SAPIENZA
UNIVERSITÀ DI ROMA

A Thorough Assessment of the Non-IID Data Impact in Federated Learning

Department of Computer, Control and Management Engineering (DIAG)
Master's degree (M.Sc.) in Data Science

Mehrdad Hassanzadeh
ID number 1961575

Advisor
Prof. Ioannis Chatzigiannakis

Co-Advisor
Daniel M. Jimenez-Gutierrez

Academic Year 2024-2025

Thesis defended on 18th of July 2025
in front of a Board of Examiners composed by:

Prof. BRUTTI PIERPAOLO (chairman)
Prof. CRESPI MATTIA
Prof. D'ECCLESIA RITA LAURA
Prof. LEOTTA FRANCESCO
Prof. POLITO POMPEO
Prof. QUATTROCIOCCHI WALTER
Prof. SCARDAPANE SIMONE

A Thorough Assessment of the Non-IID Data Impact in Federated Learning
Master thesis. Sapienza University of Rome

This thesis has been typeset by LaTeX and the Sapthesis class.

Author's email: hassanzadeh.1961575@studenti.uniroma1.it

Acknowledgments

This thesis is dedicated to Professor Ioannis Chatzigiannakis, who entrusted me with this project and provided constant intellectual, professional, and personal support throughout. His belief in my abilities, significant resource allocation, and consistent guidance have had a significant impact on both this work and my professional development as a researcher.

A special dedication goes to Dr. Daniel M. Jimenez-Gutierrez my co-advisor. More than an advisor, colleague or a friend, his constant availability, sophisticated guidance, and genuine companionship helped me overcome every challenges.

I would also want to thank Professor Andrea Vitaletti and Professor Aris Anagnostopoulos for their insightful feedback and collaborative efforts, as well as our research group for providing an environment conducive to rigorous research.

To my wife, whose self-less efforts, invincible belief, and enduring love have served as the foundation for this achievement.

To my parents and my family, for their lifelong encouragement, persistent trust in my abilities, and the values they established.

And to Farhad and Farzad, my uncles, whose continuous support and unwavering encouragement have propelled me throughout my academic journey.

Your collective contributions have made this work possible. Thank you.

Abstract

Federated Learning (FL) allows several decentralized clients to train machine learning models together without sharing their local data, ensuring data privacy. However, when client data is not independent and identically distributed (non-IID), model performance degrades significantly compared to IID or centralized settings. Previous research has often relied on ad hoc setups or has only partially addressed the full spectrum of non-IID data distributions. As a result, it remains an open question how much the non-IID data variability affects performance and where the critical thresholds lie. Furthermore, non-IID data in FL can be divided into four types: label skew, feature skew, quantity skew, and spatiotemporal skew. Currently, most existing research has focused on only one or a subset of these types. In this work, we present the first complete and systematic assessment of all four categories of heterogeneity, quantifying non-IID data and evaluating its impact on model performance using the Hellinger Distance (HD), a widely established metric in the literature. We compare five cutting-edge FL aggregation algorithms across eight thoroughly studied image and tabular datasets. Our results show that, among the four types of skew, only label skew and spatiotemporal skew significantly impact model performance. Under label skew, model performance is noticeably more affected in a double-thresholding manner at two distinct HD thresholds, happening at 0.5 and at 0.75. This work establishes a solid empirical foundation for creating more resilient and fair FL solutions by fully exploring the whole range of non-IID circumstances, including the first in-depth exploration of spatiotemporal skew.

Contents

Chapter 1

Introduction

In the era of widespread digitalization, the integration of machine learning (ML) with sensitive domains such as healthcare and finance has transformative potential, for example, in improving diagnostic accuracy [62] and detecting financial fraud [58]. However, these advances come with huge concerns regarding data privacy, particularly when working with confidential records from hospitals or financial institutions. To address this tension, Trusted Research Environments (TREs) have emerged as a critical framework for enabling secure ML research while upholding stringent privacy protections [21, 86].

Federated Learning (FL) [49] has emerged as a transformative paradigm for collaboratively training ML models across decentralized data silos while preserving data privacy and ensuring regulatory compliance. This approach is particularly advantageous in cross-silo scenarios, such as collaborations between hospitals, financial institutions, and other organizations where direct data sharing is impractical or prohibited.

Despite its promise, FL faces a core challenge: the non-IID data distributions across participating clients, commonly referred to as non-IID (non-independent and identically distributed) data. This non-IID data poses significant obstacles to model generalization and convergence efficiency during training [72, 50]. The existing literature classifies non-IID data scenarios into four main categories: label skew, feature skew, quantity skew, and spatiotemporal skew [89], each known to contribute differently to model performance degradation.

Spatiotemporal skew presents a distinct and critical challenge that remains underexplored in FL research. This form of non-IID data arises when data distributions vary among geographical regions (spatial) and/or over time (temporal)[14]. Unlike label, feature, or quantity skew, spatiotemporal skew introduces dynamic, evolving patterns that conventional FL aggregation strategies often fail to effectively handle [3]. Addressing this type of skew is essential, as it directly influences a model's ability to generalize across diverse real-world environments while preserving temporal relevance. As such, spatiotemporal skew constitutes a key frontier in developing robust, adaptive FL systems.

Detecting and quantifying non-IID data in FL remains a significant open challenge, as highlighted by recent surveys and papers [60, 40], identifying it as a key research direction to advance the field. To address this, several studies have proposed metrics

to capture the degree of non-IID data in FL systems [31, 18, 55, 73]. Among these, the Hellinger Distance (HD) [20] has emerged as one of the most robust and informative measures. Unlike the Jensen-Shannon Distance (JSD), which often saturates at moderate divergence levels, HD provides a fine-grained characterization of distributional differences and can approach a value of 1 under conditions of extreme non-IID data. Furthermore, HD is an adaptable and applicable metric across multiple types of data skew, making it a valuable tool for empirical analysis in FL research.

Motivation. Recent advancements in FL have significantly enhanced our knowledge of non-IID data concerns, with significant success in resolving isolated kinds of non-IID data, including label skew [71, 74, 51]. Building on this achievement, there is a vital opportunity to synthesize current findings using comprehensive empirical benchmarks covering the whole range of non-IID cases. While foundational theoretical frameworks [45, 60] propose essential mitigation strategies, applying these principles requires systematic quantification of how different types of non-IID data, from feature skew to spatiotemporal skew, impact model performance in practical FL settings. Filling this gap through careful empirical research will allow the creation of FL systems that are theoretically sound and empirically robust across a wide range of application domains.

Our study fills this gap by using HD to measure client distributional variances. This allows for a rigorous empirical investigation of the impact of non-IID data across four critical dimensions. Additionally, our spatiotemporal skew assessment captures the influence of dynamic data shifts over both time and space, which is especially relevant to real-world domains such as credit risk assessment in banking [87] and personalized healthcare [22]. Taking this thorough and principled approach, we have discovered robust and generalizable insights that expand our knowledge of non-IID data in FL.

Contribution. The key contributions of our study are summarized as follows:

1. We benchmark five widely used state-of-the-art FL algorithms, including aggregation and/or client selection strategies, evaluating their performance in handling non-IID data under controlled, realistic, and quantifiable synthetic partitioning. Our evaluation covers all significant types of non-IID data partitioning, such as label, feature, quantity, and spatiotemporal skews. This work includes the first empirical analysis of how spatiotemporal skew impacts FL model performance.

2. We propose using HD as a standardized and fine-grained metric to measure distributional differences among clients in FL. This allows for more systematic and reproducible analyses of non-IID data. While we focus on HD in this study as a well-accepted metric by the literature, we recognize exploring alternative metrics as an essential direction for future work (see Section 12).

3. We provide researchers with practical references by determining which FL approaches are robust against particular types of non-IID data by measuring

them on popular and widely used datasets, guiding the selection of relevant strategies for real-world deployment.

4. We provide FL researchers with practical insights and recommendations based on the systematic characterization of non-IID data across various non-IID data scenarios.

To the best of our knowledge, this work presents the most comprehensive empirical study on the effects of non-IID data in FL, covering a wide range of non-IID data types and offering in-depth performance analyses across multiple benchmark settings.

Chapter 2

Related Work

In this chapter, we examine recent empirical studies that investigate the effects of non-IID data in FL. Furthermore, we position our work in relation to existing surveys and reviews on non-IID data in FL, highlighting how our contributions extend and differ from the prior literature.

2.1 Empirical Studies

Studies that systematically analyze and benchmark the performance of methods for addressing the effects of non-IID data on FL models under controlled conditions are limited. However, in this section, we review existing works that studied this to some extent, offering empirical insights into the impact of non-IID data in FL.

