizations in this section follow a consistent aggregation strategy:

- Only the 20 most populous states are displayed, ordered by dataset size, with the remaining ones grouped under *Other states*;
- For variables with high cardinality, only the 20 most frequent categories are retained, with the remaining aggregated into a single *rest* category.

These choices preserve representativeness: together, the 20 most populous states account for roughly 76% of the dataset, and enhance readability by preventing the plots from becoming visually cluttered, especially for categorical variables with high cardinality.

SEX

We start with SEX, denoting the biological sex, where the value 1 maps to male and the value 2 to female. Figure 28 presents the relative frequency distribution of SEX across the entire dataset. The barplot reveals a near-balanced distribution between the two categories, with a slight predominance of men, which exceeds 50% of the dataset.

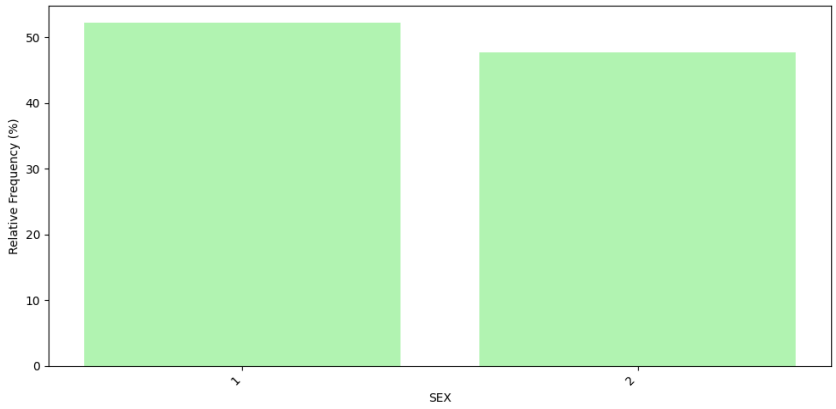


Figure 28: Distribution of biological sex across individuals, per state.

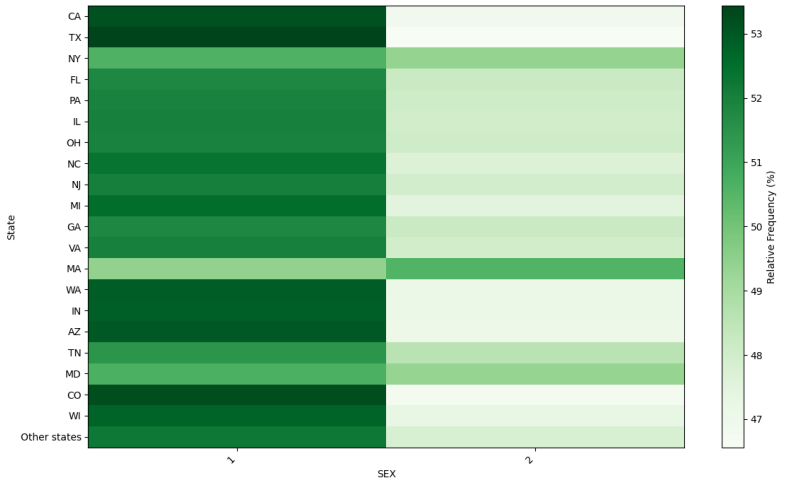


Figure 29: Distribution of biological sex across individuals in the dataset.

Regional variation is shown in Figure 29, a heatmap of SEX frequencies across states. With values ranging from approximately 46.5% to 53.5%, the distribution remains close to uniform across states, with only minor deviations. Minor deviations occur, e.g., California (CA) shows slightly more females, whereas Texas (TX) and Florida (FL) are more balanced.

RAC1P

Figure 30 and 31 present the distribution of the variable RAC1P, which encodes race categories. The barplot reveals a strong predominance of category 1 ("White Alone"), with relative frequencies approaching 70%, while the remaining categories have substantially lower representation. Categories such as 2 ("Black or African American alone"), 6 ("Asian alone"), 9 ("Two or More Races") and 8 ("Some Other Race alone") show moderate frequencies, whereas others fall below 5%, indicating a highly skewed distribution.

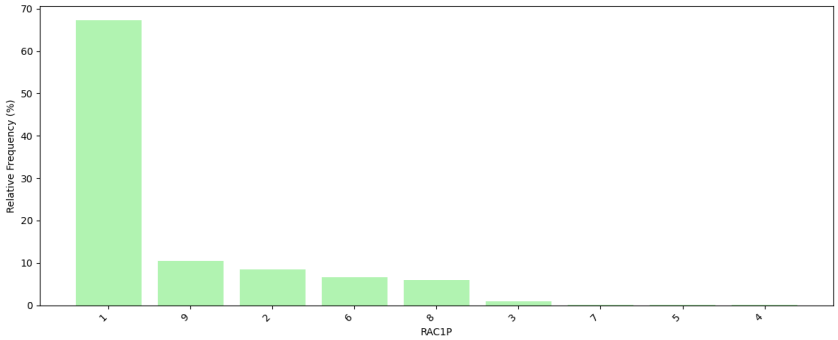


Figure 30: Race distribution across individuals, per state.

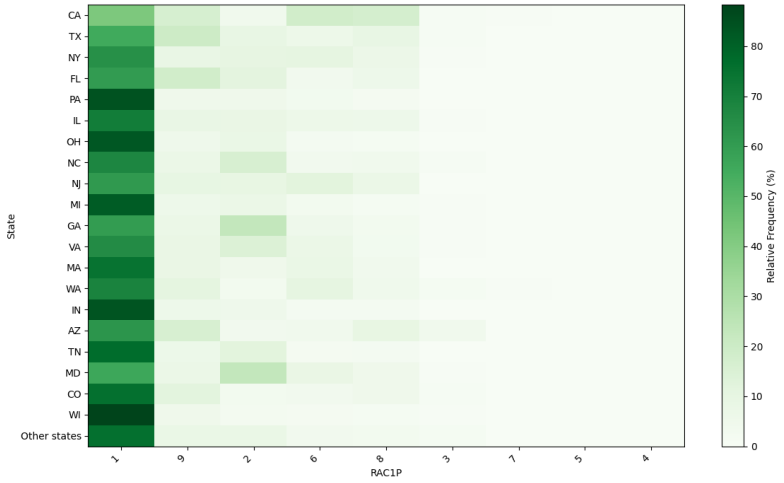


Figure 31: Race distribution across individuals in the dataset.

The heatmap confirms the national trend: most states exhibit a high concentration of category 1, with some regional variation in the representation of other race groups. States like CA and TX show relatively higher frequencies for categories beyond 1, reflecting more diverse racial compositions.

POBP

Figure 32 and 33 illustrate the distribution of the variable POBP, which encodes place of birth categories. The barplot (Figure 32) shows a highly skewed distribution: the aggregated *rest* category, grouping all less frequent values, accounts for more than 35% of the observations. All other categories occur at much lower frequencies, with the next most common values falling below 10%.

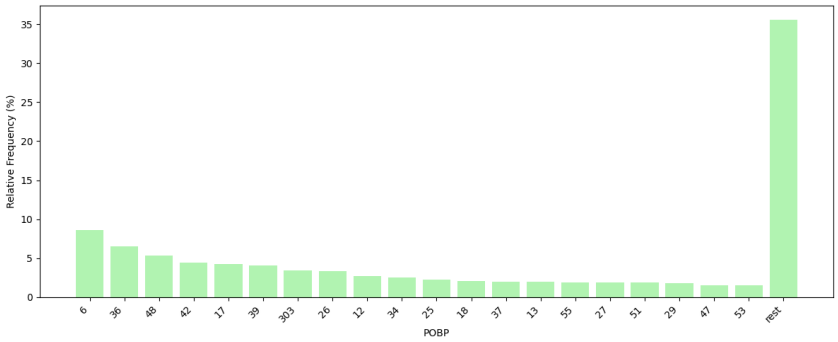


Figure 32: Distribution of place of birth across individuals, per state.

In the heatmap (Figure 33), the most frequent values at the state level generally correspond to the state itself. For example, Ohio (OH), encoded by the number 39 in POBP, shows its highest relative frequency at that code. This trend is consistently observed across most states, with the exception of Florida (FL, code 12), where the state code still exhibits the highest relative frequency, although less pronounced.

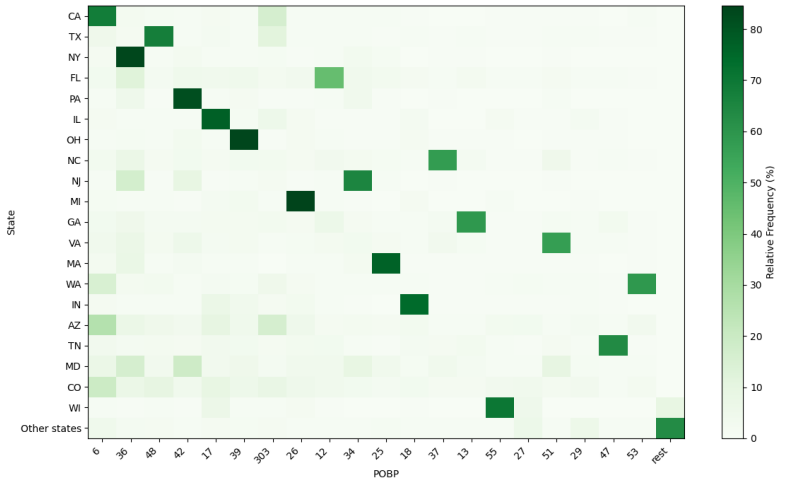


Figure 33: Distribution of place of birth across individuals in the dataset.

SCHL

Figure 34 and 35 present the distribution of SCHL, which encodes educational attainment as an ordinal scale ranging from 1 (no schooling) to 24 (doctoral degree). This ordered coding reflects progressively higher levels of formal education.

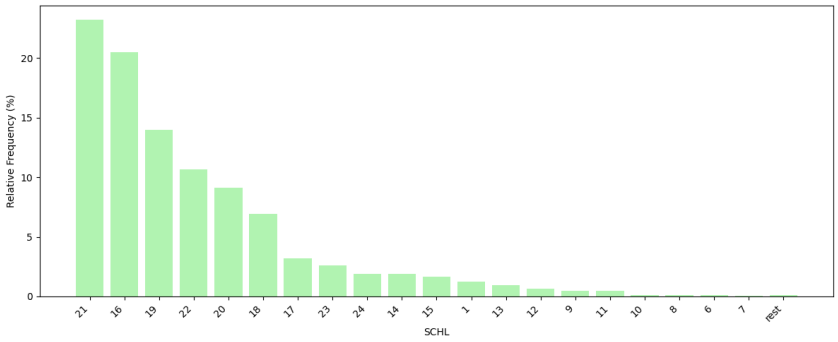


Figure 34: Distribution of educational attainment across individuals, per state.

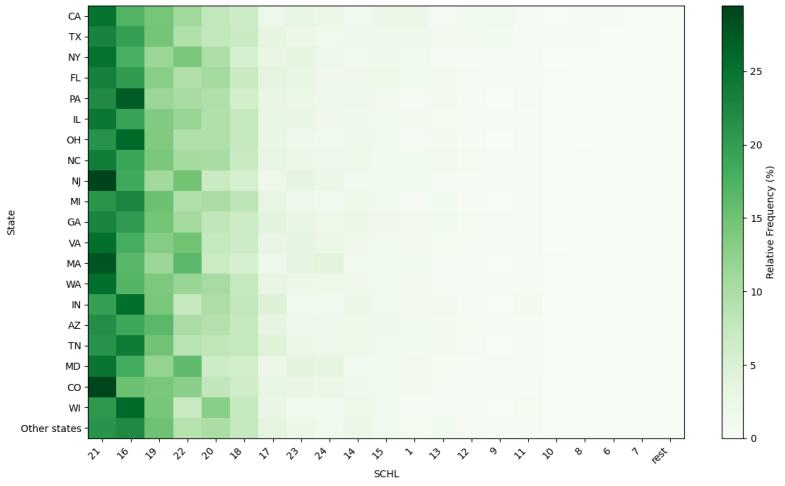


Figure 35: Distribution of educational attainment across individuals in the dataset.

The barplot (Figure 34) shows a marked concentration in higher education categories. The most frequent value is SCHL = 21 (bachelor's degree), followed by SCHL = 16 (high school graduate), SCHL = 19 (some college, no degree), and SCHL = 22 (master's degree). Together, these four categories account for the majority of observations, indicating that the dataset is skewed toward mid-to-high levels of education. By contrast, the lower educational categories (SCHL ≤ 10) collectively represent under 5% of the data. The heatmap (Figure 35) provides additional evidence that this pattern is consistent across states. Higher values are concentrated around SCHL = 16, 19, and

21. States such as Massachusetts (MA), New York (NY), and California (CA) show relatively higher proportions with SCHL ≥ 21, whereas Texas (TX) and Florida (FL) display more balanced distributions across mid-level attainment.

COW

Figure 36 and 37 illustrate the distribution of the variable COW, which encodes class of worker categories. The barplot (Figure 36) shows a highly skewed distribution, with category 1 accounting for over 60% of the sample. Categories 2, 3, and 6 exhibit moderate frequencies around 10%, while the remaining categories fall below 7%, with category 8 being particularly rare.

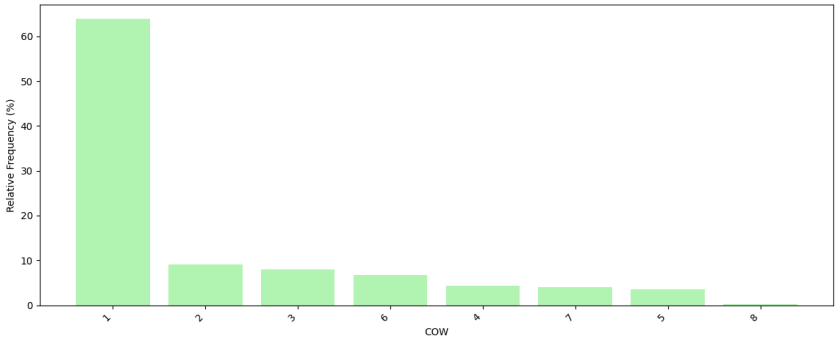


Figure 36: Distribution of class of worker across individuals, per state.

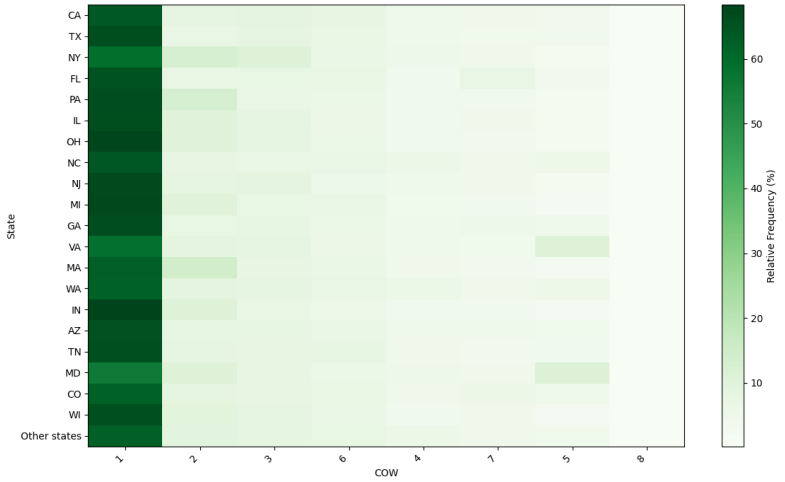


Figure 37: Distribution of class of worker across individuals in the dataset.

The heatmap (Figure 37) reinforces this national trend: category 1 dominates across nearly all states, with some regional variation in the representation of other classes. Nonetheless, the overall distribution remains heavily

concentrated in a single category and is homogeneous across states.

OCCP

Figure 38 and 39 present the distribution of OCCP, which encodes occupational categories using standardized numerical codes defined by the U.S. Census. These span management, technical, administrative, service, and manual labor roles.

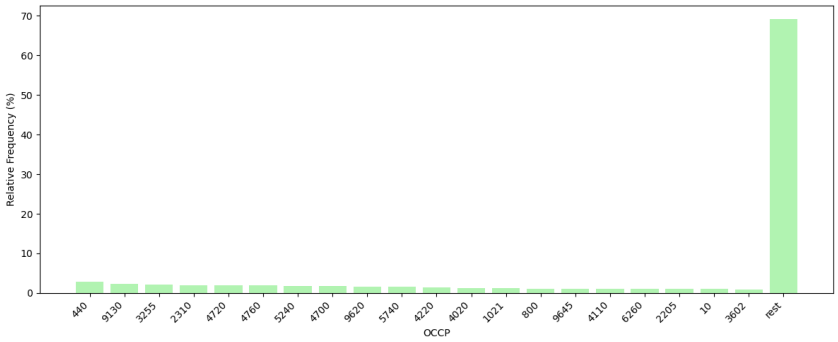


Figure 38: Distribution of occupation across individuals, per state.

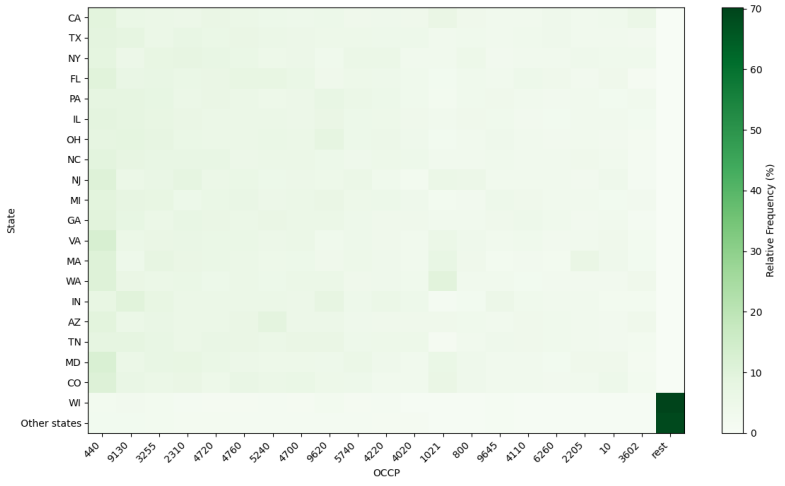


Figure 39: Distribution of occupation across individuals in the dataset.