A study by Vahidian et al. [71] challenges traditional assumptions about non-IID data in FL. They argue that different data distributions across clients are not fundamentally harmful, and in some cases offer advantages, which is a finding that aligns with some of our observations. Their analysis is grounded in two key points: (i) label skew is not the only contributor to heterogeneity, and (ii) the angle between data subspaces across clients serves as a more informative metric for capturing heterogeneity. *Complementing their work, our study extends the scope by addressing a broader range of non-IID data types including label, feature, quantity, and spatiotemporal skews considering evaluating the performance across both image and tabular datasets.*

Wong et al. [74] conduct an extensive empirical study using a large network of IoT and edge devices to characterize the real-world aspects of FL, including model performance, as well as computational and communication costs. Their analysis is focused on heterogeneous settings, acknowledging non-IID data as one of the major challenges in FL. However, their study focuses primarily on image datasets and does not include comparative evaluations of aggregation methods under highly heterogeneous conditions. *In contrast, our work extends this line of research by incorporating both image and tabular datasets, and by benchmarking state-of-the-art FL strategies across all major non-IID data types, thereby providing a more comprehensive and practically relevant perspective on handling non-IID data in FL.*

Mora et al. [51] review existing strategies in the literature aimed at addressing the challenges introduced by non-IID data in FL. They provide both a conceptual

analysis of these methods, highlighting their underlying assumptions and limitations, and an empirical comparison to identify the most promising approaches. However, their evaluation is restricted to label skew and relies on a single dataset. *In contrast, our work adopts a broader perspective by analyzing multiple types of non-IID data including label, feature, quantity, and spatiotemporal skews across different types of datasets. We also identify practical limitations and offer guidance on strategies to address them effectively.*

Li et al. [37] conducted a comprehensive experimental evaluation of FL aggregation algorithms in non-IID data scenarios. Their study systematically examines the strengths and weaknesses of several state-of-the-art aggregation strategies, employing diverse data partitioning techniques to simulate various non-IID data scenarios. The authors highlight key challenges associated with non-IID data, such as model accuracy degradation and training instability, while offering valuable empirical insights across multiple settings. *Building upon this foundation, our work extends the analysis to include spatiotemporal skew a critical yet understudied form of non-IID data, and introduces quantitative metrics to assess its impact. This broader perspective contributes to a deeper understanding of the effects of non-IID data on FL model performance in real-world scenarios.*

2.2 Surveys

Several surveys have investigated the impact of non-IID data on model performance, as outlined in Table 2.1. In particular, studies from 2024 emphasize label skew, offering critical insights into this phenomenon. A Research from 2022 briefly addresses spatiotemporal skew, broadening our appreciation of these influences. The 2021 work establishes a foundational analysis of label skew, thereby enabling more expansive examinations in the following years.

Resource	Publication Year	Label Skew	Feature Skew	Quantity Skew	Spatiotemporal Skew	non-IID data Quantification	Empirical Highlights
[45]	2024	✓	✗	✗	✗	✗	✗
[60]	2024	✓	✓	✓	✗	⊖	✗
[46]	2022	✓	✓	✓	✗	✗	✗
[14]	2022	⊖	✗	✗	✓	✗	✗
[89]	2021	✓	✓	✓	⊖	✗	✗
Ours	2025	✓	✓	✓	✓	✓	✓

Table 2.1. Comparison against surveys (resources) for non-IID data in FL (✓: Included, ⊖: Partially included, ✗: Not included)

In contrast to the surveys listed in Table 2.1, our study not only builds on their findings, but also expands them through a more thorough methodology. *We perform empirical assessments of non-IID data effects, quantify the degree of non-IID data, and carry out extensive experiments to rigorously evaluate spatiotemporal skew.* These improvements extend the analysis and increase the practical application of our work in comparison to previous research.

Chapter 3

Background

In this chapter, we present an overview of FL, including its core training methodology and the issues that arise from non-IID data. We classify various types of data skew that influence FL effectiveness and describe approaches for quantifying the degree of non-IID data. Furthermore, we discuss state-of-the-art aggregation and client selection methods that have been proposed to overcome these issues, such as FedAvg [49], FedProx [40], Random size-proportional selection (Rand)[12], Power-Of-Choice (POC)[12], and Model Contrastive Learning (MOON) [39].

3.1 Key Terms and Definitions

Before describing the mechanics and algorithms of FL in the following of this chapter, we collect the key terms and notation that will be used throughout this paper. Defining these concepts upfront provides clarity when discussing client-server interactions, performance metrics, and other non-IID data contexts.

Federated Learning (FL): A distributed learning paradigm in which different clients (e.g., mobile devices, hospitals, banks) train a common global model under the supervision of a central server without exchanging raw data. Each client performs local updates on its own private dataset, sharing only model parameters or gradients.

Client (Participant): In the FL literature, the total number of clients is generally indicated by K. An FL client is any entity that owns private data and has computational and communication capabilities. During training, clients make local model updates on their own datasets and return just the resulting parameters or gradients to the central server. Clients may range from resource-constrained IoT devices to high-performance servers run by huge enterprises.

Server (Coordinator): The server serves as the FL's coordinator, initializing the global model before regularly selecting client subsets, distributing the current global model over the clients, and collecting model updates from the clients. It aggregates these model updates, usually using a size-weighted average, to create a new global model, which it then redistributed over the clients. The server also handles client scheduling, dropouts, secure aggregation, and global model validation (if possible).

Aggregation algorithm: It is the server's approach to aggregate the model updates received from clients into a single global model. In its most basic form,

Federated Averaging (FedAvg) [49], each client's weight update is scaled by the size of its local dataset and then averaged across all participating clients. Advanced aggregation approaches, such as Federated Proximal (FedProx) [40], Trimmed Mean [80], and Krum [8], enhance the robustness of the fundamental scheme against client heterogeneity, stragglers, or malicious updates through regularization, outlier filtering, or trust-based weighting.

Communication Round: Let T denote the total number of communication rounds in FL. In each communication round, the server first selects which clients will participate in the training of this round, then sends the current global model to them. These selected clients train the model locally using their own data and simply provide the model updates to the server. The server aggregates these updates, which are often weighted by the size of each client's data to create an updated global model that is ready for the next round. In practice, FL is set up to run for a pre-defined number of communication rounds.

Local Epoch: Let E represent the number of local epochs that each client's model trains for in each communication round. A local epoch is a single pass of a client's full private dataset during on-client model training. Clients may run many epochs in each round to fine-tune their model before sending updates back to the server.

Batch size: Let B denote the batch size used by each client when computing local gradient updates. It is a fixed-size subset of a client's data used for gradient descent steps. Processing data in batches speeds up training by allowing multiple weight updates per local epoch, decreases memory needs, and introduces controlled stochasticity.

3.2 Basics of FL

FL[49] is particularly suitable for scenarios involving siloed data, also known as clients, local nodes, parties, or participants, where multiple institutions independently manage their datasets. This decentralized strategy allows collaborative training of models without requiring direct access to raw data, thus preserving data privacy and ensuring data ownership remains with the individual participating organizations[17].

Figure 3.1 gives an overview of FL. In this process, each client (M_i) using their own datasets ($\mathcal{D}_i$), will train the globally distributed model (M) [65, 85] received from the server. Rather than sharing raw data, clients securely exchange their model updates (M_i), which contain no sensitive information, with a central aggregation server. This server aggregates updates to improve the global model. Across multiple rounds of model exchanges, clients collaboratively improve the global model while maintaining data privacy, since only aggregated updates are communicated rather than raw data to the server.

3.3 Different Classes of FL

Different classification schemes can be used for FL, each emphasizing a different aspect of how FL systems are organized or deployed. Typically, the FL system can be classified based on the following two aspects [68]:

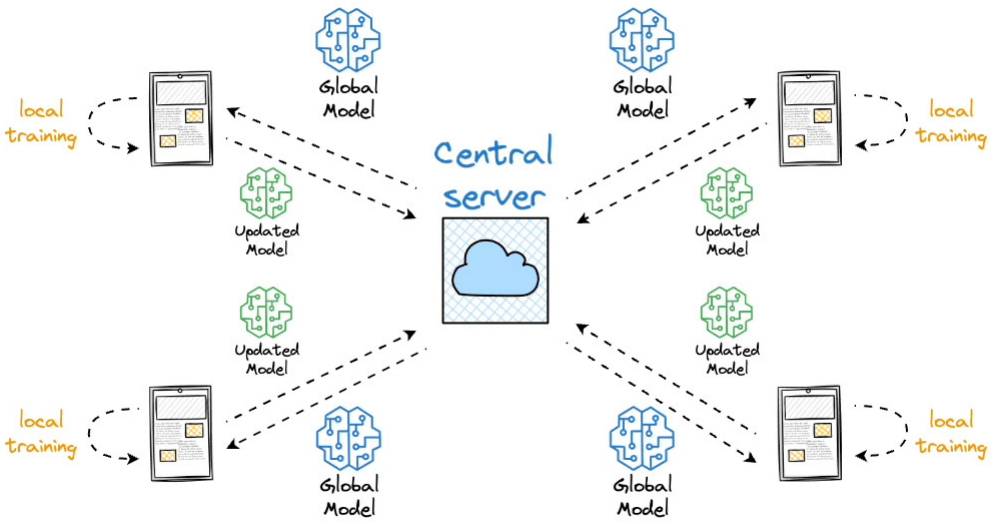

Figure 3.1. FL training process overview [15]

1. **Participating clients** based on the nature, size, and capabilities of the participating clients. In the literature, we can have the following two cases:
   - Cross-device FL
   - Cross-silo FL
2. **Data partition** based on how the data is partitioned over the client:
   - Horizontal Federated Learning
   - Vertical Federated Learning
   - Federated Transfer Learning

In the following part, we will learn more about each of these two classifications and their respective types.

3.3.1 Participating clients

FL can be categorized according to the nature, scale, and reliability of the clients who participate in the training process. The differences between these two types of categories can be seen in Figure 3.2

Cross-device clients are typically individual edge devices such as smartphones, wearables, or IoT sensors. This setup often involves thousands to millions of participants, each with a small dataset and having limited compute, storage, and network capabilities [33]. Due to differences in the client device (e.g. computational power, connectivity) and restricted communication resources on the server, not all clients can be included in every training round [75]. To avoid network congestion and ensure timely aggregation, the server employs an aggregation or sampling approach to select a fraction of clients each round while balancing communication overhead, latency, accuracy, and resource limitations.

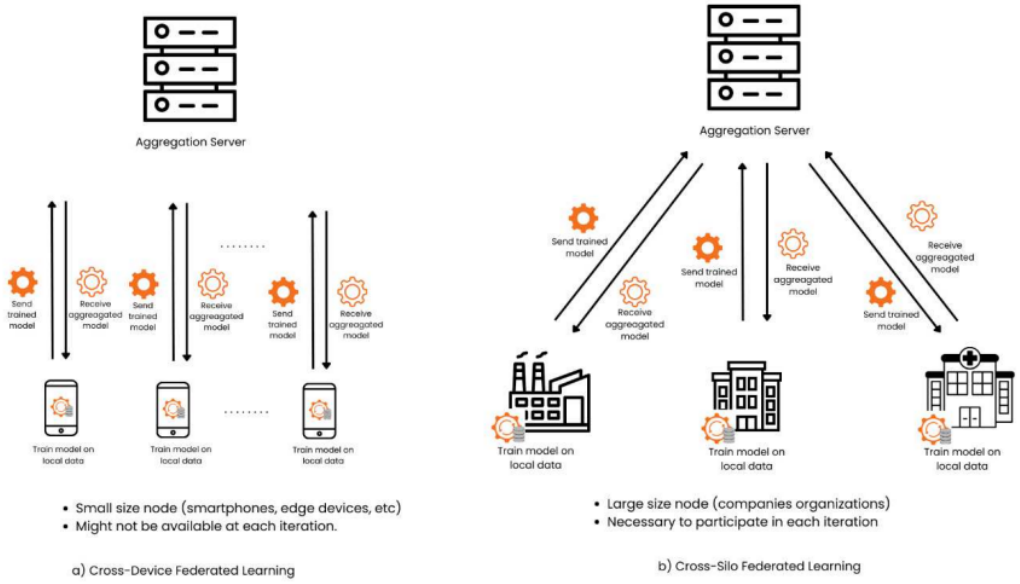

Figure 3.2. Participating clients types [68]

Cross-silo involves trusted clients such as hospitals, banks, and research institutions [27, 29, 52]. The number of participants in such a setting is often small, ranging from a few to more than a few tens. In contrast to cross-device FL, cross-silo clients are expected to actively contribute during the training process, employing their considerable computational capacity, large structured datasets, and reliable connectivity throughout the training process.

3.3.2 Data partition

FL is classified into three types according to how samples and features are distributed among clients, as seen in Figure 3.3. Because the model and its training data are stored in separate clients, decentralization allows for a variety of training approaches [84, 68].

Horizontal Federated Learning (HFL) allows many parties who share the same feature space but have distinct sample sets [78, 28] to train a ML model together without exchanging raw data. HFL increases overall sample size and improves model performance by horizontally splitting datasets along the user dimension, that is, each client keeps its own records whose feature schemas agree but whose individual users rarely overlap. For example, two healthcare providers in separate locations can integrate their patient data (which uses identical tests and measurements but covers different patient populations) to train a single prediction model, while protecting the privacy of each client's sensitive records [84].

Vertical Federated Learning (VFL) allows companies with largely similar user bases but distinct sets of attributes [44, 78] to jointly train a model without disclosing their raw data. In VFL, datasets are vertically partitioned, with each partner keeping all records for a common set of users, but just a subset of the feature columns. By aligning on user identities, contributors can enrich the model's input

while maintaining privacy. For example, a bank and an e-commerce platform in the same region may have customers who share financial and credit information, while the e-commerce site has a browsing and purchasing history [84]. VFL allows them to use these complementary qualities to create a more effective predictive model without transferring sensitive data.

Federated Transfer Learning (FTL) allows companies with datasets that have little overlap in terms of users and features to collaborate to improve a shared model by borrowing ideas from their respective domains [64, 10, 43]. Rather than partitioning data by samples or features, FTL uses transfer learning approaches to fill gaps caused by limited labeled examples or insufficient data in each party's local set. Similarly, a hospital radiology department with few annotated radiographs can use a pre-trained image recognition network to start its diagnostic model [84]. FTL tackles data scarcity and heterogeneity by reusing knowledge from separate but complementary databases.

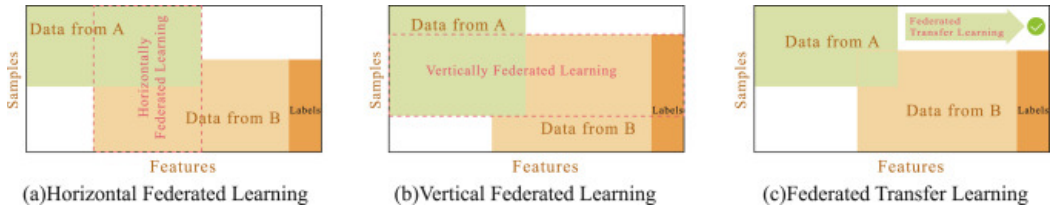

Figure 3.3. Different FL classifications based on data partitioning [84]

3.4 Data skew types.

A centralized dataset $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ is defined as a collection of n tuples. Here, each tuple comprises a feature vector $\mathbf{x}_i = [(x_i)_1, \ldots, (x_i)_m]$, representing the characteristics of the ith sample, and a corresponding label $y_i \in 1, \ldots, \ell$ indicating the true class of that sample.

In a FL scenario, the dataset $\mathcal{D}$ is partitioned among K different clients. Let $\mathcal{D}_i$ denote the subset of data associated with the client i. Thus, the entire dataset is represented as:

$$\mathcal{D} = \cup_{i=1}^{K} \mathcal{D}_i \qquad \text{and for } i \neq j: \qquad \mathcal{D}_i \cap \mathcal{D}_j = \emptyset.$$

Clearly characterizing the type of non-IID data in FL is essential, as it significantly impacts model performance. In this work, we adopt definitions consistent with previous literature [47, 89]. Specifically, for supervised learning tasks at client i (or local node i), we consider each data instance $(x, y) \in \mathcal{D}_i$, where x represents input attributes or features and y denotes the corresponding label, to be drawn from a local distribution $P_i(\mathbf{x}, y)$. We formally define:

$$P_i^Y(y) = \sum_{\substack{(\mathbf{x},z)\in\mathcal{D}_i \\ z=y}} P_i(\mathbf{x}, z) \quad \text{and} \quad P_i^{X_\ell}(x) = \sum_{\substack{(\mathbf{x},y)\in\mathcal{D}_i \\ x_\ell=x}} P_i(\mathbf{x}, y) \tag{3.1}$$

where $P_i^Y(y)$ denotes the label distribution for the ith client, and $P_i^{X_\ell}(x)$ represents the distribution of the ℓth input feature for the same client. Based on these definitions,

the classification of non-IID data, specifically the various types of data skew, is described as follows:

- Regarding the concept of *identically distributed*:
  1. **Label skew**: Indicates that the label distributions $P_i^Y(y)$ vary across different clients.
  2. **Feature skew**: Occurs when the distribution of the features $P_i^{X_\ell}(x)$ differs between clients.
  3. **Quantity skew**: Denotes considerable variation in the number of data samples across different clients. $P_i(\mathbf{x}, y)$.
- Regarding the concept of *independent*:
  4. **Spatiotemporal skew**: Commonly referred to as spatiotemporal skew in the context of federated continual learning (FCL) [81, 79, 82], this phenomenon describes the intrinsic correlation of data across temporal or spatial dimensions. Specifically, the distribution $P_i(\mathbf{x}, y)$ is non-stationary and varying as a function of time or spatial location.

3.5 Quantifying the Degree of Non-IID data

In the context of selecting representative scenarios to illustrate the impact of non-IID data in FL, existing studies frequently employ ad-hoc partitioning strategies [42, 38, 26]. To address this limitation, our approach incorporates *systematic metric to quantify the degree of non-IID data, enabling the selection of scenarios that effectively capture its effects*. Specifically, we use HD, a well-established measure for assessing the divergence between two probability distributions, defined in Equation 3.2 [20].

$$\mathsf{HD}(P_1^Y(y), P_2^Y(y)) = \frac{1}{\sqrt{2}}\sqrt{\sum_{y\in Y}\left(\sqrt{P_1^Y(y)} - \sqrt{P_2^Y(y)}\right)^2} \tag{3.2}$$

The HD offers a precise and sensitive measure of distributional divergence, approaching values near 1 in cases of severe non-IID data, unlike the Jensen-Shannon Divergence (JSD), which tends to saturate and inadequately capture high skew levels. Moreover, HD demonstrates strong adaptability to various forms of non-IID data. Compared to the Earth Mover's Distance (EMD) [31], which is heavily influenced by the choice and scale of the underlying ground distance, HD provides normalized and consistent comparisons across different tasks and datasets, making it especially suitable for FL contexts.

3.6 Aggregation and Client Selection Algorithms

In FL, the server collects model weights from each client, aggregates them, and then distributes the updated weights back to the participants. This section details the five cutting-edge aggregation and client selection algorithms evaluated in our experiments.

FederatedAveraging (FedAvg) [49]: In this strategy, each client computes model updates based on its own data and sends them to a central server. The server averages the incoming updates to produce a global model update, as illustrated in Algorithm 1, and communicates it to the clients. This cycle is repeated iteratively until the model converges, which is defined as achieving steady performance or exhausting the predefined communication rounds.

Algorithm 1 FedAvg. Here, K denotes the number of clients, B the local batch size, T number of the communication rounds, E the number of local epochs per client, and η the learning rate. [49]

Server executes:
initialize w_0
for each round $t = 1, 2, \ldots, T$ **do**
  for each client $k \in K$ **in parallel do**
    $w_{t+1}^k \leftarrow \text{ClientUpdate}(k, w_t)$
  end for
  $w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k$
end for

ClientUpdate(k, w):
$\mathcal{B} \leftarrow$ (split P_k into batches of size B)
$w_k \leftarrow w$
for each local epoch e from 1 to E **do**
  for each batch $b \in \mathcal{B}$ **do**
    $w_k \leftarrow w_k - \eta \nabla \ell(w_k; b)$
  end for
end for
return w to server

Federated Proximal (FedProx) [40]: FedProx improves FedAvg's to tackle non-IID data challenges in FL by adding a proximal regularization term across clients (see Algorithm 2). Clients want to minimize both the empirical loss and a penalty of $\frac{\mu}{2}\|w - w_t\|^2$ to prevent local models w from deviating much from the current global model w_t. Once each client has completed its task, the server collects these updates using a size-weighted average to create the next global model. This combination of proximal regularization enhances convergence stability and overall performance of non-IID data scenarios in FL.