The barplot (Figure 38) reveals a highly dispersed distribution, with no single occupational category exhibiting clear dominance. The most frequent categories, such as 440, 9130, and 3255, each represent only a small fraction of the dataset. The *rest* category, which aggregates low-frequency occupations, alone accounts for around 70%

of observations. This long-tailed behavior indicates that the dataset contains a broad and heterogeneous set of occupations. OCCP is thus the most dispersed categorical variable analyzed so far.

The heatmap (Figure 39) provides further evidence of this diversity: occupational frequencies are evenly distributed across most regions, with no strong spatial concentration. Wisconsin (WI) and the group *Other states* are notable exceptions, showing big relative frequencies in the *rest* category.

This high occupational heterogeneity contrasts sharply with the next variable, MAR, where the majority of observations are concentrated in two categories.

MAR

Figure 40 and 41 present the distribution of MAR, which encodes marital status using five categories: 1 (Married), 2 (Widowed), 3 (Divorced), 4 (Separated), and 5 (Never married).

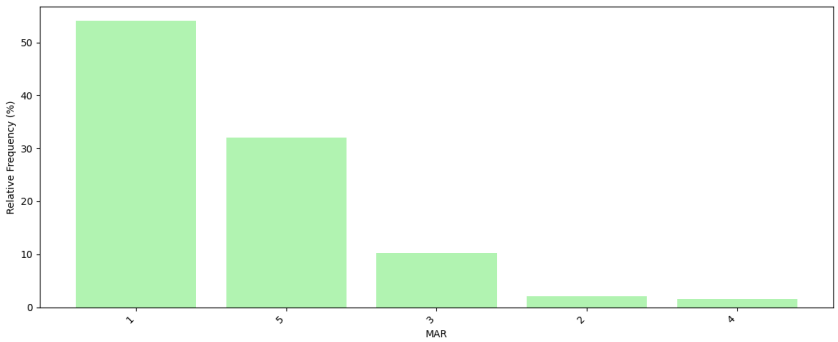


Figure 40: Distribution of marital status across individuals, per state.

The barplot (Figure 40) shows that MAR = 1 (Married) accounts for around 55% of the dataset, and MAR = 5 (Never married) for around 30%. The remaining levels, MAR = 3, 2, and 4, collectively represent < 15% of observations, indicating that alternative marital statuses are comparatively rare.

The heatmap in Figure 41 further supports the global trend: most states exhibit darker intensities for MAR = 1 and MAR = 5, indicating higher concentrations of married and never-married individuals. The heatmap also highlights subtle regional differences. For example, states such as Texas (TX) and Wisconsin (WI) show a stronger representation on the Married category, while states like California (CA), New York (NY) and Massachussets (MA) exhibit higher proportions of MAR = 5 (Never married), possibly reflecting demographic and lifestyle factors. Overall, frequencies for the remaining levels are consistently low across all regions.

While MAR highlights the dual structure of marital status, the next variable, RELP, provides structural insight into household composition.

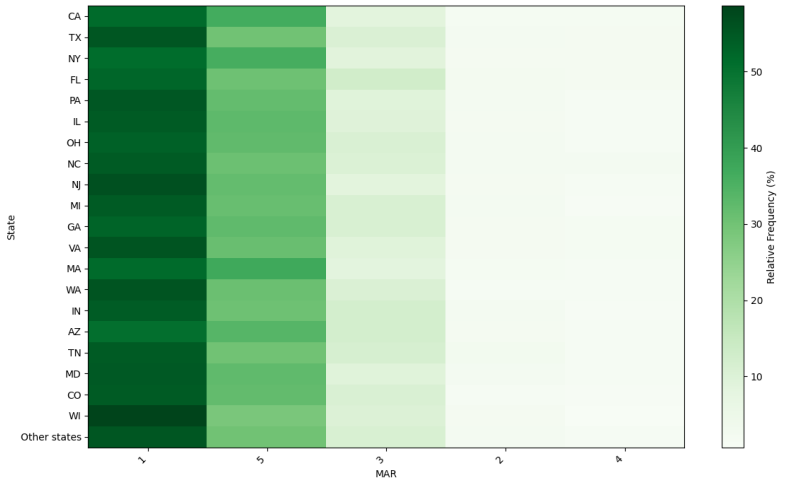


Figure 41: Distribution of marital status across individuals in the dataset.

RELP

Figure 42 and 43 present the distribution of RELP, which encodes the relationship of each individual to the household reference person. Categories include Father or mother (29), Grandchild (30), and other household members (codes 20–38).

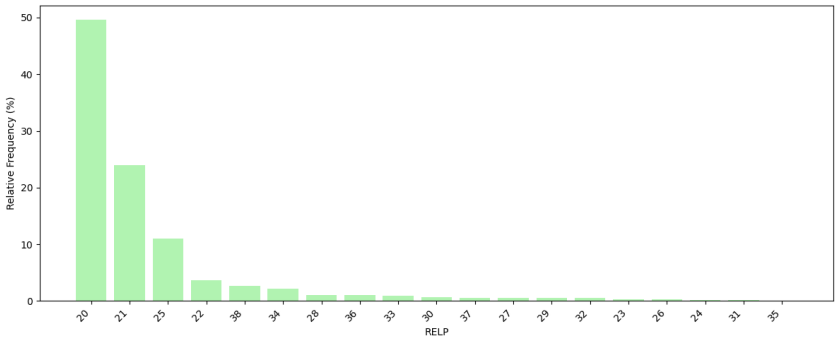


Figure 42: Distribution of relationship to householder across individuals, per state.

The barplot (Figure 42) reveals a strongly skewed distribution: RELP = 20 represents around 50% of the dataset. Frequencies then decline sharply: between 21 and 25 RELP shows moderate frequencies, whereas codes > 25 (e.g., parents, grandchildren) are relatively rare.

The heatmap (Figure 43) further substantiates this dominance, with consistently darker shades along the RELP = 20 column across all states, indicating low regional variation.

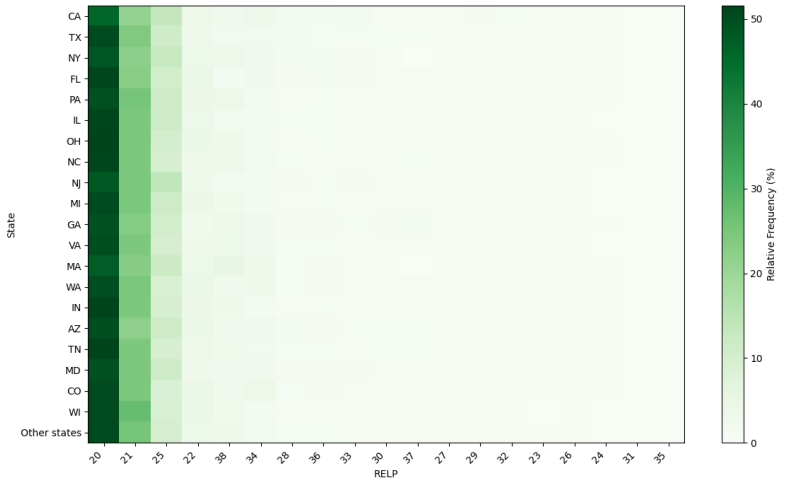


Figure 43: Distribution of relationship to householder across individuals in the dataset.

4.2.4 Correlation Analysis (Adult)

Having described the individual variables in the dataset, the analysis now turns to the relationships between them. Correlation is a fundamental statistical tool for assessing the degree of linear association between pairs of variables.

Given that much of the dataset is composed of categorical features (Table 2), these features were transformed into numerical values to enable correlation analysis. All categorical variables were transformed accordingly: binary variables were mapped to integers, one-hot for nominal variables with low cardinality, and ordinal encoding for ordinal variables. For the high-cardinality variables POBP and OCCP, two encoding strategies were tested, frequency encoding and target encoding, to evaluate their effect on correlation patterns.

For each encoded variable, the mean of the absolute correlation coefficients with all other variables was computed. Only variables whose mean absolute correlation exceeded 0.3 (indicating at least a moderate overall association) were retained in the correlograms. Moreover, only strongly correlated pairs (absolute correlation above 0.7) are reported, except for trivial self-pairs, whose correlation is always 1 and thus omitted for clarity.

Figure 44 presents the results obtained with frequency encoding for POBP and OCCP, where each category is replaced by its relative frequency in the dataset. Correlations remain moderate, capturing marginal distributional patterns without artificially inflating relationships between variables. Figure 45 shows the results obtained with target encoding, where each category is replaced by the mean value of the target variable across all instances in that category. Correlations appear stronger and more concentrated, particularly for income-related variables, but

this inflation may bias downstream methods.

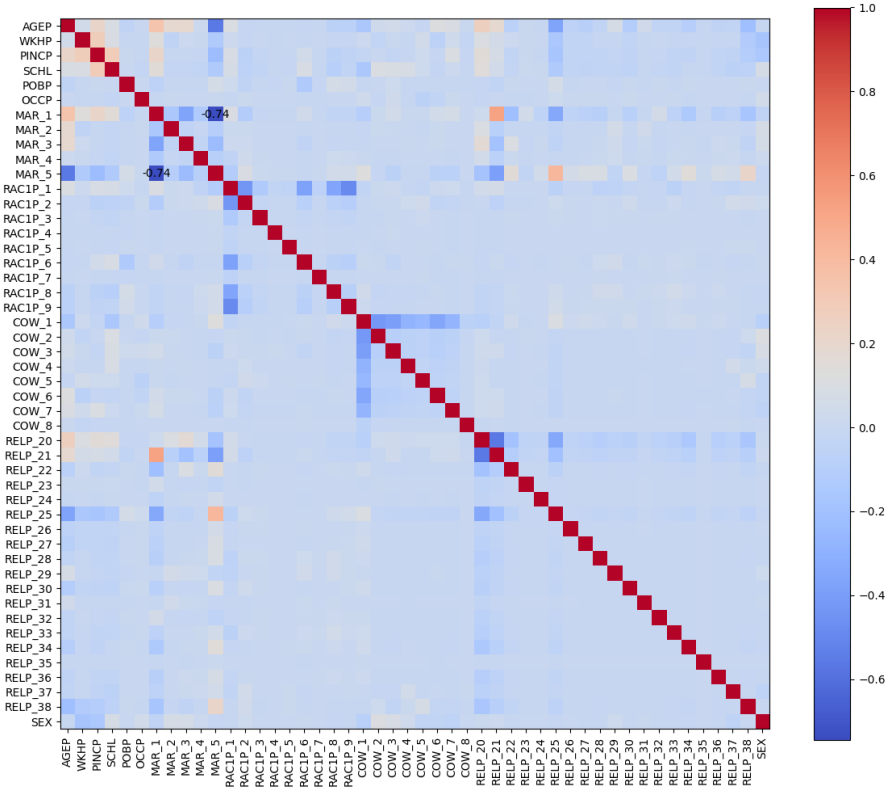


Figure 44: Correlation analysis, with frequency encoding applied to POBP and OCCP.

In summary, frequency encoding provides broader but less concentrated correlations, reflecting natural associations in the data, whereas target encoding amplifies correlations that may not be intrinsic. For downstream mathematical analyses, including cosine similarity, frequency encoding is therefore preferred.

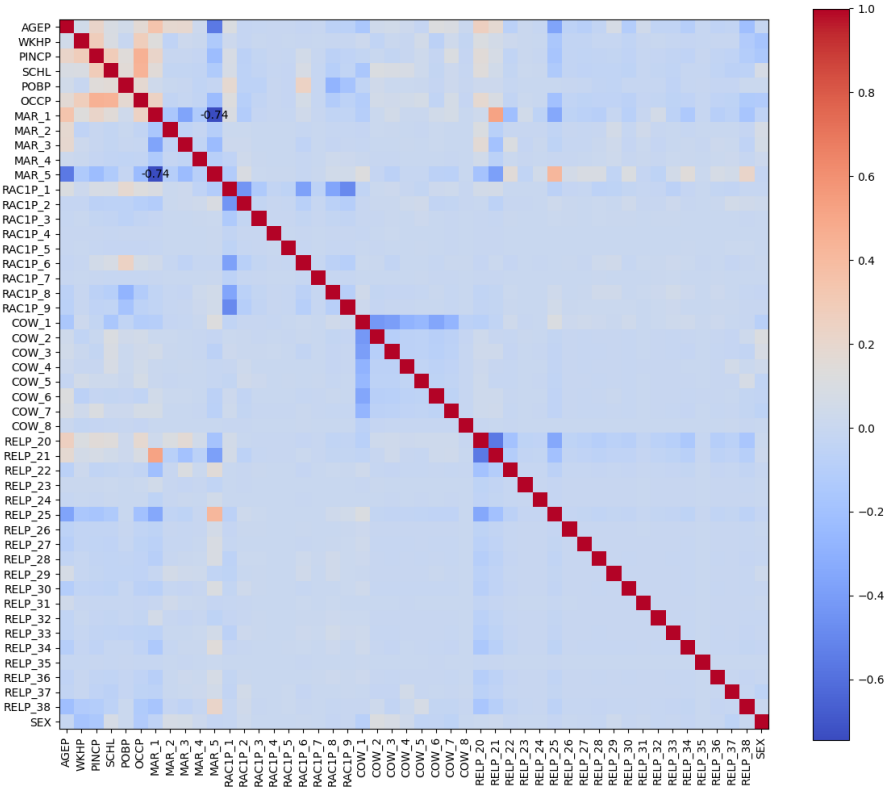


Figure 45: Correlation analysis, with target encoding applied to POBP and OCCP.

The final encoding strategy applied to all variables is summarized as follows:

1. SEX was transformed into integers, mapping "1" to 1 and "2" to 2.
2. MAR, RAC1P, COW, and RELP were one-hot encoded.
3. SCHL, given its ordinal nature (see Section 4.2.3), was transformed using an ordinal encoder.
4. POBP and OCCP, given their high cardinality (see Sections 4.2.3 and 4.2.3), were frequency encoded.

For numerical features, standard scores were used; that is, each value was subtracted by the mean and divided by the standard deviation, standardizing their magnitude for correlation computations.

4.2.5 Sent140

The analysis now shifts to the Sent140 dataset. Figure 46 displays the distribution of tweet sentiments per client in the Sent140 dataset, separated into negative (red) and positive (blue) tweets. Each bar represents a different client in the FL setup, with tweets stacked by sentiment label. The overall number of tweets per client remains within a relatively narrow band (approximately 1,000 to 1,600 tweets), and although there is some variation in sentiment proportions, most clients display a nearly balanced distribution. Importantly, the absolute dataset sizes in Sent140 are substantially smaller than those observed in Adult, and much more uniform. Unlike the Adult dataset, where a few clients hold over 100,000 examples and others fewer than 20,000 (check Figure 20), the Sent140 dataset does not exhibit such extreme concentrations. This suggests that the heterogeneity in data volume across clients is relatively limited in Sent140.

This pattern is further illustrated in Figure 47, which presents the CD of dataset sizes across clients. In Sent140, the disparity between the smallest and the largest client is relatively modest, with dataset sizes ranging from approximately 900 to 1,600 tweets, a factor of less than 2. In contrast, as previously noted, the Adult dataset exhibits a far greater imbalance, exceeding a tenfold difference between clients. Moreover, around 80% of the Sent140 clients possess dataset sizes up to roughly 1,300 tweets, with the remaining 20% holding between 1,300 and 1,600. This contrasts sharply with the Adult distribution, where the top 20% of clients dominate in terms of volume, managing datasets of 40,000 to nearly 160,000 data points. Such differences underscore the relatively uniform distribution in Sent140, which stands in stark opposition to the heavy-tailed distribution observed in Adult, resulting from an artificial construction.

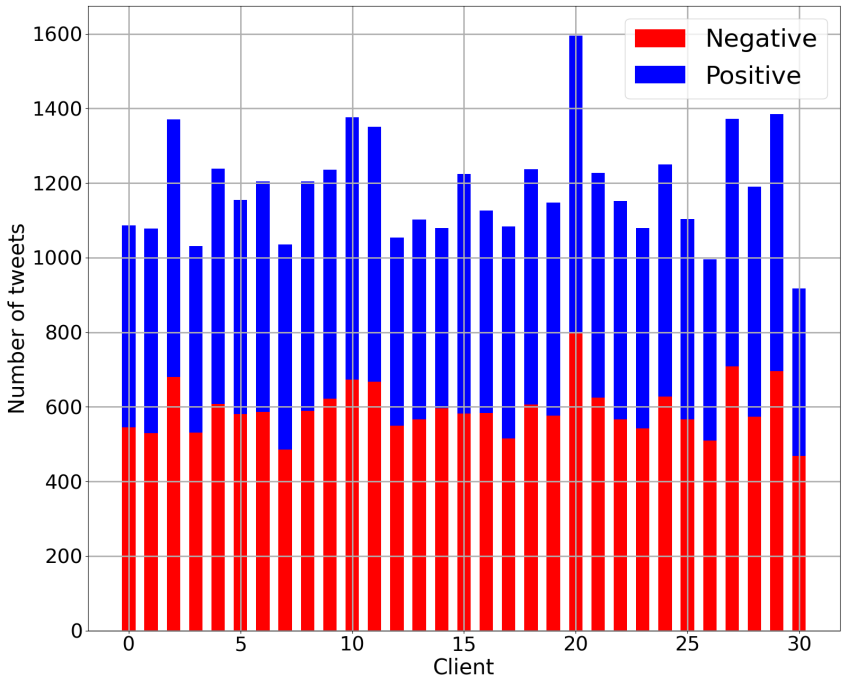


Figure 46: Proportion of individuals with positive and negative sentiment label, across clients.