Algorithm 2 FedProx. Here, K is the number of clients, B the local batch size, T number of the communication rounds, E the number of local epochs, η the learning rate, and μ the proximal regularization coefficient.

Server executes:
initialize w_0
for each round $t = 1, 2, \ldots, T$ **do**
  for each client $k \in K$ **in parallel do**
    $w_{t+1}^k \leftarrow \text{FedProxClientUpdate}(k, w_t)$
  end for
  $w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k$
end for

FedProxClientUpdate(k, w):
$\mathcal{B} \leftarrow$ (split P_k into batches of size B)
$w_k \leftarrow w$
for each local epoch e from 1 to E **do**
  for each batch $b \in \mathcal{B}$ **do**
    $w_k \leftarrow w_k - \eta(\nabla \ell(w_k; b) + \mu\,(w_k - w))$
  end for
end for
return w_k to server

Rand [11]: Rand provides a basic but informative baseline for client selection using non-IID data. Rand selects a predetermined fraction C of the total K clients only based on data volume. Specifically, during each communication round t, the server calculates

$$m \;=\; \max(\lceil C \cdot K \rceil, 1) \quad \text{and} \quad p_k \;=\; \frac{n_k}{\sum_{j=1}^{K} n_j}$$

where n_k denotes the number of local samples on client k and p_k denotes the probability that client k be selected. It then draws m clients without replacement considering the clients' probability selection $\{p_k\}$. This proportionate, randomness-based technique assures that larger clients participate more frequently simply because of their data size with the idea that these client would have more information to share. After selecting the clients, FedAvg's standard method would be performed to collect model updates from the clients and aggregate them on the server.

Power of Choice (POC) [11]: POC expands random client selection by adding two parameters: the sample fraction C and the candidate pool size d. As before, let $m = \max(\lceil C \cdot K \rceil, 1)$ represent the number of customers to be selected each round. At round t, the server selects d candidate clients (without replacement) where $d \in [m, K]$ using the data-proportion distribution $p_k = n_k / \sum_{j=1}^{K} n_j$. It then broadcasts the current global model to these selected d candidates, who compute and report their local loss values on the current global model without performing any training at this stage. Finally, the server selects the m clients with the highest losses from the candidates to participate in this round. This technique balances exploration and exploitation through randomized sampling and loss-based prioritizing, resulting in

improved performance in non-IID data distributions. This strategy aims to optimize workload distribution and prioritize clients offering more informative updates, thereby enhancing the efficiency of the FL process.

Algorithm 3 POC. Here, K is the number of clients, C the fraction of clients to be selected per each round, d candidate pool size, B the local batch size, T number of the communication rounds, E the number of local epochs, η the learning rate, and μ the proximal regularization coefficient. [11]

Server executes:
initialize w_0
$m \leftarrow \max(\lceil C \cdot K \rceil, 1)$
compute $p_k = \frac{n_k}{\sum_{j=1}^{K} n_j}$ for $k = 1, \ldots, K$
for each round $t = 1, 2, \ldots, T$ **do**
  $D_t \leftarrow$ selected d candidate clients *without replacement according to* p_k
  for each client $k \in D_t$ **in parallel do**
    $\ell_k \leftarrow \text{ClientLocalLoss}(k, w)$
  end for
  $S_t \leftarrow$ the m clients in D_t with largest ℓ_k
  for each client $k \in S_t$ **in parallel do**
    $w_{t+1}^k \leftarrow \text{ClientUpdate}(k, w_t)$
  end for
  $w_{t+1} \leftarrow \sum_{k \in S_t} \frac{n_k}{\sum_{j \in S_t} n_j} w_{t+1}^k$
end for

ClientLocalLoss(k, w):
$\ell_k \leftarrow$ local loss of w on P_k
return ℓ_k to server

ClientUpdate(k, w):
$\mathcal{B} \leftarrow$ (split P_k into batches of size B)
$w_k \leftarrow w$
for each local epoch $e = 1$ to E **do**
  for each batch $b \in \mathcal{B}$ **do**
    $w_k \leftarrow w_k - \eta \nabla \ell(w_k; b)$
  end for
end for
return w to server

MOON [39]: MOON is a straightforward yet effective FL framework that addresses non-IID data by introducing model-level contrastive learning solely at the client side, while leaving the standard server aggregation unchanged. As shown in Figure 3.4, each client's model is composed of a base encoder, a projection head, and an output layer. The base encoder extracts representation vectors from the input data, while the projection head maps these representations into a fixed-dimensional space. The output layer then generates predictions for each class. For clarity, the entire network with weights w is denoted $F_w(\cdot)$, and the network up to (but

excluding) the output layer is $R_w(\cdot)$, so that $R_w(X_\ell)$ is the mapped representation of input X_ℓ.

During each local update, MOON adds a model-level contrastive loss that encourages the current client model to stay close to the global model from the previous round and to move away from its own prior local snapshot. Concretely, given the global model parameters w^g and the client's previous local parameters w^c, and an input X_ℓ, the local loss is

$$\mathcal{L}_{\text{MOON}} = \ell_{\text{CE}}(F_w(X_\ell), y_\ell) \; + \; \mu \, \mathcal{L}_{\text{contrast}}(R_w(X_\ell), \, R_{w^g}(X_\ell), \, R_{w^c}(X_\ell)),$$

where ℓ_{CE} denotes the supervised classification loss (e.g., the standard cross-entropy loss), μ weights the contrastive term and

$$\mathcal{L}_{\text{contrast}}(z, z^+, z^-) = -\log \frac{\exp(\cos(z, z^+)/\tau)}{\exp(\cos(z, z^+)/\tau) + \exp(\cos(z, z^-)/\tau)}$$

where $z = R_w(X_\ell)$ is the current client's representation, $z^+ = R_{w^g}(X_\ell)$ is the most recent global model's representation, $z^- = R_{w^c}(X_\ell)$ client's previous local model's representation and uses cosine similarity $\cos(\cdot, \cdot)$ with temperature τ. This ensures that, during local training, each client's representation is pulled toward the global consensus and pushed away from its own stale model, mitigating client drift under heterogeneous data.

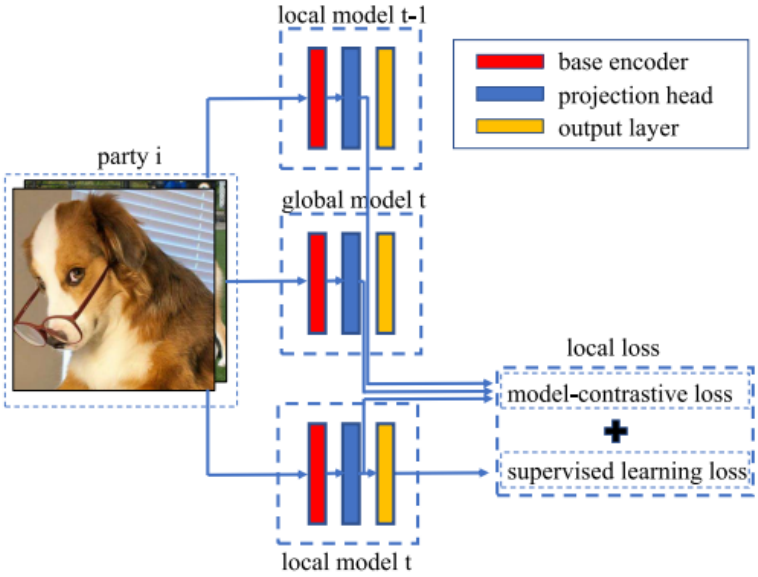

Figure 3.4. The local loss in Moon [39]

Chapter 4

Experimentation Setup

This chapter describes the configuration and environment utilized to carry out our studies. We start by presenting the datasets, architectures to create models, training parameters and the hyperparameters considered in each configuration. We also define the hardware and software frameworks utilized to carry out and manage our experiments. Finally, we establish the evaluation metrics considered that helped us to perform our analysis.

4.1 Datasets

This study utilizes eight widely recognized real-world datasets to train both CL and FL models. Among these, four datasets CIFAR10 [35], FMNIST [77], Physionet 2020 [23], and Covtype [7], are employed to simulate label, feature, and quantity skew. Additionally, to investigate label skew in contexts involving a significantly larger number of classes, the CIFAR100 dataset [34] is also incorporated.

The other three datasets—5G Network Traffic flows [13], MHEALTH [4], and Snapshot Serengeti [69]—are utilized to model spatiotemporal skew. An overview of the key characteristics of each dataset is presented in Table 4.1.

Table 4.1. Characteristics of the datasets

Dataset	Type	#training examples	#test examples	#features	#classes	Classes distribution
CIFAR10	Images	50,000	10,000	3,072	10	Balanced
FMNIST	Images	60,000	10,000	784	10	Balanced
CIFAR100	Images	50,000	10,000	3,072	100	Balanced
Physionet	Tabular	39,895	2,095	120	27	Balanced
Covtype	Tabular	522,910	58,102	54	7	Unbalanced
Serengeti	Tabular	257,927	28,659	64	13	Unbalanced
5G NTF	Tabular	74,838	13,207	7	12	Unbalanced
MHEALTH	Tabular	851,021	364,724	14	13	Unbalanced

4.2 Models

Table 4.2 summarizes the models chosen for each dataset under the two training strategies. For the CIFAR-10 and FMNIST datasets, we adopt a standard convolutional neural network (CNN) architecture widely used in computer vision tasks [9]. Figure 4.1 shows the layout of this model. The network begins with an input layer, followed by three convolutional blocks. The first two blocks each comprise a convolutional layer (with ReLU activation) followed by a 2×2 max-pooling layer; the third block consists of a convolutional layer (ReLU) followed by a flattening operation. The initial convolution uses 32 filters, while the subsequent two employ 64 filters apiece, all with 3×3 kernels. After flattening, a fully connected layer of 64 neurons with ReLU activation produces the final feature representation. For the CIFAR100 dataset, we adopt a ResNet9-based model [25], termed ResNet9+, with architectural details outlined in Table 4.3. Moreover, our experiments include transfer learning models EfficientNetB0 [70] and MobileNetV2 [66], selected for their superior classification performance on the datasets under study.

Table 4.2. Models applied to each dataset using different training approaches.

Training	Dataset	Model(s)
Full Training	CIFAR10	CNN [4.1]
	FMNIST	
	CIFAR100	ResNet+ [4.3]
	Physionet	DNN [4.2]
	Covtype	
	Serengeti	
	5G NTG	
	MHEALTH	
Transfer Learning	CIFAR10	EfficientNetB0 [70], MobilenetV2 [66]

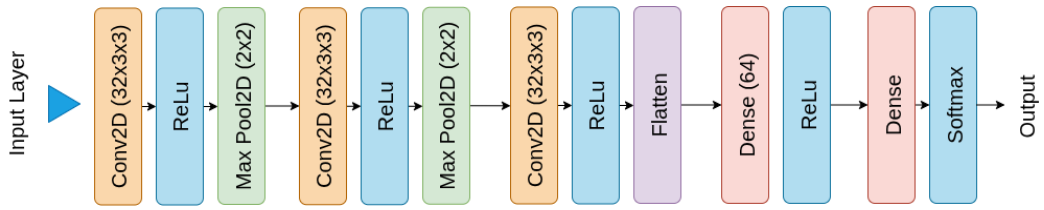

Figure 4.1. CNN model used in our experiments

Table 4.3. Architecture of the ResNet9+ model used for CIFAR100

Block	Details	Input
block 1	Conv2d(i=3, o=64, k=(3, 3), s=(1, 1)) GroupNorm(g=32, o=64) ReLU()	image
block 2	Conv2d(i=64, o=128, k=(3, 3), s=(1, 1)) GroupNorm(g=32, o=128) ReLU() MaxPool2d(k=(2, 2))	block 1

Continued on next page

Table 4.3. Architecture of the ResNet9+ model used for CIFAR100

Block	Details	Input
residual 1	Conv2d(i=128, o=128, k=(3, 3), s=(1, 1)) GroupNorm(g=32, o=128) ReLU() Conv2d(i=128, o=128, k=(3, 3), s=(1, 1)) GroupNorm(g=32, o=128) ReLU()	block 2
block 3	Conv2d(i=128, o=256, k=(3, 3), s=(1, 1)) GroupNorm(g=32, o=256) ReLU() MaxPool2d(k=(2, 2))	block 2 + residual 1
block 4	Conv2d(i=256, o=512, k=(3, 3), s=(1, 1)) GroupNorm(g=32, o=512) ReLU()	block 3
residual 2	Conv2d(i=512, o=512, k=(3, 3), s=(1, 1)) GroupNorm(g=32, o=512) ReLU() Conv2d(i=512, o=512, k=(3, 3), s=(1, 1)) GroupNorm(g=32, o=512) ReLU()	block 4
block 5	Conv2d(i=512, o=1024, k=(3, 3), s=(1, 1)) GroupNorm(g=32, o=1024) ReLU()	block 4 + residual 2
residual 3	Conv2d(i=1024, o=1024, k=(3, 3), s=(1, 1)) GroupNorm(g=32, o=1024) ReLU() Conv2d(i=1024, o=1024, k=(3, 3), s=(1, 1)) GroupNorm(g=32, o=1024) ReLU()	block 5
classifier	MaxPool2d(k=(2, 2)) Flatten() Linear(i=1024, o=100)	block 5 + residual 3

For the tabular datasets, we utilize a deep neural network (DNN), chosen for its widespread application in classification tasks involving tabular data [63]. Figure 4.2 depicts the layout of this DNN model. The architecture consists of an input layer, three hidden layers, and an output layer [53]. The input layer contains a number of units corresponding to the features in the training data. Each of the three hidden layers comprises 500 units, while the output layer contains neurons equal to the number of target classes. The hidden layers employ the ReLU activation function, and the output layer utilizes a SoftMax activation.

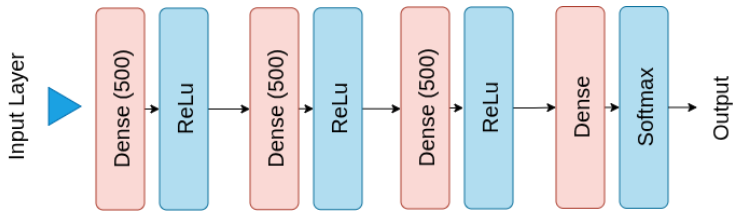

Figure 4.2. DNN model used in our experiments

4.3 Training configurations

All experiments were conducted in a **cross-silo**, **horizontal** federated learning setting:

- **Horizontal FL**: We begin with a centralized dataset and partition samples among clients according to the non-IID data skew patterns under investigation. Each client thus receives a distinct subset of examples.
- **Cross-Silo FL**: With a fixed, relatively small number of clients, our setup models a cross-silo scenario in which each client holds a disjoint partition of the global dataset. All clients share the same feature space but receive different subsets of the samples.

We used the Adam optimizer with a learning rate of 0.001 for $K = 30$ clients and a batch size of $B = 64$. In this work, all models were trained for $T = 40$ communication rounds and $E = 10$ local epochs per round, with the exception of the MOON aggregation technique and the CIFAR100 dataset, which were trained for $T = 100$ communication rounds to achieve adequate convergence. To ensure reproducibility and statistical reliability, all experiments were carried out with ten independent data partitions generated from fixed random seeds distributed among the datasets. These random seeds ensure that the data distribution among clients is consistent and that the model's starting weights are identical across trials using different aggregation procedures within each experimental configuration.

4.4 Aggregation Hyperparameter Tuning

To ensure a fair comparison among the different aggregation algorithms, we construct our hyperparameter grids using the top-performing values reported in the original studies, as detailed below:

- **FedAvg**: No specific tuning procedure is applied to this algorithm [49].
- **Rand**: The proportion of clients selected in each communication round is optimized by tuning over the set $0.3, 0.5, 0.7$ [11].
- **FedProx**: The μ parameter gets fine-tuned from {0, 0.001, 0.01, 0.1, 1, 10, 100} [40].
- **POC**: The parameter C is equal to 0.5. The parameter d gets fine-tuned from $\{15, 18, 19, 21\}$ [11].
- **MOON**: The μ is tuned from the grid of $\{0.1, 1, 5, 10\}$, and we find the best μ of 0.1, and we set the value of *temperature* to 0.5 [39].

4.5 FL Frameworks and Libraries

We implemented our experiments in Python 3.10.12, using a combination of specialized libraries and packages. NumPy [24] and Pandas [48] for data manipulation

and preparation; scikit-learn [59] for feature engineering tasks such as scaling and encoding, Plotly [61] for visualizations, and TensorFlow [1] and PyTorch [57] for defining and training neural network models. In addition to these general purpose packages we have used two FL dedicated frameworks. In the following we give an introduction regarding these frameworks and packages.

Flower [5]: is an open-source Python framework that simplifies the building and implementation of FL systems. It offers a flexible client-server API that abstracts away low-level networking concerns, allowing researchers and practitioners to focus on algorithm design and performing experiments rather than communication plumbing. Flower supports major machine learning backends (such as TensorFlow, PyTorch, and Keras), allowing you to use existing models and training loops as federated clients, while its server component handles client orchestration, scheduling, and aggregation. Its modular architecture allows for easy experimentation with new aggregation rules, client-selection strategies, and scalability optimizations, from tiny cross-silo deployments to huge cross-device networks.

FedArtML [31]: is a Python package for creating and analyzing non-IID data data partition in FL setting. The tool employs many partitioning algorithms, including Dirichlet-based label skew and feature skews, MinSize-Dirichlet quantity skew, and spatio-temporal skew. It also computes statistical distance measures, such as HD, to measure heterogeneity of the data among the clients. FedArtML integrates smoothly with federated frameworks like as Flower, allowing researchers to systematically test aggregation and client-selection algorithms under controlled skew conditions.

4.6 Hardware Specification

All configurations and trials were conducted using the facilities offered by Amazon Web Services (AWS) such as EC2 instances provisioned with Ubuntu 22.04.4 LTS, each having an Intel Xeon Platinum 8259CL CPU @ 2.50 GHz (16 vCPUs), 125 GB of RAM, and 200 GB of SSD storage. To automate this process to test different configuration, we utilized AWS Simple Queue Service (SQS) as a job queue where each message had a specific configuration parameters, which EC2 workers grabbed, ran, and then write the desired results and metrics to an AWS DynamoDB database for long-term storage. This approach enabled us to execute hundreds of studies in parallel without requiring manual intervention. Finally, we created a lightweight Python API client that uses DynamoDB to fetch and aggregate results, allowing for rapid post-processing, charting, and analysis.

4.7 Performance Metrics

This section outlines the performance and convergence metrics employed in our experiments, along with the rationale for their selection.

Accuracy [71, 74, 51] measures the proportion of correctly classified samples relative to the total number of data points. Higher accuracy values correspond to

improved model performance. It is computed as:

$$Acc = \frac{\sum_{k=1}^{K} C_k}{\sum_{k=1}^{K} n_k} \tag{4.1}$$

where C_k denotes the count of correctly classified samples for client k, and n_k represents the total number of samples for that client. Experiments were conducted over five and ten independent runs with different random seeds. To ensure the results' robustness and reliability, we present the average accuracy along with the standard deviation across these trials, offering a thorough assessment of the model's performance consistency.