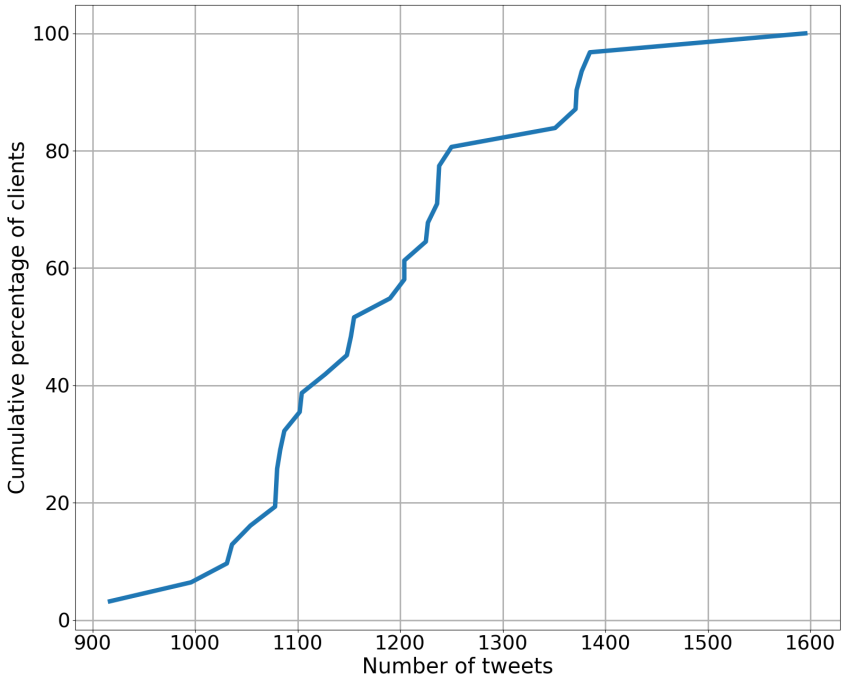


Figure 47: CD of dataset sizes across clients in the Sent140 dataset.

4.3 Base model

The experimental evaluation in this work is grounded on a neural network model, adapted to input features and classification goals.

4.3.1 Model components

The models for both the Adult and Sent140 datasets use common activation and loss functions, but specific optimizers, detailed below.

Activation and Loss Functions

ReLU Activation: For the hidden layers across both models, the Rectified Linear Unit (ReLU) activation function is used. It is defined as:

$$\text{ReLU}(x) = \max(0, x)$$

Sigmoid Activation: The output layer employs the sigmoid, σx, to map the final linear output to a probability between 0 and 1, suitable for the binary classification task:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Binary cross-entropy loss (BCE): Model training uses the binary cross-entropy Loss as the objective function, which measures the difference between the true label (y) and the predicted probability ($\hat{y}$):

$$L_{\text{BCE}}(y, \hat{y}) = -\frac{1}{N} \sum_{i=1}^{N} [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]$$

where N is the **total number of samples** being evaluated.

Adam Optimizer Formulation

The model for the Adult dataset was trained using the **Adam (Adaptative Moment Estimation) optimizer** (Kingma and Ba, 2015), an adaptation of SGD known to achieve faster convergence. Adam dynamically adjusts the learning rate for each parameter using exponential moving averages of the past gradients.

For simplicity in the Adam formulation, we define the local gradient g_t at step t as:

$$g_t = \nabla \ell\big(W_i^{t,k};\, \xi_i^{t,k}\big)$$

where $\xi_i^{t,k}$ is the mini-batch sampled from client i's local data at step k.

Adam's weight update rule is then defined as:

$$W_{t+1} = W_t - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{4.1}$$

The first moment estimate (m_t), which captures the momentum, is calculated using an exponential moving average over the current and past gradients:

$$m_t = \beta m_{t-1} + (1 - \beta) g_t \tag{4.2}$$

where $\hat{m}_t$ and $\hat{v}_t$ are the bias-corrected first and second moment estimates, respectively, and β is the decay rate.

Accuracy

In both datasets, accuracy was the evaluation metric. Accuracy is defined as the proportion of correctly classified instances:

$$\text{Accuracy} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}_{\{y_i = \hat{y}_i\}}$$

where N is the **total number of samples** being evaluated, $\mathbf{1}_{\{\cdot\}}$ is the indicator function (Section 7.5 shows the mathematical definition), y_i is the true label, and $\hat{y}_i$ is the predicted label (after thresholding the sigmoid output).

4.3.2 Adult

For the Adult dataset, a fully connected feedforward neural network was defined, with the following architecture:

- Input layer with 10 features;
- Hidden layers: 5 dense layers with 512, 256, 128, 64, and 32 units respectively, all with ReLU activation;
- Output layer: 1 neuron with a sigmoid activation, producing a probability estimate for the binary classification task.

The model was trained in a federated setting using the Adam (Adaptative Moment Estimation) optimizer (Kingma and Ba, 2015), an adaptation of SGD (Equation 2.1) which is known to achieve faster convergence. The training used a learning rate (α) of 0.001 and binary cross-entropy loss. The FL setup was as follows:

- Number of rounds: 50;
- Number of clients: 50 (one per U.S. state);
- Local training: 5 epochs per round;
- Batch size: 64.

The classic, centralised learning model was done with the the same parametrisation on the feedforward neural network but instead of 5 epochs, 250 epochs were used.

4.3.3 Sent140

For the Sent140 dataset, the model was adjusted to handle pre-processed 200-dimensional GloVe embeddings (Pennington et al., 2014) of tweet text. The architecture is composed of:

- Input layer with 200 features;
- Hidden layers: 3 dense layers with 128, 86, and 30 units respectively, all with ReLU activation;
- Output layer: 1 neuron with sigmoid activation, corresponding to the binary sentiment classification.

Training was performed using the SGD optimizer with a 0.001 learning rate, using binary cross-entropy loss.

The centralised learning model was done with the the same parametrisation on the feedforward neural network but instead of 5 epochs, 310 epochs were used.

The FL setup consisted of:

- Number of rounds: 50;
- Number of clients: 31;
- Local training: 10 epochs per round;
- Batch size: 64.

Owing to unsatisfactory performance, the Sent140 dataset was not used in the main experiments, but it is reported here for completeness.

4.3.4 Summary of Key Training Parameters

Table 3: Summary of model architectures and training parameters.

Parameter	**Adult**	**Sent140**
Input dimension	10	200
Hidden layers	5 (512–32 units)	3 (128–30 units)
Output layer	1 (sigmoid)	1 (sigmoid)
Loss function	Binary cross-entropy	Binary cross-entropy
Optimizer	Adam	SGD
Learning rate	0.001	0.001
Batch size	64	64
Local epochs	5	10
Number of clients	50	31
Number of rounds	50	50

4.4 Evaluation metrics

To quantitatively assess the benefits of FL over traditional, classic learning, accuracy gain was adopted as the primary evaluation metric, capturing the accuracy improvement achieved when clients participate in collaborative training compared to relying solely on local data.

Let W_{FL} denote the model weights obtained through a FL process FL, and W_L denote the weights resulting from isolated (local-only) training L. The **accuracy gain** for a given client i is then defined as:

$$\text{Accuracy Gain}_i = \text{Accuracy}_i(W_{\text{FL}}) - \text{Accuracy}_i(W_{\text{L}})$$

where $\text{Accuracy}_i(W_{\text{FL}})$ and $\text{Accuracy}_i(W_{\text{L}})$ denote, respectively, the accuracy of the model obtained at the final round of federated training, per client, and the accuracy of each local model obtained through local training. Note that in this setup, each client trains a model independently using only their own data.

The interpretation is straightforward:

- if Accuracy Gain$_i > 0$, C_i benefits from participating in the FL experiment, as it delivered better accuracy results compared to the local learning alternative;
- on the other hand, if Accuracy Gain$_i < 0$ the opposite is verified: C_i does not benefit for being part of the FL experiment, as it degraded its performance.

To evaluate the impact of the proposed adaptations to the standard FL scheme, one additional metrics is introduced, called **filtered gain**. **Filtered gain** assesses the effectiveness of the adapted FL scheme, FedFil, relative to local learning being defined as:

$$\text{Filtered Gain}_i = \text{Accuracy}_i(W_{\text{FedFil}}) - \text{Accuracy}_i(W_{\text{L}})$$

The interpretation of filtered gain is analogous to that of accuracy gain: if Filtered Local Gain$_{\text{i}} > 0$, then client i also benefits from being part of the FedFil experiment.

Beyond individual client accuracy, it is also important to assess the **generalization ability** of the models, that is, their capacity to perform well on data distributions not necessarily observed during local training. This is particularly relevant in the presence of significant heterogeneity across clients, as illustrated in Figure 20, where data distributions vary considerably across states.

To simulate a realistic scenario in which a model is applied to data from a different region, for instance, a person who has moved from one state to another, a global test set is introduced. This set aggregates data points from all participants, enabling the evaluation of model accuracy beyond each client's local distribution.

Each client's dataset is partitioned as follows:

- **10% local test data**: a hold-out set used to measure accuracy on the participant's own distribution;
- **10% global or general test data**: another hold-out subset, combined across all participants to form a shared global test set;
- **80% training data**: used in both federated and local training procedures.

This setup allows for the evaluation and comparison of accuracy gains not only on local data, but also on composite (global) data. In this context, global versions of the previously introduced metrics — accuracy gain, filtered gain — can be defined, where the test accuracy is computed on the global test set instead of the local one. This provides a measure of whether the federated approach and the proposed filtering strategies contribute to improved cross-distribution generalization. In classic local learning, each client trains an independent model using only their own data, without access to information from other states. To evaluate accuracy on local test data, the model trained by client is assessed using the 10% hold-out subset corresponding to their own distribution. For example, to compute the accuracy for AL (Alabama), the model trained exclusively on data from AL is evaluated on the local test set containing data data from AL. For global test evaluation, the same locally trained model is assessed on the shared global test set, which aggregates data from all states. This setup allows for measuring both local accuracy and cross-distribution generalization for models trained isolatedly.

4.5 Comparative baselines

As mentioned in Chapter 1, this dissertation is a continuation of a previous project. In that project, other comparative learning schemes were implemented. These include:

- FL with LDP (FedDP) (Naseri et al., 2020): To assess the impact of privacy protection, a FL scheme with DP is considered. In this scheme, it was introduced white noise following a Gaussian distribution to the model updates computed through a standard FL corresponding to an epsilon parameter set to 0.5;
- Private Personalized Layers (FedPer) (Arivazhagan et al., 2019): This FL scheme uses private personalized layers. In such a scheme, only the lower layers of the model (capturing coarse-grained information) are exchanged with the aggregation server while the upper layers of the model (capturing fine grain information) are personalized and kept private for each user. In the evaluation presented here, the last three upper layers (out of the six layers of the fully connected model) were considered private and remained on the participant, being personalized with its data.

In this dissertation, these models were initially tested on the Sent140 dataset to ensure code correctness and reproducibility. However, they were not the major focus of further experimentation nor comparison, as they had already been studied in depth in the previous stage of the project and the Sent140 dataset was subsequently set aside. The emphasis of this work lies instead on the development and evaluation of the **FedFil** scheme.

A more detailed explanation of FedDP and FedPer was presented earlier, respectively in Sections 2.3.1 and 2.3.2.

Chapter 5

Experimental results

5.1 Initial Considerations

This chapter presents the experimental results obtained in this work. Two datasets were initially considered for experiments: the Adult dataset and the Sent140 dataset. However, only the Adult dataset was retained for the main experiments due to performance limitations observed during preliminary testing with Sent140 (presented in Appendix 7.2).

Given the accuracy issues observed with Sent140, the subsequent analyses focus exclusively on the Adult dataset. The following sections provide a detailed exploration of baseline and results of the proposed model, along with the evaluations of the proposed intervals. This includes:

- Baseline FL performance on Adult;
- Evaluation of intervals and experimental schemes;
- Studying what impacts FedFil performance.

Three levels of fine-tuning were employed to study the impact of personalization: a low level with 20 epochs, a medium level with 50 epochs, and a high level with 100 epochs. To distinguish between accuracy gains measured on local versus global test data (cf. Section 4.4), the following terminology is introduced:

- **Local accuracy gain**: accuracy gain computed on each client's local test data;

- **General accuracy gain**: accuracy gain computed on the global test set.

The same decomposition applies to **filtered gain**, yielding **Filtered local gain** and **filtered general gain**.

By convention, when both **filtered gain** and **accuracy gain** are presented together, the general term **accuracy gain** is used, as **filtered gain** represents a particular case of **accuracy gain**. Additionally, terms such as **local accuracy** and **general accuracy** are used interchangeably to denote accuracy measured on local test data and global test data, respectively. Finally, throughout this work, the terms **baseline FL model** or simply **baseline** are used interchangeably with **standard FL model** to refer to the conventional FL setup using FedAvg.

5.2 IQR filtering strategy

To evaluate the impact of applying a **IQR filtering strategy** in FedFil, we analyse its effects both globally and per client. At the global level (Figure 48a), the data removal ratio varies substantially from 3% to nearly 11%, with around 60% of clients below 5% and roughly 20% between 6–11%. Client-level analysis (Figure 48b) reveals no clear link between dataset size and filtering proportion, as variability is high across participants.

Accuracy gain impact without fine-tuning.

Figure 49a compares the accuracy gain of FedFil (IQR filtering) with the standard FL model (FL with FedAvg). Without local fine-tuning, filtering tends to reduce both local and general accuracy gains: around 60% of clients show negative local filtered gains and 40% negative general ones, compared to only about 20% local and general accuracy gains under the baseline. Despite this, most clients achieve higher general filtered gains than local ones, consistent with baseline trends.

Accuracy gain impact with fine-tuning.

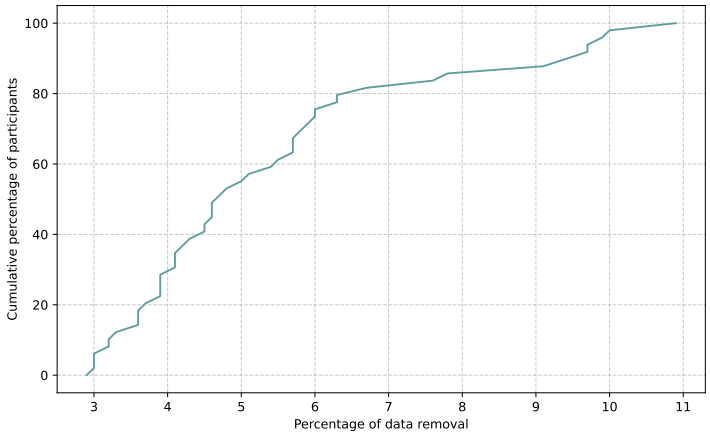
Figure 49b compares the accuracy gain of FedFil (IQR filtering) after fine-tuning with the baseline. Personalization improves local accuracy to an extent: with 20 epochs, approximately 60% of clients surpass the local accuracy gains observed under the baseline. However, extending fine-tuning to 50 and 100 epochs does not yield further improvements, as local filtered gains stagnate and generalization deteriorates accordingly. As in Figure 49a, for the three fine-tuning settings, most clients achieve higher general filtered gains than local ones, consistent with baseline as well.

Client-level Effect of FedFil.

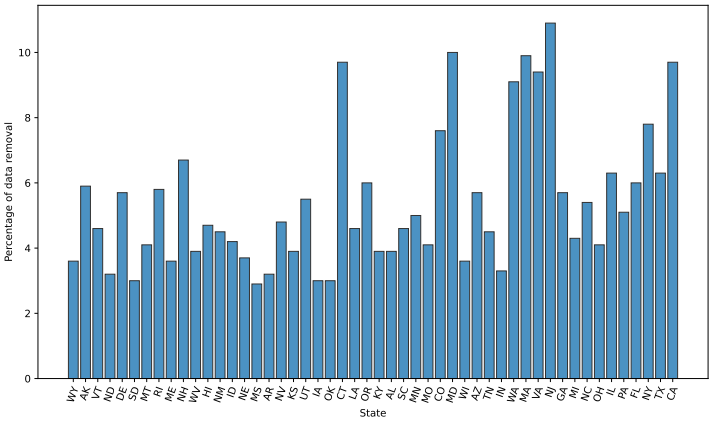
We now focus on how FedFil affects individual clients under the same IQR filtering scheme. Figure 50a shows the **maximum local filtered gain** obtained per client under FedFil. This gain represents the maximum improvement achieved either during the first phase of FedFil (without fine-tuning) or during the second phase (using 20, 50, or 100 fine-tuning epochs) over classical learning. The black dashed line in the figure represents the relative size of each client's local dataset within the global dataset after the IQR filtering strategy is applied.

It is possible to see that most clients manage to benefit when testing on their own data (i.e., the local filtered gain is positive). Notably, **smaller clients seem to benefit proportionally more**, as demonstrated by states such as MT (Montana) or ID (Idaho).

Figure 50b shows the same metric, but for the **general filtered gain**. Here, we observe considerably more variability: there are more counts of (and more pronounced) negative values and fewer counts of (and less pronounced) positive values. This leads to a bigger share of clients whose generalization accuracy is not as good as that obtained using classical learning. However, **positive generalization ability is still seen more frequently among clients with smaller local datasets** than among their counterparts with bigger local datasets.

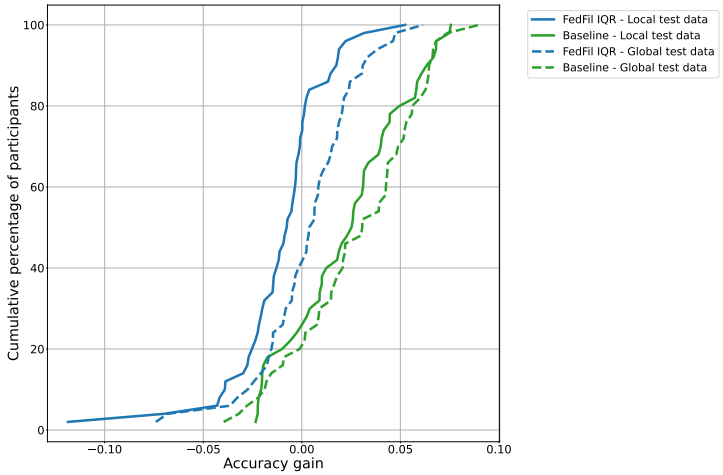


(a) CD of data removal percentage using IQR interval.

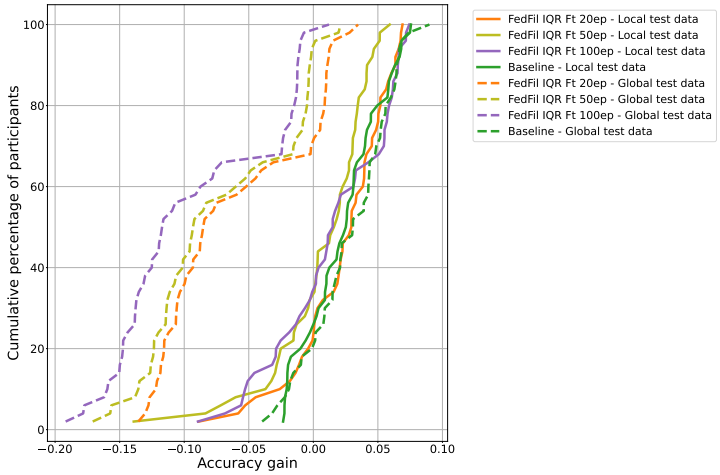


(b) Data removal percentage per client, using IQR interval, ordered by dataset size.

Figure 48: Data removal proportions under the global IQR interval, showing both the overall distribution (a) and the per-client results (b).

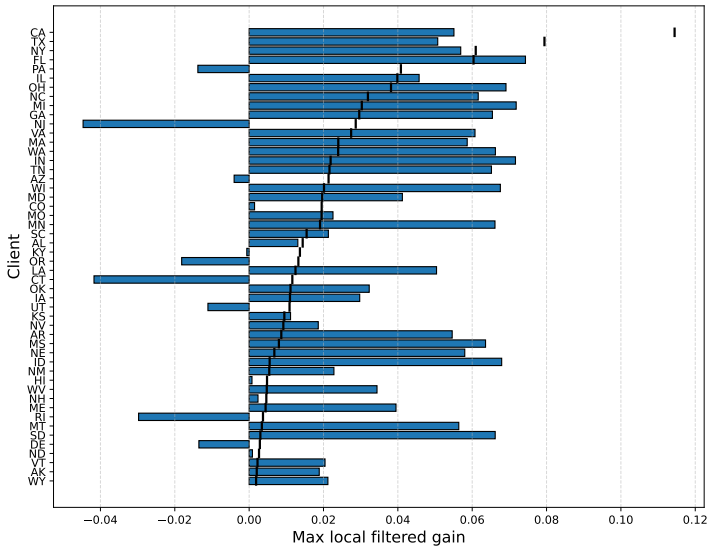


(a) Filtered gain with IQR method.

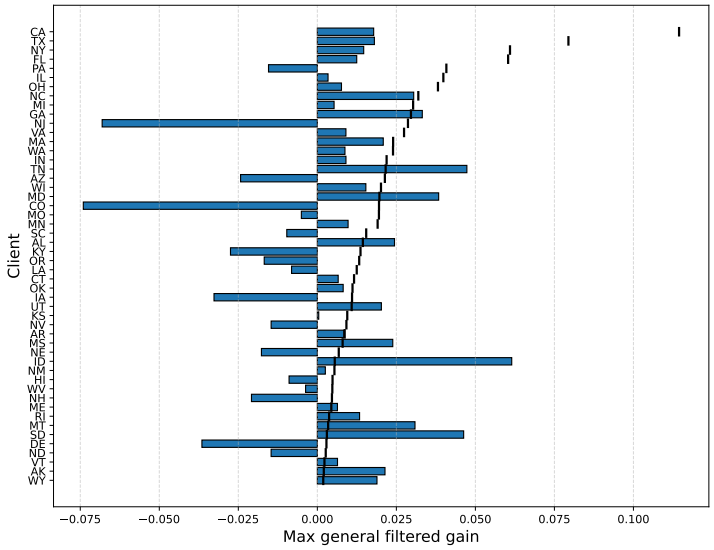


(b) Filtered gain with IQR method, after fine-tuning.

Figure 49: CD of client-wise filtered local gain with IQR method. Solid lines represent filtered local gain; dashed lines correspond to filtered general gain. Subfigure (a) shows baseline and FedFil with IQR interval; subfigure (b) includes fine-tuning (Ft) for different epochs (ep), applied on FedFil with IQR interval.



(a) Maximum local filtered gain with IQR method, for each client.



(b) Maximum general filtered gain with IQR method, for each client.

Figure 50: Subfigure (a) shows the maximum local filtered gain for each client using the IQR method; subfigure (b) shows the maximum general filtered gain for each client. The black dash indicates the proportion of each client's local data in the global dataset after IQR filtering.

5.3 3SR filtering strategy

To evaluate the impact of applying a **global 3SR filtering strategy** in FedFil, we analyse its effects both globally and per client, following the same structure as for the IQR-based analysis in Section 5.2.

At the global level (Figure 51a), the data removal ratio ranges from approximately 0.5% to 3.5%, with 40% of clients below 1.5% and roughly 20% above 2%. Client-level analysis (Figure 51b) shows no clear relationship between dataset size and filtering proportion, displaying less dispersion than the IQR approach and substantially lower overall removal rates. Overall, the 3SR method yields smoother and less pronounced reductions than the global IQR strategy (Figures 48a and 48b).

Accuracy impact without fine-tuning. Figure 52a compares FedFil (global 3SR filtering) with the standard FL model without fine-tuning. Both local and general filtered gains tend to exceed those observed under the baseline, contrasting with the degradation seen under IQR interval (Section 5.2). Around 20% of clients still show negative filtered gains, similar to the baseline, but most clients experience higher filtered than accuracy gains.

Accuracy impact with fine-tuning.

As in Section 5.2, moderate fine-tuning (20 epochs) enhances local accuracy while extended fine-tuning increases variability and weakens generalization. This can be seen at Figure 52b: all fine-tuned clients show higher local filtered gains than local accuracy gains from the baseline FL model, while general filtered gains worsen with additional fine-tuning: fewer than 40% of clients outperform the baseline after 20 epochs, and this proportion becomes negligible at 50 and 100 epochs.

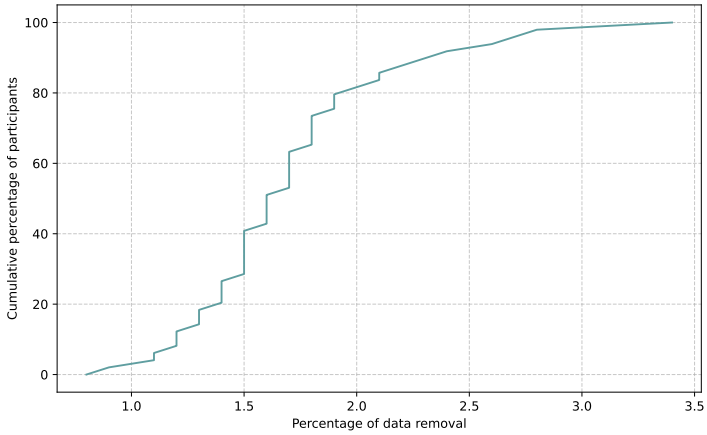
Client-level Effect of fine-tuning.

We now focus on how FedFil affects individual clients under the same 3SR filtering scheme.

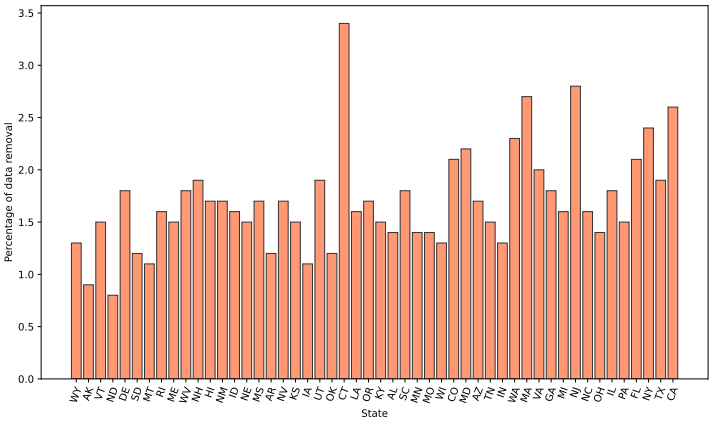
Figure 53a and Figure 53b respectively show the maximum local and general filtered gain obtained per client under FedFil. This gain represents the maximum improvement achieved either during the first phase of FedFil (without fine-tuning) or during the personalization phase (using 20, 50, or 100 fine-tuning epochs) over classical learning. The black dashed line in the figure represents the relative size of each client's local dataset within the global dataset after the 3SR filtering strategy is applied.

As noted in Section 5.2, **smaller clients seem to benefit proportionally more** in both cases. However, under 3SR, FedFil achieves considerably better gains than with the IQR interval. Both when testing on their own data

or a general test set, 3SR presents a case where almost all clients have positive filtered gains, which are generally more pronounced and simultaneously less negative. This is especially significant in Figure 53b, where there's much less variability in values than what was observed with IQR.

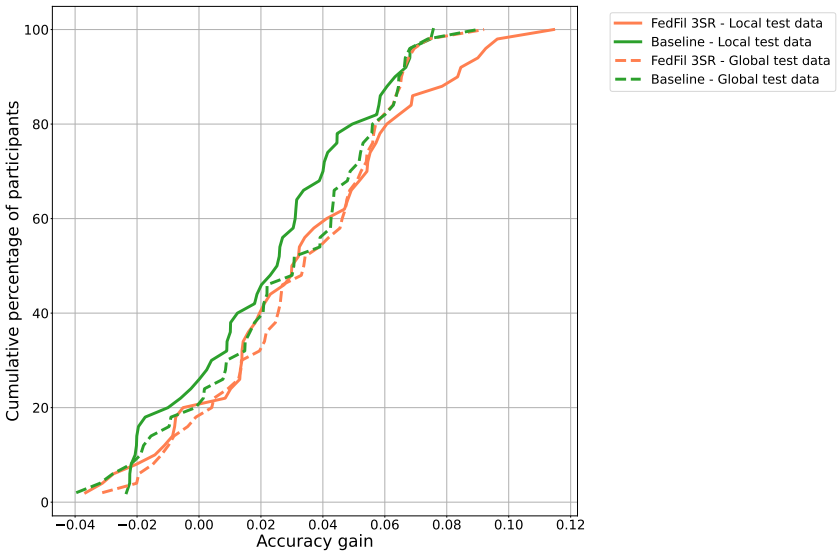


(a) CD of data removal percentage using global 3SR interval.

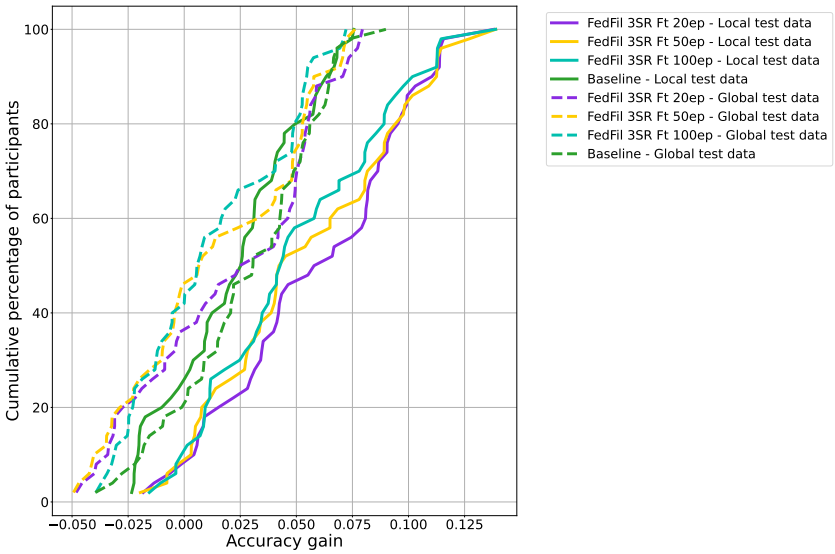


(b) Data removal percentage per client, using global 3SR interval, ordered by dataset size.

Figure 51: Data removal proportions under the global 3SR criterion, showing both the overall distribution (a) and the per-client results (b).

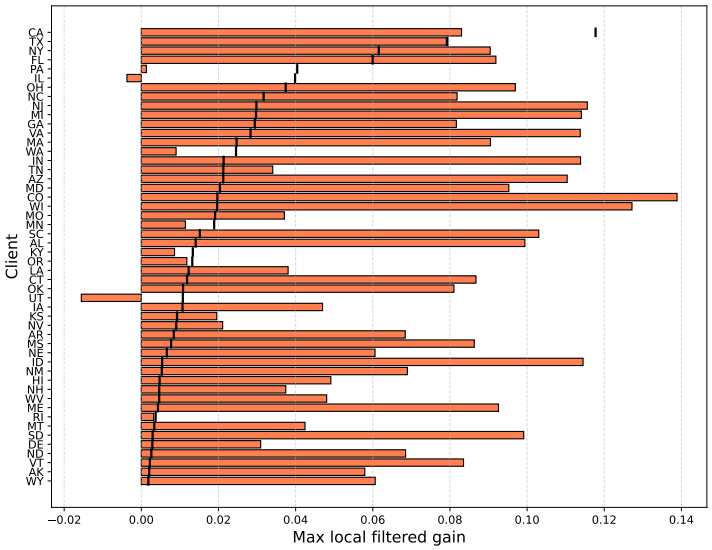


(a) Filtered gain with 3SR interval.

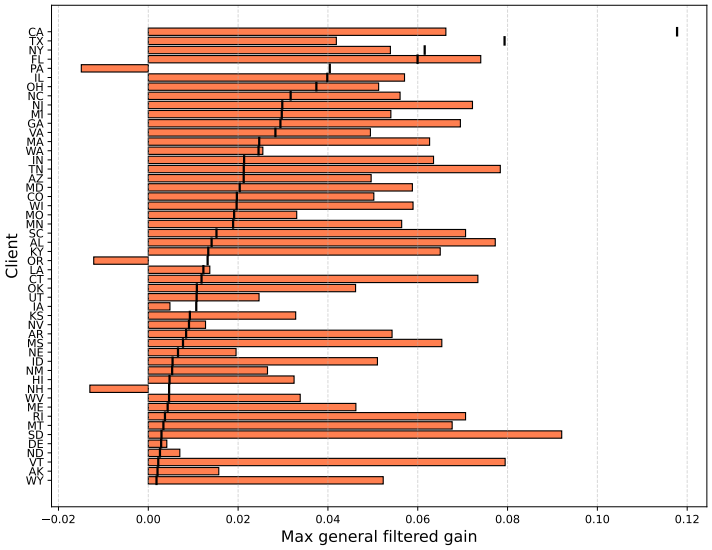


(b) Filtered gain with 3SR interval, after fine-tuning.

Figure 52: CD of client-wise filtered local gain with 3SR interval. Solid lines represent filtered local gain; dashed lines correspond to filtered general gain. Subfigure (a) shows baseline and FedFil with global 3SR interval; subfigure (b) includes fine-tuning (Ft) for different epochs (ep), applied on FedFil with global 3SR interval.



(a) Maximum local filtered gain with 3SR method, for each client.



(b) Maximum general filtered gain with 3SR method, for each client.

Figure 53: Subfigure (a) shows the maximum local filtered gain for each client using the 3SR method; subfigure (b) shows the maximum general filtered gain for each client. The black dash indicates the proportion of each client's local data in the global dataset after 3SR filtering.

5.4 CS filtering strategy

To evaluate the impact of applying a **CS filtering strategy** in FedFil, we analyze its effects both globally and per client, following the same structure used for the IQR-based analysis in Section 5.2 and the 3SR-based one in Section 5.3.

Two global data removal thresholds are tested, **1%** and **2.5%**, referred to as **CS (Thr 1%)** and **CS (Thr 2.5%)** respectively, to better understand the behavior of the CS filtering approach.

At the global level (Figure 54a), the data removal ratio in CS (Thr 1%) ranges from very low values up to around 2%, with more than 80% of participants falling below the global 1% threshold. For CS (Thr 2.5%), (Figure 58a), a similar pattern is observed, with most clients' data filtering remaining below the 2.5% global cutoff and only a few reaching values close to 5%.

Client-level analysis (Figures 54b and 58b) reveals a clearer relationship, albeit with some variability, between dataset size and filtering proportion under this strategy, compared to the previously discussed approaches (Figures 48b and 51b).

Accuracy impact without fine-tuning. Figure 55 compares FedFil (CS filtering) with 1% and 2.5% thresholds, with the baseline. Without fine-tuning, both local and general filtered gains tend to exceed those observed under the baseline, being generally higher in the case where less data is removed (the 1% global threshold). Fewer than 20% of clients exhibit negative filtered gains, indicating that most clients clearly outperform the baseline. The vast majority achieve higher filtered gains, particularly local ones, when compared to the baseline accuracy gains. General filtered gains are less pronounced.

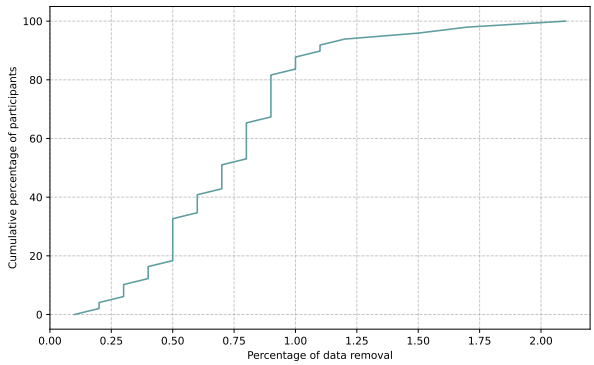
Accuracy impact with fine-tuning.

Under the CS (Thr 1%) configuration (Figure 56a), fine-tuning for 20 epochs is generally sufficient to enhance the filtered gains relative to the non-personalized model. Both the general filtered gains, where 20 epochs consistently outperforms other fine-tuning configurations, and the local filtered gains for most clients are either higher or only marginally lower than those achieved with alternative fine-tuning settings. For CS (Thr 2.5%) (Figure 56b), the results follow a similar trend, albeit less pronounced and exhibiting greater variability.

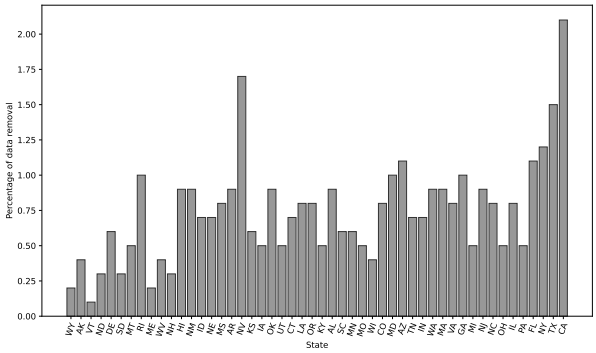
Client-level Effect of fine-tuning.