Number of times that performed the best [37] measures how often a given aggregation algorithm outperforms other methods across multiple experimental runs. A higher frequency reflects greater consistency and robustness. This metric is particularly useful for identifying algorithms that reliably achieve superior results across varying data partitions, which is crucial in FL due to the significant performance variability caused by non-IID data distributions.

Rounds-to-accuracy (RTA) [76] quantifies the minimum number of communication rounds required for the global model to reach at least 90% of the highest accuracy achieved among the aggregation algorithms. This metric serves as an indicator of the FL process's efficiency, with lower values signifying faster convergence of the model.

Chapter 5

Label Skew Results

This chapter investigates the impact of label skew in client data on model performance. Notably, the Covtype dataset exhibits label imbalance, whereas CIFAR10, FMNIST, CIFAR100, and Physionet are balanced datasets. The model accuracy achieved through CL serves as the baseline for evaluating the accuracy of models trained in the FL setting.

5.1 Synthetic Partitioning Method

Utilizing the FedArtML tool [31], we applied the Dirichlet distribution (DD) to divide data among clients according to label distribution. The DD produces random values that sum to one, governed by the parameter α. Larger α values (e.g., 1000) result in more similar local distributions across clients, whereas smaller values increase the likelihood that clients receive data predominantly from a single, randomly selected class [42]. It is important to note that the DD is a multivariate extension of the Beta distribution, which itself generalizes the Uniform distribution. Consequently, partitioning datasets using the DD leads to a skewed division of the data distribution [41].

We measure the degree of non-IID data in the clients' data partitions using the HD metric. The non-IID data levels, as quantified by HD, correspond approximately to the following values: $0.0, 0.25, 0.5, 0.75, 0.9$. Figure 5.1 illustrates the label skew distribution across thirty clients using CIFAR10 dataset. In the IID scenario ($\alpha = 1000, \text{HD} = 0.0$), all ten classes are evenly distributed among clients. As the α parameter in the Dirichlet distribution decreases, class distributions among clients become increasingly heterogeneous. At the extreme setting of $\alpha = 0.03, \text{HD} = 0.9$, some clients lack certain classes entirely.

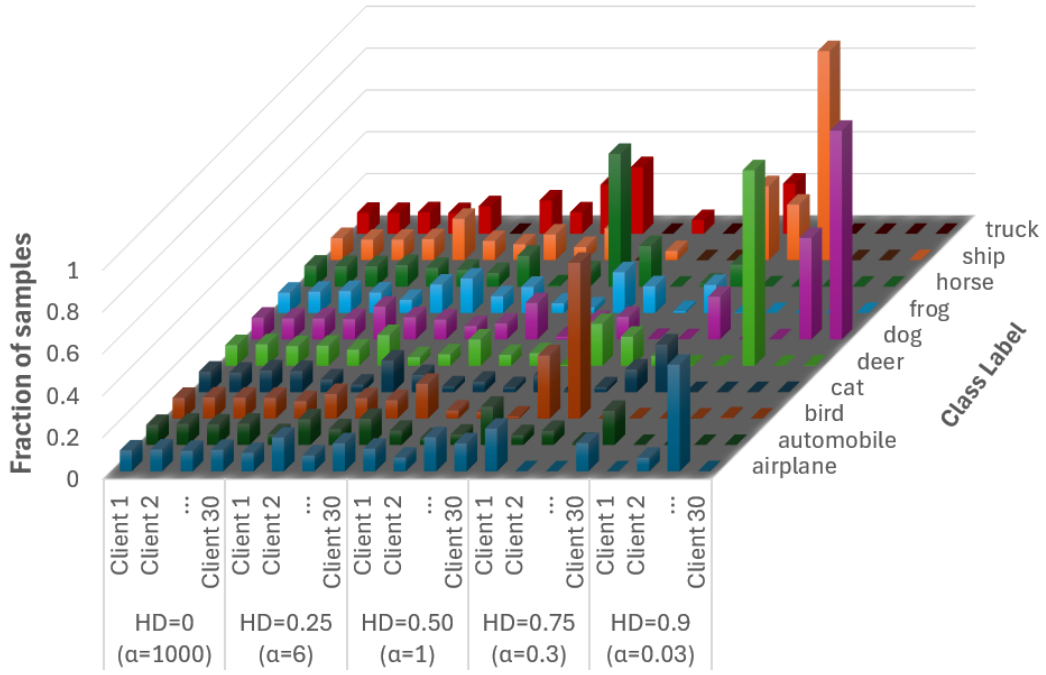

Figure 5.1. Distribution of CIFAR10 among 30 clients for different levels of non-IID data. The x-axis shows distinct α values used to partition the data and the resulting HD for clients from 1 to 30. The y-axis shows the participation of each class depicted on the z-axis.

5.2 Classification Power

This section details the simulation outcomes that compare the classification power (a.k.a accuracy) of multiple aggregation algorithms across differnt datasets.

Highlight 1: *The decline in model performance due to label skew occurs at two thresholds, with a significant drop becoming apparent once the HD surpasses 0.5 and again at 0.75.*

Previous studies have demonstrated that data non-IID data impacts the performance of FL models [45, 46, 30]. However, for the first time, we reveal that the effect of non-IID data varies across different degrees of heterogeneity. Figure 5.2 illustrates how accuracy changes as the non-IID data of client data distributions, measured by HD, deviates from the baseline CL model. Model accuracy declines as the level of non-IID data increases, with a particularly sharp decrease observed when the HD between client data distributions surpasses 0.75.

A possible reason for this double-threshold phenomenon is that once the HD exceeds 0.5, the model begins to experience a noticeable performance decline caused by increasing divergence among local data distributions, which adversely affects the global model's ability to generalize. When the HD surpasses 0.75, heterogeneity may reach a critical level where client models become excessively specialized to their individual local data, substantially diminishing the effectiveness of global aggrega-

Table 5.1. Mean and standard deviation Accuracy for each dataset for CL, FedAvg, Rand, FedProx, Power-Of-Choice, and MOON, considering different levels of non-IID data as measured by HD for $K = 30$. Each model has undergone ten different trials (random seeds).

Category	Dataset	HD	CL	FedAvg	Rand	FedProx	POC	MOON
Label distribution skew	CIFAR10	0	70.50% ± 0.60%	66.12% ± 0.73%	66.16% ± 0.74%	**66.35% ± 0.72%**	66.26% ± 0.70%	64.45% ± 1.05%
		0.25		**65.91% ± 0.49%**	65.49% ± 0.52%	65.86% ± 0.60%	65.61% ± 0.67%	63.40% ± 0.74%
		0.5		63.41% ± 0.95%	62.93% ± 1.55%	**63.56% ± 0.83%**	62.86% ± 1.71%	60.95% ± 1.33%
		0.75		**58.85% ± 1.06%**	56.80% ± 2.24%	58.80% ± 1.04%	55.84% ± 1.48%	55.27% ± 0.51%
		0.9		43.22% ± 2.24%	40.95% ± 2.43%	**44.33% ± 2.83%**	39.04% ± 2.99%	38.84% ± 2.31%
	FMNIST	0	90.90% ± 0.20%	90.68% ± 0.18%	90.63% ± 0.18%	**90.69% ± 0.15%**	90.62% ± 0.17%	88.70% ± 0.27%
		0.25		90.44% ± 0.14%	**90.52% ± 0.17%**	90.51% ± 0.17%	90.51% ± 0.18%	88.17% ± 0.22%
		0.5		89.92% ± 0.11%	89.84% ± 0.31%	**89.96% ± 0.20%**	89.74% ± 0.35%	87.37% ± 0.22%
		0.75		88.15% ± 0.54%	87.51% ± 0.81%	**88.17% ± 0.47%**	87.23% ± 0.78%	84.70% ± 0.84%
		0.9		80.37% ± 3.78%	79.08% ± 4.67%	**81.10% ± 2.21%**	77.83% ± 3.28%	70.79% ± 5.73%
	CIFAR100	0	67.47% ± 0.46%	62.88% ± 0.28%	62.72% ± 0.35%	**63.05% ± 0.12%**	62.80% ± 0.38%	56.51% ± 0.30%
		0.25		**62.66% ± 0.35%**	62.45% ± 0.25%	62.55% ± 0.32%	62.30% ± 0.21%	56.32% ± 0.49%
		0.5		61.72% ± 0.60%	61.49% ± 0.39%	**61.74% ± 0.26%**	61.23% ± 0.17%	56.47% ± 0.19%
		0.75		59.47% ± 0.37%	58.46% ± 0.73%	**59.72% ± 0.51%**	59.10% ± 0.25%	56.24% ± 0.42%
		0.9		54.38% ± 0.69%	52.87% ± 1.17%	**54.80% ± 0.63%**	52.85% ± 0.99%	51.45% ± 1.18%
	Physionet	0	63.74% ± 1.24%	57.97% ± 0.49%	57.48% ± 0.40%	58.16% ± 0.62%	57.86% ± 0.76%	**61.80% ± 0.62%**
		0.25		57.65% ± 0.47%	57.42% ± 0.54%	57.79% ± 0.55%	57.48% ± 0.53%	**60.94% ± 0.60%**
		0.5		55.69% ± 0.93%	55.26% ± 1.05%	56.29% ± 1.00%	55.24% ± 1.48%	**58.76% ± 0.71%**
		0.75		50.88% ± 1.18%	50.19% ± 2.07%	51.47% ± 1.20%	49.51% ± 2.30%	**53.51% ± 1.37%**
		0.9		41.35% ± 2.70%	39.68% ± 3.01%	41.95% ± 2.07%	38.81% ± 3.68%	**42.49% ± 3.00%**
	Covtype	0	95.60% ± 0.10%	94.95% ± 0.06%	94.89% ± 0.08%	94.96% ± 0.09%	94.84% ± 0.10%	**95.64% ± 0.06%**
		0.25		92.05% ± 1.14%	91.63% ± 1.52%	92.10% ± 1.28%	**93.89% ± 0.73%**	93.36% ± 1.18%
		0.5		84.92% ± 3.71%	83.10% ± 2.79%	85.54% ± 3.39%	**88.21% ± 2.17%**	84.61% ± 3.44%
		0.75		**77.55% ± 3.63%**	76.25% ± 4.10%	77.55% ± 3.64%	74.68% ± 5.31%	57.79% ± 12.66%
		0.9		59.10% ± 8.70%	59.02% ± 9.19%	**59.46% ± 8.74%**	57.29% ± 10.72%	50.51% ± 5.57%
Number of times that performed the best				4	1	12	2	6