Similar to the IQR-based analysis (Section 5.2) and the 3SR-based analysis (Section 5.3), the CS (Thr 1%) and CS

(Thr 2.5%) strategies also exhibit proportionally greater local and general filtered gains for clients with smaller local datasets, client's local dataset proportions are marked by a black dash, as seen in Figures 57 and 59, respectively. The CS (Thr 1%) strategy yields a positive benefit for all clients, showing similar results in both scenarios: testing on their own data (Figure 57a) and testing on a general composite set ((Figure 57b)). The CS (Thr 2.5%) strategy shows **higher variability** across both local and general filtered gains compared to CS (Thr 1%). It is the only method that presents (marginally) more clients benefiting over classical learning in terms of generalization ability (i.e., general filtered gains, seen in Figure 59b) than in terms of local filtered gains (Figure 59a).



(a) CD of data removal percentage using CS, under a global 1% data removal threshold.



(b) Data removal percentage per client using CS, under a global 1% data removal threshold, ordered by dataset size.

Figure 54: Data removal proportions under CS (Thr 1%), showing both the overall distribution (a) and the per-client results (b).

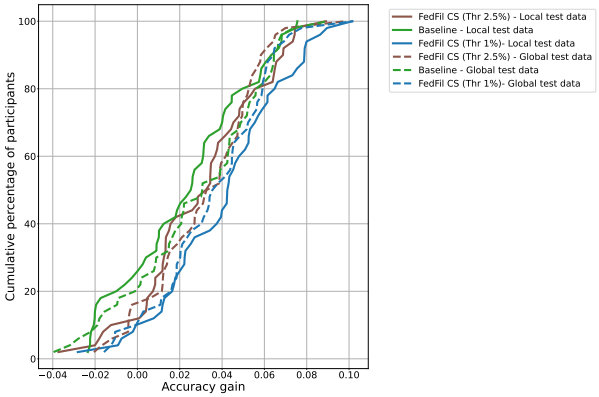
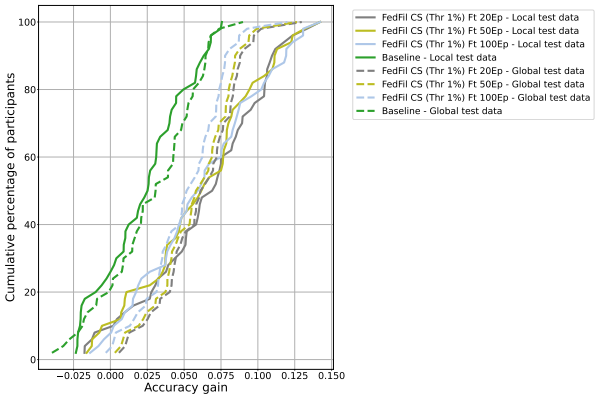
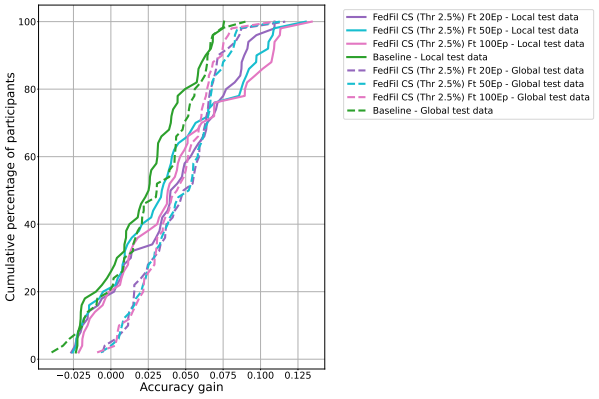


Figure 55: CD of client-wise filtered gains. Baseline and FedFil with cosine similarity (CS) at 1% and 2.5% global removal thresholds (Thr).

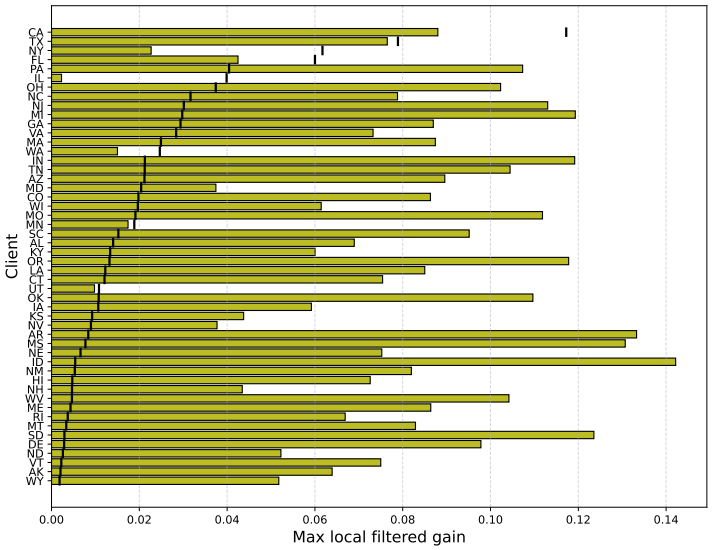


(a) Filtered gain with CS under a 1% global data removal threshold, after fine-tuning.

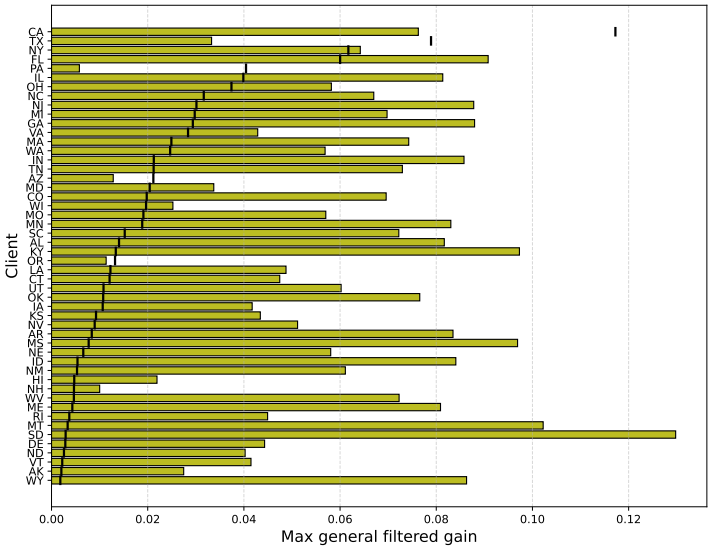


(b) Filtered gain with CS under a 2.5% global data removal threshold, after fine-tuning.

Figure 56: CD of client-wise filtered gains with fine-tuning (Ft) for different numbers of epochs (ep). FedFil with CS (Thr 1%) and CS (Thr 2.5%).

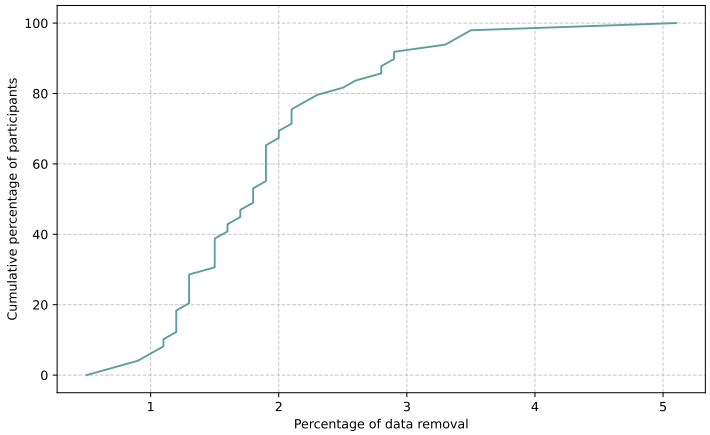


(a) Maximum local filtered gain with CS (Thr 1%) method, for each client.

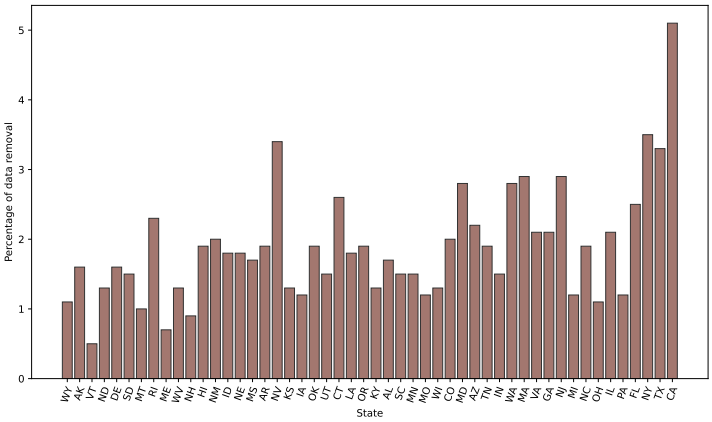


(b) Maximum general filtered gain with CS (Thr 1%) method, for each client.

Figure 57: Subfigure (a) shows the maximum local filtered gain for each client using the CS (Thr 1%) method; subfigure (b) shows the maximum general filtered gain for each client. The black dash indicates the proportion of each client's local data in the global dataset after CS (Thr 1%) filtering.

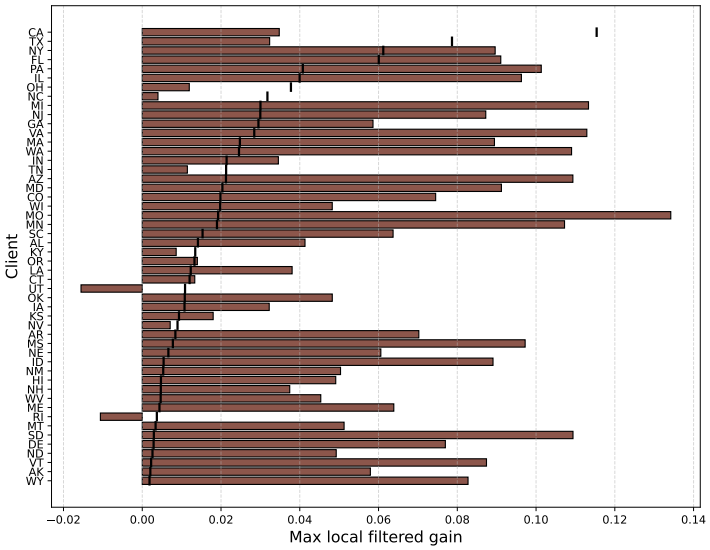


(a) CD of data removal percentage using CS, under a global 2.5% data removal threshold.

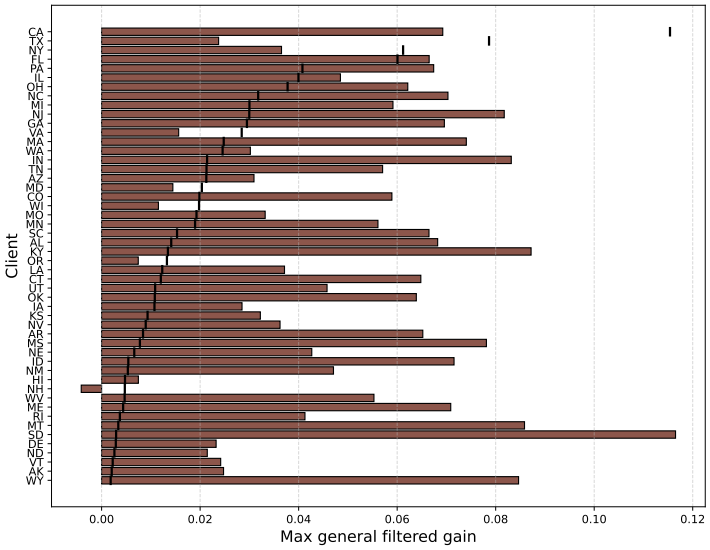


(b) Data removal percentage per client using CS, under a global 2.5% data removal threshold (Thr), ordered by dataset size.

Figure 58: Data removal proportions under the CS (Thr 2.5%) criterion, showing both the overall distribution (a) and the per-client results (b).



(a) Maximum local filtered gain with CS (Thr 2.5%) method, for each client.



(b) Maximum general filtered gain with CS (Thr 2.5%) method, for each client.

Figure 59: Subfigure (a) shows the maximum local filtered gain for each client using the CS (Thr 2.5%) method; subfigure (b) shows the maximum general filtered gain for each client. The black dash indicates the proportion of each client's local data in the global dataset after CS (Thr 2.5%) filtering.

5.4.1 Summary of results

To summarize all of the analyses, Table 4 presents a statistical summary of the maximum filtered gains across the implemented methods, using weighted statistics calculated as shown in Section 7.3. FedFil CS (Thr 1%) method demonstrates the highest overall performance, yielding the greatest weighted mean for both local (0.0743) and general (0.0619) filtered gains. This suggests that a selective filtering approach based on cosine similarity is highly effective. Conversely, the FedFil IQR method shows the weakest performance in generalization, with its weighted mean dropping significantly from 0.0423 (Local) to 0.0065 (General), indicating an overly aggressive filtering mechanism that removes too much data. In terms of stability, methods FedFil 3SR and FedFil CS (Thr 2.5%) achieve the lowest weighted standard deviations (0.0229 and 0.0221, respectively) for general filtered gains, highlighting their robustness and consistency when applied to diverse client data. For local filtered gains, the lowest variability is obtained with FedFil IQR (0.0317) and with FedFil CS (Thr 1%), whose variability is 0.0340.

The results further suggest that removing fewer datapoints, particularly in the case of CS (Thr 1%), yields the strongest performance in both local and general evaluations. This method also achieves the lowest coefficient of variation, a statistic that measures variability by dividing the weighted standard deviation by the weighted mean, for both local and general testing. This indicates that CS (Thr 1%) provides the most consistent and reliable improvements among all methods. On the other hand, IQR, the approach that removes the largest number of datapoints, shows the highest coefficient of variation, reinforcing the conclusion that it may be too stringent by removing too many datapoints.

Table 4: Statistical analysis of maximum filtered gains

Filtering Strategy	**Statistic**	**Local Filtered Gain**	**General Filtered Gain**
FedFil IQR	Weighted mean	0.0423	0.0065
	Weighted sd	0.0317	0.0240
	CV	0.749	3.692
FedFil 3SR	Weighted mean	0.0729	0.0509
	Weighted sd	0.0388	0.0229
	CV	0.532	0.450
FedFil CS (Thr 1%)	Weighted mean	0.0743	0.0619
	Weighted sd	0.0340	0.0247
	CV	0.458	0.399
FedFil CS (Thr 2.5%)	Weighted mean	0.0622	0.0521
	Weighted sd	0.0374	0.0221
	CV	0.601	0.424

Note: sd stands for standard deviation; CV stands for coefficient of variation.

Chapter 6

Conclusion and future work

6.1 Conclusion

This dissertation proposed FedFil, a two-stage scheme for cross-silo Federated Learning (FL) aimed at enhancing the learning gain of participating clients. FedFil performs a selective filtering of local training data prior to global aggregation, followed by a personalization step, with the overarching goal of improving both local and general accuracy in heterogeneous environments.

Across all experiments, FedFil proved capable of increasing the learning gain for FL participants, even in the presence of significant data heterogeneity. A particularly noteworthy finding is that clients with smaller local dataset sizes exhibited proportionally larger benefits, while clients with larger datasets did not dominate the accuracy improvements. This finding also suggests that factors beyond dataset size, such as the alignment with the global distribution, may influence the effectiveness of collaboration.

Four filtering strategies were systematically evaluated: IQR (based on the interquartile range), 3SR (based on the three-sigma rule) and CS with thresholds 1% and 2.5% (based on cosine similarity), along with two quantile estimation procedures. Although the latter showed mixed performance (sometimes producing accurate approximations, other times large errors), the overall experimental evaluation confirmed that filtering based on cosine similarity was the most effective approach for improving both local and global accuracy gains. In contrast, IQR was shown to be the worst performing approach.

Taken together, these findings show that modifying the local learning scheme through selective data filtering can meaningfully improve global accuracy gains, answering the third research question affirmatively. FedFil thus provides

a principled mechanism for improving cross-silo FL, mitigating heterogeneity-induced instabilities, and maintaining an equitable distribution of benefits across clients. Nonetheless, the scope of this work is not exhaustive, and several avenues remain open for further exploration.

6.2 Future work

Several directions for future research emerge naturally from this work.

6.2.1 Alternative AD methods

Beyond the statistical and threshold-based rules employed in this work, future research should explore more mechanisms for identifying anomalies. These methods can be broadly categorized as enhanced metrics and dedicated AD models.

Similarity and distributional approaches. Cosine similarity proved effective, but alternative measures could capture deeper structural relationships within the data or quantify distributional differences more explicitly:

- **Gower distance** (Gower, 1971): suitable for mixed-type data, combining numerical and categorical features without heavy preprocessing;
- **Mahalanobis distance**: accounts for feature covariance and identifies outliers relative to the global structure of the data, being appropriate when correlations between features are meaningful;
- **Distributional measures**: such as **Kullback-Leibler divergence**, enabling filtering intervals based on differences between local and global distributions.

Model-based AD. FedFil could also benefit from integrating established AD algorithms:

- **Isolation Forest**: an unsupervised tree-based algorithm that isolates anomalies by recursively partitioning the feature space;

- **Local Outlier Factor**: a density-based unsupervised method that identifies anomalies through local neighborhood comparisons;
- **One-Class Support Vector Machine**: an unsupervised algorithm learns a decision boundary that encapsulates normal data, identifying any deviations outside this region as outliers or anomalies.

6.3 Types of intervals and parameter estimation

As discussed in Section 3.2.1, parameter estimation could be made more robust to adversarial clients through secure aggregation protocols. Furthermore, collaborative quantile estimation remains a bottleneck when client distributions differ significantly, as observed in this work where the proposed estimators produced mixed results. A promising direction is the method outlined in Cai et al. (2025), which proposes an LDP-based quantile estimation optimized with SGD.

Filtering intervals could also be redesigned more flexibly—for instance, by applying intervals only to a subset of clients, or by clustering clients based on similarity (Section 6.2.1). Another direction involves borrowing ideas from the data-selection literature (e.g. Li et al. (2021a); Tuor et al. (2020)), which uses benchmark models or scoring functions to assess data samples' utility before and during training. Additionally, recent work has explored game-theoretic mechanisms to incentivize rational clients to share high-quality data in FL (Karimireddy et al., 2022), emphasizing the prioritization of certain data samples for global aggregation.

6.4 Security and privacy considerations

Although this work focused primarily on utility, filtering mechanisms also raise important security and privacy considerations:

- Can filtering mitigate poisoning attacks in FL, as suggested by AD applications in Mothukuri et al. (2021); Sagar et al. (2022)?
- Does filtering amplify or reduce privacy risks for individual clients? Evidence from Jourdan et al. (2021)

suggests that, if local specificities (e.g., outliers or anomalous datapoints) are not leveraged during training, privacy may improve because other participants are less able to infer such information.

Given the well-known privacy–utility trade-off, whereby increasing privacy protections (e.g., through noise addition) typically reduces model utility, the fact that FedFil has been shown to preserve utility suggests that it may provide a promising pathway to improving both privacy and utility simultaneously.

6.5 Fairness evaluation

Clients with smaller local datasets disproportionately benefited from FedFil. Fairness-oriented metrics could more rigorously assess how equitable this distribution of gains is, and whether FedFil reduces or amplifies disparities relative to other FL schemes.

6.6 Characterizing the nature of removed data

Understanding why certain datapoints are removed can provide valuable insights into:

- **local vs. global differences**: whether removed datapoints share similar statistical properties across clients or deviate from the global dataset;
- **differences in removal rates**: why some clients experience substantially higher removal ratios;
- **method consistency**: whether different filtering strategies tend to remove the same datapoints or produce distinct subsets.

Such analysis could inform the development of more interpretable and principled filtering methods.

6.7 Understanding mechanisms of learning gain

Future work could investigate the mechanisms underlying the observed improvements in accuracy. For example:

- Are there systematic cross-client influences during aggregation? Time-series methods such as Granger causality tests (Granger, 1969) could quantify directional dependencies, and additional statistical tests could assess significant relationships;
- Can clients' accuracy results be modelled using ARIMA (Box et al., 2013) or other stochastic models?
- Do information-theoretic (Shannon, 1948) measures (e.g. mutual information or entropy) help characterize inter-client relationships?

Such analyses would deepen the theoretical understanding of collaborative learning dynamics.

6.8 Extending use cases

FedFil was evaluated on tabular and text data. Future research should explore additional domains such as image classification, speech processing, time-series forecasting, to assess the generalizability of the methodology across different data modalities.

6.9 Testing new models and parametrizations

In this work, each client used a feedforward neural network. Future research could try different parametrizations within feedforward neural networks but also different models.

References

Arivazhagan, M. G., Aggarwal, V., Singh, A. K., and Choudhary, S. (2019). Federated learning with personalization layers.

Bevington, P. R. (1969). *Data Reduction and Error Analysis for the Physical Sciences*. McGraw-Hill.

Bhowmick, A., Duchi, J., Freudiger, J., Kapoor, G., and Rogers, R. (2018). Protection against reconstruction and its applications in private federated learning.

Blagec, K., Dorffner, G., Moradi, M., and Samwald, M. (2020). A critical analysis of metrics used for measuring progress in artificial intelligence.

Boutet, A., Castelluccia, C., Cunche, M., Roca, V., Baud, A., Raverdy, P.-G., and Lauradoux, C. (2021). Desire: Leveraging the best of centralized and decentralized contact tracing systems. *ACM Digital Threats: Research and Practice*. Special Issue on Security and Privacy for Covid-19.

Box, G. E. P., Jenkins, G. M., and Reinsel, G. C. (2013). *Time Series Analysis: Forecasting and Control*. John Wiley & Sons, 4 edition.

Cai, L., Hu, Q., and Wu, S. (2025). Federated learning of quantile inference under local differential privacy.

Cao, Y. and Yang, J. (2015). Towards making systems forget with machine unlearning. In *2015 IEEE Symposium on Security and Privacy*, pages 463–480. IEEE.

Castelluccia, C. and Le Métayer, D. (2019). Understanding algorithmic decision-making: Opportunities and challenges. European Parliament, STOA Study. Accessed: 10/11/2025.

Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys*, 41(3):15.

Chaum, D., Crépeau, C., and Damgård, I. (1987). Multiparty unconditionally secure protocols (abstract). In *Advances in Cryptology – CRYPTO '87: Proceedings of the Conference on the Theory and Applications of Cryptographic Techniques*, page 462, Santa Barbara, California, USA. Springer.

Chen, W.-N., Choquette-Choo, C. A., Suresh, A. T., and Kairouz, P. (2022). The fundamental price of secure aggregation in differentially private federated learning. *arXiv preprint*, arXiv:2203.03761. Preprint.

Cho, Y. J., Jhunjhunwala, D., Li, T., Smith, V., and Joshi, G. (2024). Maximizing global model appeal in federated learning. https://arxiv.org/abs/2205.14840.

Colosimo, B. M., Jones-Farmer, L. A., Megahed, F. M., Paynabar, K., Ranjan, C., and Woodall, W. H. (2024). Statistical process monitoring from industry 2.0 to industry 4.0: Insights into research and practice. *Technometrics*, 66(4):507–530.

Cramér, H. (1946). *Mathematical Methods of Statistics*. Princeton University Press, Princeton, NJ.

Dash, C. S. K., Behera, A. K., Dehuri, S., and Ghosh, A. (2023). An outliers detection and elimination framework in classification task of data mining. *Decision Analytics Journal*, 6:100164.

Ding, F., Hardt, M., Miller, J., and Schmidt, L. (2022). Retiring adult: New datasets for fair machine learning. *arXiv preprint arXiv:2108.04884*. Version 3, accessed on 2025-07-19.

Ding, W., Gelman, A., and Vehtari, A. (2021). Folktables: The american community survey dataset for fairness in ml. https://github.com/socialfoundations/folktables.

Dwork, C. and Roth, A. (2014). The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3–4):211–407.

Edgeworth, F. Y. (1887). On discordant observations. *Philosophical Magazine*, 23(5):364–375.

Fallah, A., Mokhtari, A., and Ozdaglar, A. (2020). Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 3557–3568. Curran Associates, Inc.

Gentry, C. (2021). A decade (or so) of fully homomorphic encryption. Slides presented at EUROCRYPT 2021.

Giles, D. E. (2022). Unbiased estimation of the standard deviation for non-normal populations. *WSEAS Transactions on Mathematics*, 21:119–121.

Go, A., Bhayani, R., and Huang, L. (2009). Twitter sentiment classification using distant supervision. https://www-cs.stanford.edu/people/alecmgo/papers/TwitterDistantSupervision09.pdf. CS224N Project Report, Stanford University.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

Gower, J. C. (1971). A general coefficient of similarity and some of its properties. *Biometrics*, 27(4):857–871.

Granger, C. W. J. (1969). Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37(3):424–438.

Grey, K. (2019). Xmr chart | step-by-step guide by hand and with r. https://r-bar.net/xmr-control-chart-tutorial-examples/. Accessed: 2025-09-12.

Grid'5000 Project Team (2025a). Getting started. Project Website. Accessed: 3/11/2025.

Grid'5000 Project Team (2025b). Grid'5000: Home. Project Website. Accessed: 3/11/2025.

Gundersen, G. (2022). Weighted ols: When should you use weighted least squares? Accessed: 2025-10-18.

Heyndrickx, W., Mervin, L., Morawietz, T., Sturm, N., Friedrich, L., Zalewski, A., Pentina, A., Humbeck, L., Oldenhof, M., Niwayama, R., Schmidtke, P., Fechner, N., Simm, J., Arany, A., Drizard, N., Jabal, R., Afanasyeva, A., Loeb, R., Verma, S., Harnqvist, S., Holmes, M., Pejo, B., Telenczuk, M., Holway, N., Dieckmann, A., Rieke, N., Zumsande, F., Clevert, D.-A., Krug, M., Luscombe, C., Green, D., Ertl, P., Antal, P., Marcus, D., Do Huu, N., Fuji, H., Pickett, S., Acs, G., Boniface, E., Beck, B., Sun, Y., Gohier, A., Rippmann, F., Engkvist, O., Göller, A. H., Moreau, Y., Galtier, M. N., Schuffenhauer, A., and Ceulemans, H. (2024). Melloddy: Cross-pharma federated learning at unprecedented scale unlocks benefits in qsar without compromising proprietary information. *Journal of Chemical Information and Modeling*, 64(7):2331–2344. PMID: 37642660.

High-Level Expert Group on Artificial Intelligence (2019). Ethics guidelines for trustworthy ai. https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai. European Commission.

Hsieh, K., Phanishayee, A., Mutlu, O., and Gibbons, P. B. (2020). The non-iid data quagmire of decentralized machine learning.

Huang, C., Huang, J., and Liu, X. (2022). Cross-silo federated learning: Challenges and opportunities. arXiv preprint arXiv:2206.12949. Accessed: 2025-11-03.

Inria (2025a). Federated machine learning over the internet (fedmalin). Inria Project Website. Accessed: 10/11/2025.

Inria (2025b). Inria startup studio. Website. Accessed: 10/11/2025.

Inria (2025c). Our history. Website. Accessed: 17/11/2025.

Inria Privatics Team (2025). Research. Website. Accessed: 17/11/2025.

Jin, W., Yao, Y., Han, S., Gu, J., Joe-Wong, C., Ravi, S., Avestimehr, S., and He, C. (2024). Fedml-he: An efficient homomorphic-encryption-based privacy-preserving federated learning system.

Jourdan, T., Boutet, A., and Frindel, C. (2021). Privacy assessment of federated learning using private personalized layers. In *MLSP 2021 - IEEE International Workshop on Machine Learning for Signal Processing*, pages 1–5, Queensland, Australia.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., D'Oliveira, R. G. L., Eichner, H., El Rouayheb, S., Evans, D., Gardner, J., Garrett, Z., Gascón, A., Ghazi, B., Gibbons, P. B., Gruteser, M., Harchaoui, Z., He, C., He, L., Huo, Z., Hutchinson, B., Hsu, J., Jaggi, M., Javidi, T., Joshi, G., Khodak, M., Konečný, J., Korolova, A., Koushanfar, F., Koyejo, S., Lepoint, T., Liu, Y., Mittal, P., Mohri, M., Nock, R., Özgür, A., Pagh, R., Qi, H., Ramage, D., Raskar, R., Raykova, M., Song, D., Song, W., Stich, S. U., Sun, Z., Suresh, A. T., Tramèr, F., Vepakomma, P., Wang, J., Xiong, L., Xu, Z., Yang, Q., Yu, F. X., Yu, H., and Zhao, S. (2021). Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 14(1–2):1–210.

Karimireddy, S. P., Guo, W., and Jordan, M. I. (2022). Mechanisms that incentivize data sharing in federated learning. *arXiv preprint arXiv:2207.04557*.

Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S. J., Stich, S. U., and Suresh, A. T. (2020). Scaffold: Stochastic controlled averaging for on-device federated learning. In *Proceedings of the 37th International Conference on Machine Learning*. PMLR.

Kasiviswanathan, S. P., Lee, H. K., Nissim, K., Raskhodnikova, S., and Smith, A. D. (2011). What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826.

Kelly, J., Zafar, S. A., Heidemann, L., Zacchi, J.-V., Espinoza, D., and Mata, N. (2024). Navigating the eu ai act: A methodological approach to compliance for safety-critical products. In *2024 IEEE Conference on Artificial Intelligence (CAI)*, pages 979–984.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*. arXiv preprint arXiv:1412.6980, first version 2014.

Kwak, B. I., Han, M. L., and Kim, H. K. (2021). Cosine similarity based anomaly detection methodology for the can bus. *Expert Systems with Applications*, 166:114066.

Law, A. M. and Kelton, W. D. (1991). *Simulation Modeling and Analysis*. McGraw-Hill, New York, 2nd edition.

LearnDataSci (n.d.). Cosine similarity (glossary). https://www.learndatasci.com/glossary/cosine-similarity/. Accessed: 2025-09-15.

Lehmann, R. (2013). 3σ-rule for outlier detection from the viewpoint of geodetic adjustment. *Journal of Surveying Engineering*, 139(4):157–165.

Li, A., Zhang, L., Tan, J., Qin, Y., Wang, J., and Li, X.-Y. (2021a). Sample-level data selection for federated learning. In *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, pages 1–10.

Li, Q., Diao, Y., Chen, Q., and He, B. (2021b). Federated learning on non-iid data silos: An experimental study.

Li, T., Sahu, A. K., Talwalkar, A., and Smith, V. (2020a). Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60.

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. (2020b). Federated optimization in heterogeneous networks. In Dhillon, I., Papailiopoulos, D., and Sze, V., editors, *Proceedings of Machine Learning and Systems*, volume 2, pages 429–450.

Liu, J., Huang, J., Zhou, Y., Li, X., Ji, S., Xiong, H., and Dou, D. (2021). From distributed machine learning to federated learning: A survey. *CoRR*, abs/2104.14362. arXiv:2104.14362.

McMahan, H. B., Moore, E., Ramage, D., Hampson, S., and Arcas, B. A. y. (2017). Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1273–1282.

McMahan, H. B., Ramage, D., Talwar, K., and Zhang, L. (2018). Learning differentially private recurrent language models. In *International Conference on Learning Representations (ICLR)*.

Melis, L., Song, C., De Cristofaro, E., and Shmatikov, V. (2019). Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 691–706. IEEE.

Mothukuri, V., Parizi, R. M., Pouriyeh, S., Huang, Y., Dehghantanha, A., and Srivastava, G. (2021). A survey on security and privacy of federated learning. *Future Generation Computer Systems*, 115:619–640.

Mugunthan, V., Byrd, D., Polychroniadou, A., and Balch, T. H. (2019). Smpai: Secure multi-party computation for federated learning. In *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS)*, Vancouver, Canada.

Naseri, M., Hayes, J., and Cristofaro, E. D. (2020). Local and central differential privacy for robustness and privacy in federated learning. In *Proceedings 2022 Network and Distributed System Security Symposium*. Accessed on 2025-10-18.

Nasr, M., Shokri, R., and Houmansadr, A. (2023). Training data extraction from diffusion models. *Proceedings of the IEEE Symposium on Security and Privacy*.

Nguyen, D. C., Ding, M., Pathirana, P. N., Seneviratne, A., Li, J., and Poor, H. V. (2021). Federated learning for internet of things: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 23(3):1622–1658.

Ouahi, M., Khoulji, S., and Kerkeb, M. L. (2024). Advancing sustainable learning environments: A literature review on data encoding techniques for student performance prediction using deep learning models in education. In *E3S Web of Conferences, STAR 2024*. E3S Web of Conferences. Accessed: 24/11/2025.

Ouyang, X., Xie, Z., Zhou, J., Huang, J., and Xing, G. (2021). Clusterfl: A similarity-aware federated learning system for human activity recognition. In *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '21)*, pages 54–66. Association for Computing Machinery.

Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.

Pati, S., Baid, U., Edwards, B., Sheller, M., Wang, S.-H., Reina, G. A., Foley, P., Gruzdev, A., Karkada, D., and Davatzikos, C. (2022). Federated learning enables big data for rare cancer boundary detection. *Nature Communications*, 13(1):7346.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Priya, B. (2020). Advances and open problems in federated learning. OpenMined Blog. Inspired by Peter Kairouz's talk at the OpenMined Privacy Conference 2020.

Pukelsheim, F. (1994). The three sigma rule. *The American Statistician*, 48(2):88–91.

Rabie, A. H., Saleh, A. I., Elkhalik, S. H. A., and Takieldeen, A. E. (2024). An optimum load forecasting strategy (olfs) for smart grids based on artificial intelligence. *Technologies*, 12:19.

Research and Markets (2025). Federated learning solutions market – global forecast 2025-2032. https://www.researchandmarkets.com/report/federated-learning#tag-pos-3.

RFI (2022). Government win as tousanticovid becomes 'most downloaded' app in france. RFI News. Accessed: 3/11/2025.

Rice, J. A. (2007). *Mathematical Statistics and Data Analysis*. Duxbury, an imprint of Thomson Brooks/Cole, Belmont, CA, USA, 3rd edition.

Rieke, N., Hancox, J., Li, W., Milletarì, F., Roth, H. R., Albarqouni, S., Bakas, S., Galtier, M. N., Landman, B. A., Maier-Hein, K., Ourselin, S., Sheller, M. J., Summers, R. M., Trask, A., Xu, D., Baust, M., and Cardoso, M. J. (2020). The future of digital health with federated learning. *npj Digital Medicine*, 3:119.

Robbins, H. and Monro, S. (1951). A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407.

Sagar, S., Li, C.-T., Loke, S. W., and Choi, J. (2022). Poisoning attacks and defenses in federated learning: A survey.

Segal, A., Marcedone, A., Kreuter, B., Ramage, D., McMahan, H. B., Seth, K., Bonawitz, K. A., Patel, S., and Ivanov, V. (2017). Practical secure aggregation for privacy-preserving machine learning. In *CCS*.

Senavirathne, N. and Torra, V. (2020). On the role of data anonymization in machine learning privacy. In *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 664–675.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423.

Shete, S., Mronga, D., Jadhav, A., and Kirchner, F. (2024). Online-adaptive anomaly detection for defect identification in aircraft assembly. In *2024 IEEE 20th International Conference on Automation Science and Engineering (CASE)*, pages 4126–4133, Bari, Italy. IEEE.

Shewhart, W. A. (1930). Economic quality control of manufactured product. *Bell System Technical Journal*, 9(2):364–389.

Shewhart, W. A. and Deming, W. E. (1986). *Statistical Method from the Viewpoint of Quality Control*. Dover Publications, New York.

Shokri, R., Stronati, M., Song, C., and Shmatikov, V. (2017). Membership inference attacks against machine learning models. *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18.

Sikder, M. N. K. and Batarseh, F. A. (2021). Outlier detection using ai: A survey.

Singh, P. (2025). 5 top papers of neurips 2024 that you must read. Analytics Vidhya Blog. Accessed: 17/11/2025.

Soudan, B., Abbas, S., Kubba, A., et al. (2025). Scalability and performance evaluation of federated learning frameworks: A comparative analysis. *International Journal of Machine Learning and Cybernetics*, 16:3329–3343.

Stanton, J. M. (2001). Galton, pearson, and the peas: A brief history of linear regression for statistics instructors. *Journal of Statistics Education*, 9(3).

Subasi, O., Bel, O., Manzano, J., and Barker, K. (2023). The landscape of modern machine learning: A review of machine, distributed and federated learning.

Sun, P., Chen, X., Liao, G., and Huang, J. (2022). A profit-maximizing model marketplace with differentially private federated learning. In *Proceedings of IEEE INFOCOM*, pages 1439–1448. IEEE.