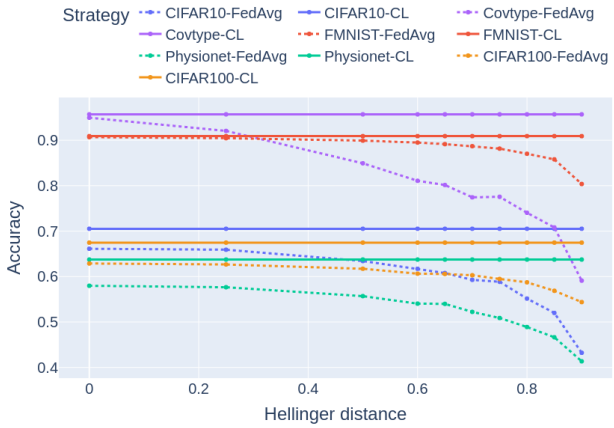

Figure 5.2. Changes in the models' accuracy considering different levels of non-IID data measured by HD for $K = 30$.

tion. This pronounced drop in accuracy indicates that under extreme non-IID data scenarios, FedAvg faces difficulty in converging to a well-generalized solution, likely due to conflicting optimization directions arising from highly dissimilar client updates.

Highlight 2: *When the level of non-IID data is high, the performance of transfer learning models declines significantly.*

Figure 5.3 displays the accuracy of models developed with three architectures: the previously described CNN, EfficientNetB0, and MobileNetV2, the latter two employing transfer learning. The figure clearly demonstrates a drop in model performance at two key thresholds, corresponding to HD values of 0.5 and 0.75.

Notably, the transfer learning models underperform compared to the simpler CNN model in extreme non-IID data scenarios.

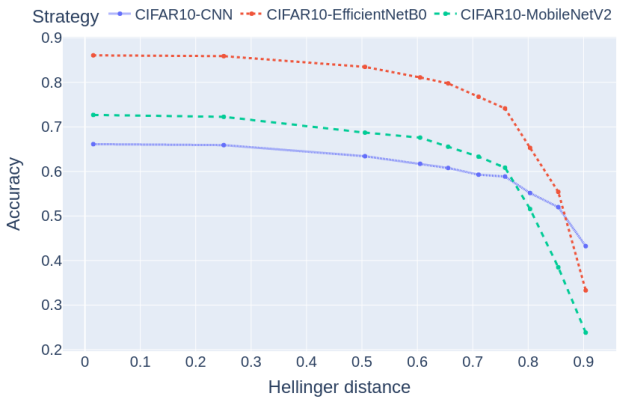

Figure 5.3. Changes in the models' accuracy considering different levels of non-IID data measured by HD for $K = 30$ using CNN, EfficientNetB0, and MobileNetV2 on the CIFAR10 dataset.

The more significant decrease in performance of transfer learning models under high non-IID data conditions is attributable to their use of fixed feature extractors, which restricts their ability to adjust to heterogeneous data. As HD rises, local updates to the final layers lead to inconsistent feature representations, weakening the impact of global aggregation. In comparison, the CNN trained from scratch is better equipped to adapt to decentralized data, making it more resilient in scenarios with extreme non-IID data.

Highlight 3: *Aggregation algorithms like Rand, POC, and MOON are especially susceptible to declines in performance when faced with high levels of non-IID data.*

Consider the scenario where the HD is 0.9, indicating high non-IID data, across all datasets listed in Table 5.1. When comparing the accuracy of Rand and POC to their respective centralized learning (CL) baselines, these algorithms exhibit a more pronounced decline in performance compared to other aggregation methods. In FL settings characterized by high non-IID data, the distribution of data labels among clients is highly imbalanced, meaning some clients possess larger or differing data types than others. In such cases, Rand and POC may unintentionally select clients with skewed or unrepresentative data, resulting in poor generalization during the aggregation of their updates.

MOON experiences the most pronounced performance decline when transitioning from CL to FL. Its contrastive loss, which aims to align local and global representations, loses effectiveness when client data distributions differ significantly, as previous representations no longer provide reliable reference points. This misalignment intensifies performance deterioration, rendering MOON less suitable for environments with extreme non-IID data.

Highlight 4: *Datasets with imbalanced classes exhibit a more pronounced decline in performance when transitioning from IID data to highly non-IID data conditions compared to balanced datasets.*

Consider both the IID data scenario (HD = 0) and the most extreme non-IID data scenario (HD = 0.9) for each dataset, as shown in Table 5.2. It is notable that the performance drop (calculated as the difference in accuracy between HD = 0 and HD = 0.9) is more pronounced for the unbalanced dataset (Covtype) compared to the balanced datasets (CIFAR10, FMNIST, CIFAR100, and Physionet).

Table 5.2. The performance's decrease range of the model for the four datasets, moving from the lowest (HD=0) to the highest (HD=0.9) levels of non-IID data and considering FedAvg.

Dataset	(HD=0) - (HD=0.9)
CIFAR10	22.9%
FMNIST	10.31%
CIFAR100	8.5%
Physionet	16.62%
Covtype	35.85%

The more pronounced decline in performance observed in unbalanced datasets under severe non-IID data conditions results from the combined impact of class imbalance and non-IID data. In these scenarios, some classes may be disproportionately represented in certain clients while being almost entirely missing in others, causing local models to become biased. When aggregated, these biased updates do not accurately reflect the overall class distribution, leading to reduced generalization. Conversely, balanced datasets maintain a more uniform distribution of class information across clients, alleviating this issue and resulting in a milder performance drop.

5.3 Convergence

This section highlights the results derived from the CIFAR10 dataset, comparing various aggregation algorithms in terms of their learning behavior and training stability.

Highlight 5: *As the degree of label non-IID data increases, the more number of communication rounds is necessary to reach convergence [37].*

Table 5.3 evaluates the convergence of different algorithms from an alternative viewpoint. We assess the performance of each aggregation method independently within specific non-IID data scenarios. For each non-IID data level defined by HD, we measure the number of rounds each algorithm requires to attain 90% of its maximum accuracy. Across all aggregation methods, as the degree of non-IID data in client data partitions increases, more communication rounds are needed to achieve stable convergence. This observation is consistent with the results reported by Li et al. [37].

Table 5.3. RTA for FedAvg, Rand, FedProx, POC, and MOON reached in different levels of non-IID data conditions as determined by HD over CIFAR10 for K = 30.

Category	Dataset	Aggregation algorithm	HD = 0	HD = 0.25	HD = 0.5	HD = 0.75	HD = 0.9
Label distribution skew	CIFAR10	FedAvg	5	5	6	9	15
		Rand	5	5	6	10	14
		FedProx	5	5	6	9	15
		POC	5	5	6	8	15
		MOON	8	8	10	12	20

This behavior occurs because the data distributions across clients are highly heterogeneous, causing models trained on individual clients to be optimized primarily for their local data, which deviates from the global optimum. However, as training advances, the server's aggregated weights improve by incorporating information from the combined data of all clients, leading to enhanced overall performance.

Additionally, FedAvg, Rand, FedProx, and POC demonstrate comparable convergence patterns, reaching a stable point after approximately the same number of communication rounds. In contrast, MOON, as anticipated, takes a greater number of rounds to achieve equivalent convergence

Chapter 6

Feature Skew Results

This section explores the impact of feature skew in client data on model performance.

6.1 Synthetic Partitioning Method

To simulate feature skew, we utilized two distinct methods from FedArtML [31] to evaluate their characteristics:

Gaussian noise method: This method applies varying levels of Gaussian noise to each client's local dataset to create diverse feature distributions. Specifically, for client i, noise $\hat{x}$ is added based on a user-defined noise parameter σ, where $\hat{x} \sim \text{Gau}\left(\sigma \cdot \frac{i}{K}\right)$. Here, $\hat{x}$ denotes the modified features after noise is applied, and $\text{Gau}\left(\sigma \cdot \frac{i}{K}\right)$ is a Gaussian distribution with mean zero and variance $\sigma \cdot \frac{i}{K}$, with K being the total number of clients.

Hist-Dirichlet-based method: The process begins by characterizing each client's attributes using averaged values, which are then discretized through a binning procedure. Next, the distribution of each feature category across clients is determined using the Dirichlet distribution with a specified parameter α. Unlike the Gaussian Noise method, this approach allocates data to clients without altering the feature values. The level of non-IID data of features is quantified by measuring the HD among features across clients (FHD), with values ranging over $0, 0.25, 0.5, 0.75, 0.9$.

6.2 Classification Power

This subsection presents the simulation results comparing various aggregation algorithms and datasets in terms of their classification accuracy under conditions of feature skew across client data. Table 6.1 summarizes the performance of models trained with different aggregation algorithms in varying levels of non-IID data of feature distributions, measured by FHD.

Highlight 6: *Models trained via FL exhibit lower performance compared to those trained using CL.*

Table 6.1. Mean and standard deviation Accuracy for each dataset for CL, FedAvg, Rand, FedProx, Power-Of-Choice, and MOON, considering different levels of feature skewness measured by FHD for K = 30. Each model has undergone ten different trials (random seeds).