Tamirisa, R., Xie, C., Bao, W., Zhou, A., Arel, R., and Shamsian, A. (2024). Fedselect: Personalized federated learning with customized selection of parameters for fine-tuning.

Torra, V. (2017). *Data Privacy: Foundations, New Developments and the Big Data Challenge*. Springer, Cham.

Tramèr, F., Zhang, F., Juels, A., Reiter, M. K., and Ristenpart, T. (2016). Stealing machine learning models via prediction apis. In *25th USENIX Security Symposium (USENIX Security 16)*, pages 601–618. USENIX Association.

Tukey, J. W. (1977). *Exploratory Data Analysis*. Addison-Wesley, Reading, Massachusetts.

Tuor, T., Wang, S., Ko, B. J., Liu, C., and Leung, K. K. (2020). Overcoming noisy and irrelevant data in federated learning.

University of Luxembourg (2025). Getting started on grid'5000. Presentation Slides. Accessed: 3/11/2025.

U.S. Census Bureau (2024). Pums data. https://www.census.gov/programs-surveys/acs/microdata/access.html. Accessed: 2025-08-14.

U.S. Census Bureau (2025). 2019-2023 acs pums data dictionary. https://www2.census.gov/programs-surveys/acs/tech_docs/pums/data_dict/PUMS_Data_Dictionary_2019-2023.pdf. Accessed: 2025-08-14.

van Gestel, R. and de Visser, M. (2021). The dutch benefits scandal: What have we learnt from it? https://eulawenforcement.com/?p=7941. EU Law Enforcement Blog.

Vanhaesebrouck, P., Bellet, A., and Tommasi, M. (2017). Decentralized collaborative learning of personalized models over networks. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 54 of *JMLR: Workshop and Conference Proceedings*, Fort Lauderdale, Florida, USA.

Vinutha, H. P., Poornima, B., and Sagar, B. M. (2018). Detection of outliers using interquartile range technique from intrusion dataset. In Satapathy, S., Tavares, J., Bhateja, V., and Mohanty, J., editors, *Information and Decision Sciences*, volume 701 of *Advances in Intelligent Systems and Computing*. Springer, Singapore.

Voigt, P. and von dem Bussche, A. (2017). *The EU General Data Protection Regulation (GDPR): A Practical Guide*. Springer International Publishing.

Wang, J., Liu, Q., Liang, H., Joshi, G., and Poor, H. V. (2020). Tackling the objective inconsistency problem in heterogeneous federated optimization. In *Advances in Neural Information Processing Systems*, volume 33.

Wei, K., Li, J., Ma, C., Ding, M., Wei, S., Wu, F., Chen, G., and Ranbaduge, T. (2024). Vertical federated learning: Challenges, methodologies and experiments.

Wen, J., Zhang, Z., Lan, Y., Cui, Z., Cai, J., and Zhang, W. (2024). A survey for federated learning evaluations: Goals and measures.

Whaley, D. L. (2005). The interquartile range: Theory and estimation. Master's thesis, East Tennessee State University.

Wheeler, D. and Chambers, D. (1992). *Understanding Statistical Process Control*. SPC Press.

Yang, Q., Liu, Y., Chen, T., and Tong, Y. (2019). Federated machine learning: Concept and applications. *CoRR*, abs/1902.04885.

Zhang, C., Xie, Y., Bai, H., Yu, B., Li, W., and Gao, Y. (2021). A survey on federated learning. *Knowledge-Based Systems*, 216:106775.

Zhang, X., Li, Q., Xue, J., Wang, H., Wang, S., Yang, B., Shen, W., Zhang, H., Liu, X., Gao, C., and Zhang, L. (2022). Federated learning for healthcare: A systematic review. *PLOS Computational Biology*, 18(9):e1010585. PMCID: PMC9650178.

Zhu, H., Xu, J., Liu, S., and Jin, Y. (2021). Federated learning on non-iid data: A survey.

Chapter 7

Appendix

7.1 US State abbreviations

- AL – Alabama
- AK – Alaska
- AZ – Arizona
- AR – Arkansas
- CA – California
- CO – Colorado
- CT – Connecticut
- DE – Delaware
- FL – Florida
- GA – Georgia
- HI – Hawaii
- ID – Idaho
- IL – Illinois
- IN – Indiana
- IA – Iowa
- KS – Kansas
- KY – Kentucky

- LA – Louisiana
- ME – Maine
- MD – Maryland
- MA – Massachusetts
- MI – Michigan
- MN – Minnesota
- MS – Mississippi
- MO – Missouri
- MT – Montana
- NE – Nebraska
- NV – Nevada
- NH – New Hampshire
- NJ – New Jersey
- NM – New Mexico
- NY – New York
- NC – North Carolina
- ND – North Dakota
- OH – Ohio
- OK – Oklahoma
- OR – Oregon
- PA – Pennsylvania
- RI – Rhode Island
- SC – South Carolina
- SD – South Dakota
- TN – Tennessee
- TX – Texas
- UT – Utah
- VT – Vermont
- VA – Virginia
- WA – Washington
- WV – West Virginia

- WI – Wisconsin
- WY – Wyoming

7.2 Sent140 Preliminary Assessment

Figure 60 illustrates the main issues encountered with the Sent140 dataset. The plot shows the accuracy results per client, cumulatively, both in terms of cross-distribution generalization (global test data) and local evaluation (local test data) (see 4.4) for a variety of learning procedures. In the case of the federated approaches, the accuracy in the plot pertains to the final round, while in classical learning it corresponds to the standard result, both presented in an ordered manner.

For instance, the grey curve representing FedPer ranges approximately from 0.5 to 0.7 means that:

- the lowest accuracy across clients is around 0.5, and the highest one is almost 0.7;
- up to 60% of the participants had an accuracy below 0.6, whereas 40% had an accuracy higher or equal to 0.6.

In classical learning (red curve), the model is trained on all data, and then the accuracy is computed separately for each state—treated analogously to a client in the FL experiments (see Section 4.4). The CD is built from these state-wise accuracies. For example, Figure 60 shows that under classical learning, 60% of states had an accuracy below 0.6.

The results on the local test data are not the main reason for discarding Sent140. Rather, it is the global generalization accuracy, represented by the dashed curves, that shows poor behavior. All five paradigms evaluated on the global test data exhibit constant or nearly constant accuracy values, meaning they did not generalize: results are approximately the same across all clients and data points for each learning type.

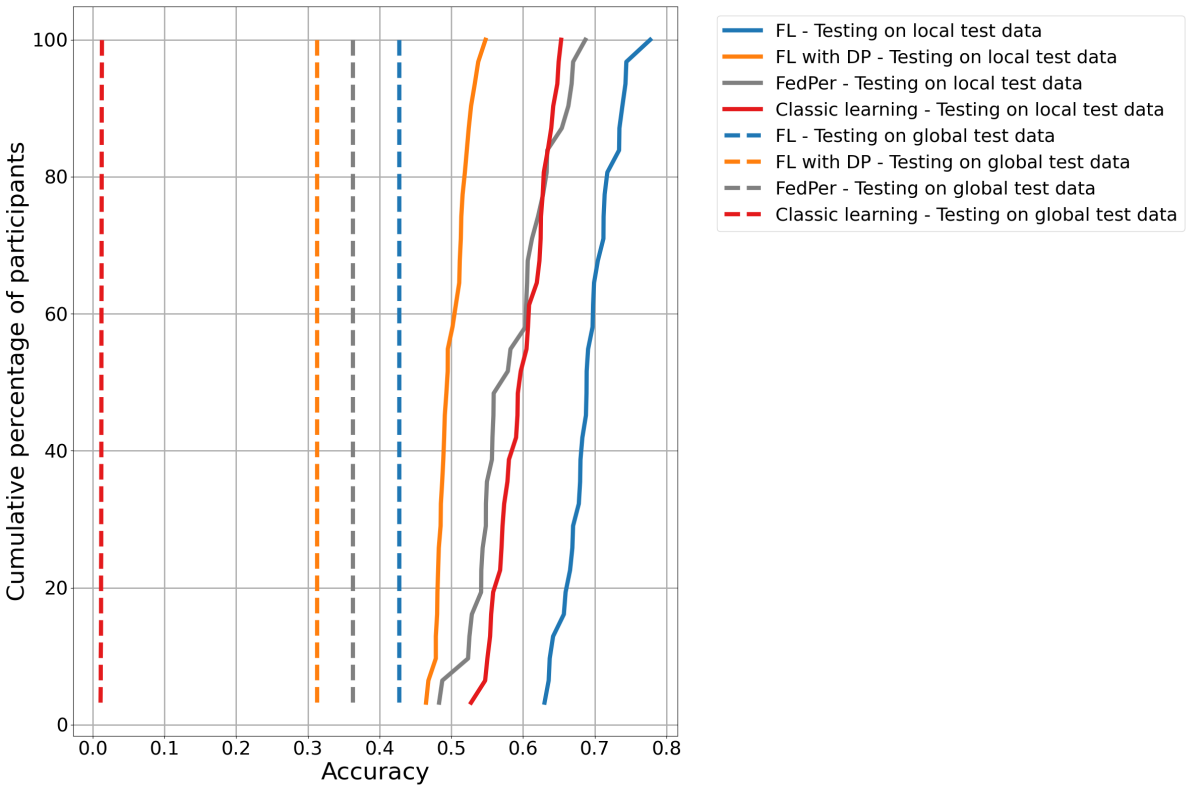


Figure 60: Sent140 dataset shows poor accuracy, motivating its exclusion from the main experiments.

7.3 Global data removal statistics per filtering strategy

Let $X_1, X_2, \ldots, X_k$ denote k disjoint client datasets, where each client i contributes $n_i = |X_i|$ data points. Let the total number of data points across all clients be

$$N = \sum_{i=1}^{k} n_i.$$

For a given filtering method j, let ϕ^{ij} denote the removal rate obtained for client i under method j. We define the client weights as

$$w_i = \frac{n_i}{N},$$

so that $\sum_{i=1}^{k} w_i = 1$.

The global removal rate for method j is then the weighted mean:

$$\phi^{j}_{\text{global}} = \sum_{i=1}^{k} w_i \, \phi^{ij}.$$

To quantify the variability of removals across clients, the (unbiased) weighted variance (Bevington, 1969) is computed:

$$s^2_{w,j} = \left(\sum_{i=1}^{k} w_i \left(\phi^{ij} - \phi^{j}_{\text{global}} \right)^2 \right) \cdot \left(\frac{k}{k-1} \right),$$

and the corresponding weighted standard deviation

$$s_{w,j} = \sqrt{s^2_{w,j}}.$$

Table 5 reports, for each filtering strategy, the global removal rate ϕ^{j}_{global} and its weighted standard deviation $s_{w,j}$. The results indicate that the IQR-based method produces the highest overall proportion of data removed, and also the highest cross-client heterogeneity, as reflected by its large weighted standard deviation. In contrast, cosine similarity (CS) with a 1% threshold exhibits both the lowest global removal rate and the smallest weighted dispersion across clients.

Table 5: Global data removal per filtering strategy

Strategy	Global removal (%)	Weighted standard deviation (%)
IQR	6.449	2.346
3SR	1.927	0.499
CS (Thr 2.5%)	2.500	1.230
CS (Thr 1%)	1.000	0.514

7.4 Global mean and standard deviation from sample statistics

The proofs in this section follow the idea presented in (Rice, 2007).

Let $X_1, X_2, \ldots, X_k$ be k disjoint samples with $n_i = |X_i|$, the number of elements in sample i.

Define the global sample as:

$$X = \bigcup_{i=1}^{k} X_i, \qquad N = \sum_{i=1}^{k} n_i.$$

7.4.1 Univariate case: global mean

The global mean $\bar{X}$ is defined as the arithmetic mean of all elements in X:

$$\bar{X} = \frac{1}{N} \sum_{x \in X} x.$$

Since X is composed of disjoint samples $X_1, X_2, \ldots, X_k$, we can rewrite the sum over X as a sum over each sample:

$$\bar{X} = \frac{1}{N} \sum_{i=1}^{k} \sum_{x \in X_i} x.$$

Let $\bar{X}_i$ denote the mean of the i-th sample, i.e.,

$$\bar{X}_i = \frac{1}{n_i} \sum_{x \in X_i} x. \tag{7.1}$$

Substituting into the equation 7.1 the equality:

$$\sum_{x \in X_i} x = n_i \bar{X}_i$$

we obtain the global mean as a weighted average of the sample means:

$$\bar{X} = \frac{1}{N} \sum_{i=1}^{k} n_i \bar{X}_i.$$

This formula allows us to compute the global mean using only the sample means $\bar{x}_i$ and the sample sizes n_i.

7.4.2 Univariate case: global standard deviation

By definition, the (unbiased) global variance is:

$$S^2(X) = \frac{1}{N-1} \sum_{i=1}^{k} \sum_{x \in X_i} (x - \bar{X})^2.$$

We can decompose $x - \bar{X}$ as:

$$x - \bar{X} = (x - \bar{X}_i) + (\bar{X}_i - \bar{X})$$

where $\bar{X}_i$ is the sample mean of client i.

Expanding the square:

$$(x - \bar{X})^2 = (x - \bar{X}_i)^2 + 2(x - \bar{X}_i)(\bar{X}_i - \bar{X}) + (\bar{X}_i - \bar{X})^2.$$

Summing over $x \in X_i$:

$$\sum_{x \in X_i} (x - \bar{X})^2 = \sum_{x \in X_i} (x - \bar{X}_i)^2 + 2(\bar{X}_i - \bar{X}) \sum_{x \in X_i} (x - \bar{X}_i) + \sum_{x \in X_i} (\bar{X}_i - \bar{X})^2.$$

Observation:

$$\sum_{x \in X_i} (x - \bar{X}_i) = \sum_{x \in X_i} x - n_i \bar{X}_i = n_i \bar{X}_i - n_i \bar{X}_i = 0.$$

This is because the sum of deviations from the mean within a sample is always zero by definition of the sample mean. Thus, the middle term disappears, leaving:

$$\sum_{x \in X_i} (x - \bar{X})^2 = \sum_{x \in X_i} (x - \bar{X}_i)^2 + n_i (\bar{X}_i - \bar{X})^2.$$

Dividing by $N - 1$, we get the global variance:

$$S^2(X) = \frac{1}{N-1} \sum_{i=1}^{k} (n_i - 1) S^2(X_i) + \frac{1}{N-1} \sum_{i=1}^{k} n_i (\bar{X}_i - \bar{X})^2.$$

where:

$$S^2(X_i) = \frac{1}{n_i - 1} \sum_{x \in X_i} (x - \bar{X}_i)^2$$

is the variance of X_i.

Finally, the global standard deviation is the square root of the global variance:

$$S(X) = \sqrt{S^2(X)} = \sqrt{\frac{1}{N-1} \sum_{i=1}^{k} \Big[(n_i - 1) S^2(X_i) + n_i (\bar{X}_i - \bar{X})^2 \Big]}.$$

This formula allows the exact computation of the global standard deviation using only the sample means $\bar{X}_i$, the sample variances $S^2(X_i)$, and the sample sizes n_i, without needing to concatenate all data.

7.4.3 Multidimensional case: global mean vector

Each participant i has a **local dataset** represented as a matrix:

$$X_i \in \mathbb{R}^{n_i \times m},$$

where n_i is the number of observations and m is the number of features. Each row corresponds to an observation, and each column to a feature:

$$X_i = \begin{bmatrix} x_{i,1,1} & x_{i,1,2} & \dots & x_{i,1,m} \\ x_{i,2,1} & x_{i,2,2} & \dots & x_{i,2,m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{i,n_i,1} & x_{i,n_i,2} & \dots & x_{i,n_i,m} \end{bmatrix}.$$

Equivalently, each row can be seen as a vector in $\mathbb{R}^m$:

$$X_i = \begin{bmatrix} x_{i,1}^\top \\ x_{i,2}^\top \\ \vdots \\ x_{i,n_i}^\top \end{bmatrix}, \qquad x_{i,j} \in \mathbb{R}^m,$$

where $x_{i,j}$ is the j-th observation of participant i.

The **local mean vector** of participant i is:

$$\bar{x}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} x_{i,j} \in \mathbb{R}^m,$$

containing the mean of each feature for that participant.

By extending the univariate derivation of the global mean (cf. Sec. 7.4.1) to each feature separately, we can write the global mean vector $\bar{X} \in \mathbb{R}^m$ as a weighted average of the local mean vectors $\bar{x}_i \in \mathbb{R}^m$:

$$\bar{X} = \frac{1}{N} \sum_{i=1}^{k} n_i \, \bar{x}_i.$$

More explicitly, for each feature $f = 1, \dots, m$, the f-th component of the global mean vector is

$$\bar{X}_f = \frac{1}{N} \sum_{i=1}^{k} n_i \, \bar{x}_{i,f},$$

where $\bar{x}_{i,f}$ is the mean of feature f in the local dataset of participant i.

This formula allows us to compute the global mean using only the sample means $\bar{x}_i$ and the sample sizes n_i.

7.5 Global quantile estimation from sample quantiles

Let $X_1, X_2, \ldots, X_k$ be k disjoint samples with:

- $n_i = |X_i|$, the number of elements in sample i;
- $Q_{p,i} = \inf\{x : \hat{F}_{n_i}(x) \geq p\}$, the p-th sample quantile of X_i, where where $\hat{F}_{n_i}(x) = \frac{1}{n_i} \sum_{j=1}^{n_i} \mathbf{1}_{\{X_{i,j} \leq x\}}$ is the empirical cumulative distributive function (ECDF) of X_i.

where $X_{i,j}$ is the j-th element of sample X_i, and $\mathbf{1}_A$ is the indicator function, defined as:

$$\mathbf{1}_{\{A\}} = \begin{cases} 1, & \text{if condition } A \text{ is true,} \\ 0, & \text{otherwise.} \end{cases}$$

Define the global, aggregated sample as:

$$X = \bigcup_{i=1}^{k} X_i, \qquad N = \sum_{i=1}^{k} n_i,$$

The p-th global sample quantile $\chi_p(X)$ is defined as:

$$\chi_p(X) = \inf\{x : \hat{F}_N(x) \geq p\},$$

where $\hat{F}_N(x)$ is the ECDF of the aggregated sample X:

$$\hat{F}_N(x) = \frac{1}{N} \sum_{i=1}^{k} \sum_{j=1}^{n_i} \mathbf{1}_{\{X_{i,j} \leq x\}}$$

7.5.1 Weighted average estimator scheme

An estimate $\hat{Q}_p^{\text{weighted}}$ can be obtained by weighting the sample quantiles according to the sample sizes:

$$\hat{Q}_p^{\text{weighted}} = \sum_{i=1}^{k} w_i Q_{p,i}, \qquad w_i = \frac{n_i}{N}.$$

Derivation of the weighted average as weighted least squares estimator

Define the weighted loss function:

$$\ell(q) = \sum_{i=1}^{k} w_i \left(Q_{p,i} - q\right)^2$$

where $w_i = \frac{n_i}{N}$ are the weights proportional to sample sizes.

Step 1: Derivative Take the derivative of $\ell(q)$ with respect to q:

$$\frac{d\ell}{dq} = \sum_{i=1}^{k} w_i \frac{d}{dq} \left(Q_{p,i} - q\right)^2 = \sum_{i=1}^{k} w_i \cdot 2(q - Q_{p,i}) = 2\sum_{i=1}^{k} w_i (q - Q_{p,i})$$

Step 2: Set derivative equal to zero To find the minimizer, set the derivative to zero:

$$2\sum_{i=1}^{k} w_i (q - Q_{p,i}) = 0$$

This is to say

$$\sum_{i=1}^{k} w_i (q - Q_{p,i}) = 0$$

Step 3: Solve for q Expand the sum:

$$\begin{aligned}
&\sum_{i=1}^{k} w_i q - \sum_{i=1}^{k} w_i Q_{p,i} = 0 \\
\iff\ & q\sum_{i=1}^{k} w_i - \sum_{i=1}^{k} w_i Q_{p,i} = 0 \\
\iff\ & q\sum_{i=1}^{k} w_i = \sum_{i=1}^{k} w_i Q_{p,i}
\end{aligned}$$

Step 4: Substitute $w_i = \frac{n_i}{N}$

$$q = \frac{\sum_{i=1}^{k} \frac{n_i}{N} Q_{p,i}}{\sum_{i=1}^{k} \frac{n_i}{N}} = \frac{\sum_{i=1}^{k} n_i Q_{p,i}}{N}$$

Step 5: Final estimator Thus, the weighted average of sample quantiles is

$$\hat{Q}_p^{\text{weighted}} = q = \sum_{i=1}^{k} \frac{n_i}{N} Q_{p,i} = \frac{\sum_{i=1}^{k} n_i Q_{p,i}}{N}.$$

This shows that the weighted average is exactly the minimizer of the **weighted least squares problem** (Gundersen, 2022), justifying its use as a stable estimator of the global quantile.

7.5.2 Iterative quantile estimation via stochastic approximation

Definition of the Stochastic Iterative Estimator

We define an iterative procedure to aggregate the sample quantiles $Q_{p,i}$ into an estimate of the global quantile $\chi_p(X)$ using only the local quantiles and their sample sizes.

Let

$$w_i = \frac{n_i}{N}, \qquad \sum_{i=1}^{k} w_i = 1$$

be the relative weights of the samples.

Iterative update scheme

Initialize

$$Q_p^{(0)} = \min_i Q_{p,i}.$$

At iteration t:

1. Select an index $I_t \in \{1, \ldots, k\}$ randomly with probability $\mathbb{P}(I_t = i) = w_i$.

2. Update the current estimate toward the selected sample's quantile:

$$Q_p^{(t+1)} = (1 - \eta_{I_t})Q_p^{(t)} + \eta_{I_t} Q_{p,I_t}, \qquad \eta_{I_t} = w_{I_t}.$$

Equivalently, this can be written as

$$Q_p^{(t+1)} = Q_p^{(t)} + \eta_{I_t}\big(Q_{p,I_t} - Q_p^{(t)}\big).$$

Expectation analysis

Conditional on the current value $Q_p^{(t)}$, the next iteration satisfies

$$\mathbb{E}[Q_p^{(t+1)} \mid Q_p^{(t)}] = \sum_{i=1}^{k} w_i \Big((1 - \eta_i)Q_p^{(t)} + \eta_i Q_{p,i}\Big) = Q_p^{(t)} \sum_{i=1}^{k} w_i(1 - \eta_i) + \sum_{i=1}^{k} w_i \eta_i Q_{p,i}.$$

Since $\eta_i = w_i$, define

$$S = \sum_{i=1}^{k} w_i^2, \qquad \tilde{Q}_p = \sum_{i=1}^{k} w_i^2 Q_{p,i}.$$

Then the conditional expectation simplifies to

$$\mathbb{E}[Q_p^{(t+1)} \mid Q_p^{(t)}] = (1 - S)Q_p^{(t)} + \tilde{Q}_p.$$

Iteration of the expectation

Knowing the conditional expectation $\mathbb{E}[Q_p^{(t+1)} \mid Q_p^{(t)}]$, we proceed to find the expected value $\mathbb{E}[Q_p^{(t+1)}]$.

Law of Total Expectation To transition from the conditional expectation $\mathbb{E}[Q_p^{(t+1)} \mid Q_p^{(t)}]$ to $\mathbb{E}[Q_p^{(t+1)}]$, we apply the Law of Total Expectation (Cramér, 1946):

$$\mathbb{E}[X] = \mathbb{E}[\mathbb{E}[X \mid Y]].$$

Recognizing $X = Q_p^{(t+1)}$ and $Y = Q_p^{(t)}$, we take the expectation $\mathbb{E}[\cdot]$ of the conditional result:

$$\mathbb{E}[\mathbb{E}[Q_p^{(t+1)} \mid Q_p^{(t)}]] = \mathbb{E}[(1 - S)Q_p^{(t)} + \tilde{Q}_p].$$

Left-Hand Side (LHS): by the Law of Total Expectation, the LHS simplifies to:

$$\mathbb{E}[\mathbb{E}[Q_p^{(t+1)} \mid Q_p^{(t)}]] = \mathbb{E}[Q_p^{(t+1)}] = m_{t+1}.$$

Right-Hand Side (RHS): since S and $\tilde{Q}_p$ are constants:

$$\mathbb{E}[(1-S)Q_p^{(t)} + \tilde{Q}_p] = (1-S)\mathbb{E}[Q_p^{(t)}] + \mathbb{E}[\tilde{Q}_p].$$

Defining $m_t = \mathbb{E}[Q_p^{(t)}]$, we obtain:

$$\mathbb{E}[(1-S)Q_p^{(t)} + \tilde{Q}_p] = (1-S)m_t + \tilde{Q}_p.$$

Equating both sides (LHS and RHS), we arrive at the following recurrence for the expected value:

$$m_{t+1} = (1-S)m_t + \tilde{Q}_p.$$

Proof by mathematical induction To illustrate why this simplifies, we use mathematical induction to prove the following closed-form result:

$$m_t = (1-S)^t m_0 + \tilde{Q}_p \sum_{k=0}^{t-1} (1-S)^k$$

Base Case ($t = 1$): From the recursive definition:

$$m_1 = (1-S)m_0 + \tilde{Q}_p.$$

From the proposed closed-form:

$$m_1 = (1-S)^1 m_0 + \tilde{Q}_p \sum_{k=0}^{1-1} (1-S)^k = (1-S)m_0 + \tilde{Q}_p (1-S)^0 = (1-S)m_0 + \tilde{Q}_p.$$

Since both definitions match, the base case $t = 1$ holds true.

Inductive Hypothesis: Assume the formula holds true for an arbitrary step t:

$$m_t = (1-S)^t m_0 + \tilde{Q}_p \sum_{k=0}^{t-1} (1-S)^k$$

Inductive Step: We show that the formula holds for $t+1$ by substituting the Inductive Hypothesis into the recursive definition $m_{t+1} = (1-S)m_t + \tilde{Q}_p$:

$$\begin{aligned}
& m_{t+1} = (1-S)m_t + \tilde{Q}_p \\
\implies & m_{t+1} = (1-S)\left[(1-S)^t m_0 + \tilde{Q}_p \sum_{k=0}^{t-1}(1-S)^k\right] + \tilde{Q}_p \\
\implies & m_{t+1} = (1-S)^{t+1} m_0 + \tilde{Q}_p (1-S) \sum_{k=0}^{t-1}(1-S)^k + \tilde{Q}_p \\
\implies & m_{t+1} = (1-S)^{t+1} m_0 + \tilde{Q}_p \left[\sum_{k=0}^{t-1}(1-S)^{k+1} + 1\right]
\end{aligned}$$

The summation term can be rewritten by shifting the index and adding the missing term for $k=0$:

$$\begin{aligned}
\sum_{k=0}^{t-1}(1-S)^{k+1} + 1 &= \sum_{j=1}^{t}(1-S)^j + (1-S)^0 \\
&= \sum_{j=0}^{t}(1-S)^j
\end{aligned}$$

Substituting this back into the expression for m_{t+1}:

$$m_{t+1} = (1-S)^{t+1} m_0 + \tilde{Q}_p \sum_{k=0}^{t}(1-S)^k$$

The formula for m_{t+1} holds. Hence, by mathematical induction, the closed-form expression for m_t holds for all $t \geq 0$.

Convergence Using the formula for the finite geometric series: $\sum_{k=0}^{n} r^k = \frac{1-r^{n+1}}{1-r}$, with $r = (1-S)$, the general expression for the expected value is:

$$m_t = (1-S)^t m_0 + \tilde{Q}_p \left[\frac{1-(1-S)^t}{S} \right]$$

Since $0 < S \leq 1$, we have $|1-S| < 1$, which implies that $(1-S)^t \to 0$ as $t \to \infty$. Therefore, the expectation of the estimator converges to

$$m_t = \frac{\tilde{Q}_p}{S}$$

$$\iff m_t = \frac{\sum_{i=1}^{k} w_i^2 Q_{p,i}}{\sum_{i=1}^{k} w_i^2}$$

Connection to stochastic approximation

Conceptually, this procedure is similar to stochastic approximation algorithms, such as Robbins–Monro (Robbins and Monro, 1951), but with three key differences:

- The update is based on a randomly selected component (sample quantile) rather than the full function;
- The step size η_{I_t} is determined by the selected sample and does not decay over iterations;
- Convergence is guaranteed in expectation, not almost surely.

Advantage over weighted estimator (Section 7.5.1)

The weighted sampling strategy allows samples with larger n_i (hence larger w_i) to have more influence in early iterations, accelerating convergence toward the global quantile. This may be beneficial in highly heterogeneous datasets, where some samples contain far more observations than others.

7.6 Results of global quantile estimation

In order to assess the quality of the estimation, two error measures are defined:

$$\text{Relative error} = \frac{\text{Estimated value} - \text{True value}}{\text{True value}} \times 100,$$

$$\text{Absolute error} = \text{Estimated value} - \text{True value}.$$

The absolute error is considered because there are instances where the true value is zero, making the relative error undefined.

7.6.1 IQR method

Table 6 presents the estimation results used in the IQR method, segmented by the two proposed approaches, comparing them to the true results and their associated errors. It can be seen that the stochastic iterative proposal (Section 7.5.2) has a smaller associated error on the first quartile estimate, while the weighted average estimation (Section 7.5.1) has a lower error on the third quartile estimate. A caveat is the fact that while weighted average estimation is deterministic, the stochastic iterative one is not, which naturally introduces variability across runs.

Table 6: Global quantile estimation for IQR method

Statistic	Estimation scheme	iterations	Estimated value	True value	Error	Error type
First quartile	Weighted average	–	0.0875	0	0.0875	Absolute
First quartile	Stochastic iterative	50	0	0	0	Absolute
Third quartile	Weighted average	–	43663.4917	43000	1.543%	Relative
Third quartile	Stochastic iterative	50	45114.42	43000	4.621%	Relative

Note: – means not applicable.

7.6.2 CS method

It can be seen that the stochastic iterative proposal (Section 7.5.2) has a smaller associated error on the 2.5% percentile estimation compared to the weighted average scheme (Section 7.5.1), while the weighted average scheme has a lower associated error on the 1% percentile estimation. A caveat is the fact that while weighted average estimation is deterministic, the stochastic iterative one is not, which naturally introduces variability across runs.

Table 7: Global quantile estimation for CS method

Statistic	Estimation scheme	iterations	Estimated Value	True Value	Error	Error type
Percentile 1%	Weighted average	–	0.3113	0.0809	284.796%	Relative
Percentile 1%	Stochastic iterative	50	0.7701	0.0809	851.916%	Relative
Percentile 2.5%	Weighted average	–	0.9089	0.9544	-4.763%	Relative
Percentile 2.5%	Stochastic iterative	50	0.9894	0.9544	3.667%	Relative

Note: – means not applicable.

7.7 Experiment deployment on Grid'5000

As described in Figure 61, when using Grid'5000 a user typically:

1. connects via SSH, a standard network protocol that establishes a secure communication channel between two machines, to an access machine;
2. connects from this access machine to a site frontend;
3. reserves resources (nodes) from the site frontend and then connects to those nodes.

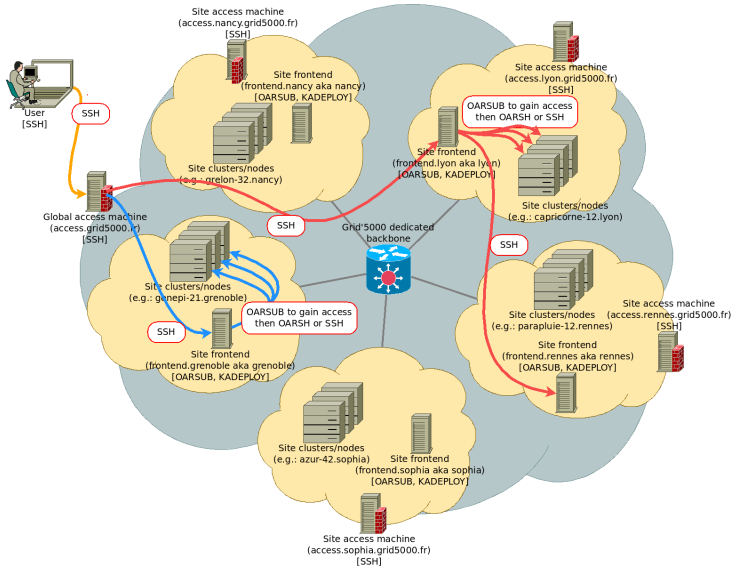


Figure 61: Grid'5000 SSH access. Source: Grid'5000 Project Team (2025a).

Figure 62 shows the global access machine used, running Ubuntu through WSL (Windows Subsystem for Linux). As part of Inria Privatics, it was possible to run experiments in the Abaca (production) queue, a long-running jobs queue with higher priority compared to standard users. Abaca sites include Grenoble, Nancy, Rennes, and Sophia.

This section focuses on Sophia, as it was the main experimentation site, but the workflow is analogous across all other sites.

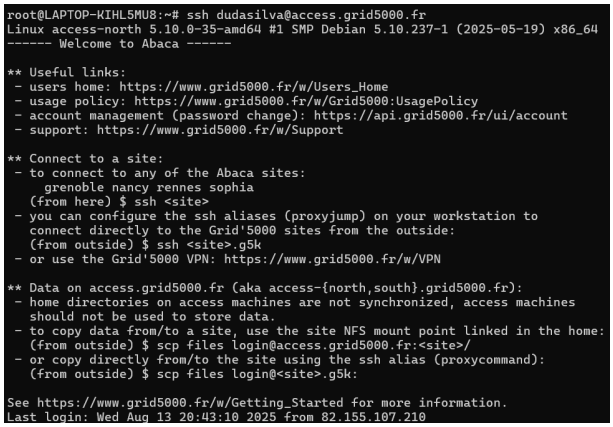


Figure 62: Global access machine.

Figure 63 shows an example of some of the available resources on the Sophia access machine.

Figure 64 illustrates a submitted job. Each job generates an output file with a `.stdout` extension, using either the automatically generated job ID or an user-defined name, and an error file with a `.stderr` extension. The job description also indicates the queue type, in this case production, with the assigned priority level (`p3`, where `p1` corresponds to higher priority). It also includes the start time, requested walltime, end time, and job status; in this example, the job is marked as *Terminated*, meaning it completed successfully without errors. The *initial_request* field displays the full job creation request, including the number of requested GPUs.

Given the extensive usage of Grid'5000 and the heterogeneous hardware across nodes, it is often useful to check the current node availability of a site. Grid'5000 provides several tools for resource monitoring and management (University of Luxembourg, 2025), one of which is a live view of resource usage. Figure 65 shows an example for the Sophia site.

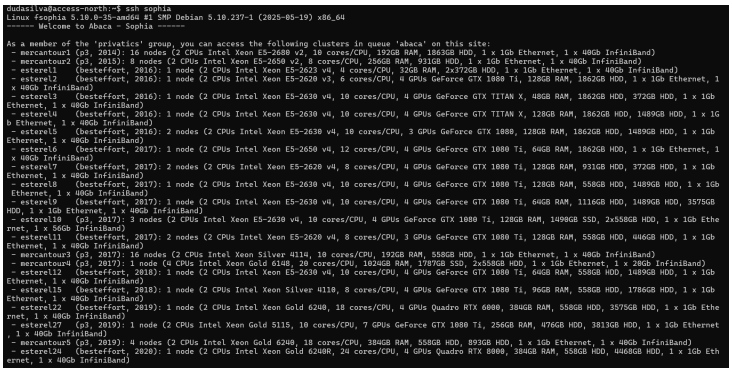


Figure 63: Sophia access machine.

Figure 64: Detailed job status.

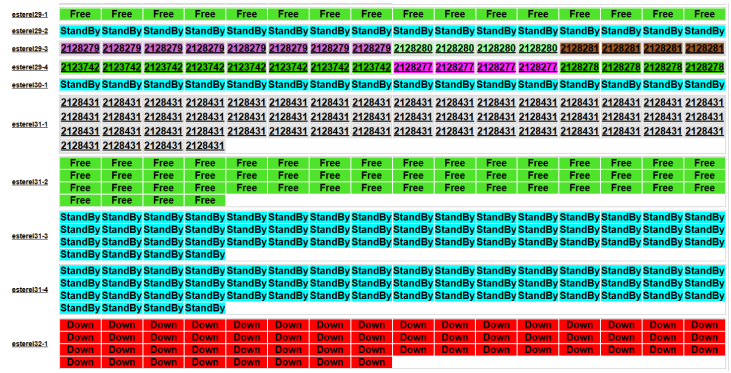


Figure 65: Partial node status on the Sophia site.