Category	Method	Dataset	FHD	CL	FedAvg	Rand	FedProx	POC	MOON
Feature distribution skew	Gaussian Noise	CIFAR10	0	70.50% ± 0.60%	66.12% ± 0.70%	66.16% ± 0.74%	**66.35% ± 0.72%**	66.26% ± 0.70%	64.45% ± 1.05%
			0.35	70.81% ± 0.45%	66.18% ± 0.57%	66.04% ± 0.44%	**66.29% ± 0.63%**	66.24% ± 0.50%	64.23% ± 0.53%
			0.75	70.44% ± 0.29%	66.17% ± 0.44%	66.31% ± 0.61%	**66.36% ± 0.56%**	66.27% ± 0.56%	64.58% ± 0.55%
			0.9	69.71% ± 0.03%	65.81% ± 0.82%	65.89% ± 0.72%	65.85% ± 0.68%	**65.90% ± 0.77%**	63.52% ± 0.66%
		FMNIST	0	90.90% ± 0.20%	90.68% ± 0.18%	90.57% ± 0.14%	**90.69% ± 0.15%**	90.62% ± 0.17%	88.70% ± 0.27%
			0.35	90.91% ± 0.26%	90.69% ± 0.19%	**90.97% ± 0.15%**	90.71% ± 0.17%	90.97% ± 0.17%	88.67% ± 0.28%
			0.75	90.96% ± 0.11%	90.54% ± 0.09%	90.53% ± 0.11%	**90.57% ± 0.23%**	90.51% ± 0.19%	88.64% ± 0.19%
			0.9	90.76% ± 0.13%	90.59% ± 0.09%	**90.71% ± 0.15%**	90.70% ± 0.29%	90.51% ± 0.11%	88.55% ± 0.21%
		Physionet	0	63.74% ± 1.24%	57.97% ± 0.49%	57.48% ± 0.40%	58.16% ± 0.62%	57.86% ± 0.76%	**61.80% ± 0.62%**
			0.35	63.30% ± 1.31%	57.89% ± 0.39%	57.92% ± 0.64%	58.08% ± 0.61%	57.47% ± 1.13%	**61.22% ± 0.72%**
			0.75	60.13% ± 1.11%	52.81% ± 0.83%	52.84% ± 0.74%	52.28% ± 0.38%	51.39% ± 0.25%	**56.10% ± 0.90%**
			0.9	28.97% ± 3.22%	29.43% ± 1.49%	29.88% ± 1.27%	29.43% ± 1.15%	25.25% ± 1.97%	**32.06% ± 1.30%**
		Covtype	0	95.60% ± 0.10%	94.95% ± 0.06%	94.89% ± 0.08%	94.96% ± 0.09%	94.84% ± 0.10%	**95.53% ± 0.04%**
			0.35	95.68% ± 0.10%	94.94% ± 0.06%	94.90% ± 0.04%	94.90% ± 0.08%	94.88% ± 0.08%	**95.65% ± 0.06%**
			0.75	95.53% ± 0.04%	94.79% ± 0.08%	94.62% ± 0.04%	94.74% ± 0.07%	94.68% ± 0.13%	**95.11% ± 0.07%**
			0.9	68.53% ± 1.49%	50.01% ± 0.35%	49.81% ± 0.50%	50.03% ± 0.42%	**50.10% ± 1.67%**	49.20% ± 0.11%
	Hist-Dirichlet	CIFAR10	0	70.50% ± 0.60%	**66.42% ± 0.34%**	66.35% ± 0.35%	66.21% ± 0.59%	66.40% ± 0.70%	64.79% ± 0.32%
			0.25		66.09% ± 0.49%	66.13% ± 0.48%	65.82% ± 0.46%	**66.15% ± 0.53%**	65.12% ± 0.85%
			0.5		66.30% ± 0.67%	66.04% ± 0.66%	66.22% ± 0.40%	**66.52% ± 0.56%**	64.52% ± 1.08%
			0.75		66.23% ± 0.33%	66.04% ± 0.67%	66.17% ± 0.55%	**66.34% ± 0.51%**	64.99% ± 0.31%
			0.9		**66.25% ± 0.47%**	65.25% ± 0.51%	65.25% ± 0.90%	66.17% ± 0.47%	64.12% ± 0.56%
		FMNIST	0	90.90% ± 0.20%	**90.72% ± 0.20%**	90.68% ± 0.12%	90.68% ± 0.15%	90.59% ± 0.22%	88.75% ± 0.13%
			0.25		90.62% ± 0.10%	90.66% ± 0.27%	**90.70% ± 0.15%**	90.62% ± 0.18%	88.72% ± 0.32%
			0.5		**90.78% ± 0.12%**	90.66% ± 0.09%	90.63% ± 0.17%	90.78% ± 0.23%	88.43% ± 0.30%
			0.75		90.53% ± 0.24%	90.67% ± 0.18%	**90.73% ± 0.17%**	90.56% ± 0.12%	88.26% ± 0.30%
			0.9		89.77% ± 0.32%	89.73% ± 0.35%	89.66% ± 0.24%	**89.81% ± 0.29%**	87.38% ± 0.16%
		Physionet	0	63.74% ± 1.24%	57.73% ± 0.72%	57.76% ± 0.51%	58.02% ± 0.76%	57.52% ± 0.68%	**61.13% ± 0.42%**
			0.25		57.67% ± 0.61%	57.62% ± 0.69%	58.16% ± 0.63%	57.05% ± 0.37%	**61.91% ± 0.39%**
			0.5		57.92% ± 0.40%	57.50% ± 0.68%	57.86% ± 0.33%	57.35% ± 0.47%	**61.20% ± 0.83%**
			0.75		57.27% ± 0.64%	57.18% ± 0.97%	57.34% ± 0.88%	56.99% ± 0.86%	**62.11% ± 0.44%**
			0.9		56.47% ± 0.80%	55.49% ± 1.34%	56.49% ± 0.60%	55.80% ± 1.24%	**59.82% ± 1.02%**
		Covtype	0	95.60% ± 0.10%	94.95% ± 0.03%	94.81% ± 0.02%	95.00% ± 0.03%	**98.84% ± 0.09%**	95.62% ± 0.11%
			0.25		94.95% ± 0.05%	94.83% ± 0.09%	94.97% ± 0.09%	94.77% ± 0.03%	**95.63% ± 0.05%**
			0.5		94.90% ± 0.02%	94.88% ± 0.11%	94.90% ± 0.07%	94.88% ± 0.02%	**95.65% ± 0.02%**
			0.75		94.80% ± 0.05%	94.67% ± 0.10%	94.78% ± 0.08%	94.74% ± 0.10%	**95.57% ± 0.07%**
			0.9		93.30% ± 0.42%	93.11% ± 0.52%	93.38% ± 0.31%	93.47% ± 0.33%	**94.51% ± 0.07%**
Number of times that performed the best					4	2	7	7	16

Shifting the training approach from CL to FL results in a performance decline, which is particularly evident in image datasets such as CIFAR10 compared to tabular datasets like Covtype. This outcome is expected, as models are trained without access to the full dataset, with each client optimizing weights using only its local data. The larger drop in performance observed in image datasets is attributed to the greater complexity of classification tasks in this domain relative to tabular data.

Highlight 7: *The performance of models on image datasets remains stable despite increases in non-IID data of features [37].*

Consider only the image datasets (CIFAR10, FMNIST) and the models generated in FL. Regardless of the aggregation algorithm employed, the performance of the final model remains stable across different levels of non-IID data of features, consistently converging to specific values for each aggregation method. This behavior is consistent with the observations reported by Li et al. [37].

This pattern results from the inherent robustness of convolutional layers, which capture spatial features and mitigate minor pixel variations. In the Gaussian noise approach, slight perturbations have minimal effect on critical patterns because convolutional filters effectively smooth out the noise. Additionally, deeper layers aggregate features further, maintaining important information and reducing the impact on overall performance.

Highlight 8: *For tabular datasets, applying Gaussian noise levels above FHD = 0.9 leads to a significant decrease in model performance, highlighting the pronounced dissimilarity between samples.*

Using tabular datasets such as Covtype and Physionet with the Hist-Dirichlet method indicates that increasing non-IID data of features does not affect performance. In contrast, when Gaussian noise is applied, performance significantly decreases once FHD exceeds 0.9.

This performance decrease arises because the data become highly heterogeneous and noisy, affecting the model's ability to identify meaningful patterns. Even in CL, where the data is generally more consistent, excessive noise interferes with feature extraction, diminishing the model's ability to generalize and resulting in degraded performance.

Highlight 9: *In situations where features are distributed non-IID among clients, MOON outperforms all other aggregation algorithms when applied to tabular datasets.*

Table 6.1 confirms that no algorithm significantly outperforms the others in image datasets, as their final performance metrics are similar. However, MOON stands out as the best-performing algorithm for tabular datasets, outperforming all others and achieving results nearly comparable to models trained using CL.

This difference arises because MOON can immediately begin learning meaningful contrasts between label differences using the explicit features provided in tabular datasets. In contrast, with image data, the model must first learn to extract relevant features from raw pixel input before it can effectively distinguish between different object classes. For example, in the Physionet dataset, features such as age, sex, heart rate, and P-R interval have clear medical meanings, allowing the model to use these values directly without the need to learn initial representations. In contrast, for CIFAR10, the model must first develop meaningful feature representations from raw pixels to enable effective contrastive learning.

6.3 Convergence

This subsection focuses on the simulation results, with the goal of comparing different aggregation methods and datasets based on their convergence behavior.

Highlight 10: *Feature skew does not affect the point at which the model converges.*

Table 6.2 offers a different perspective on this observation. It details the number of iterations required for FedAvg, Rand, FedProx, POC, and MOON to reach 90% of their maximum accuracy under varying levels of non-IID data of features, measured by FHD, using the CIFAR10 dataset. The results indicate that increasing the non-IID data of features has little effect on the model's capacity to converge to optimal performance.

The limited effect of feature skew on convergence indicates that although feature distributions vary among clients, the fundamental learning task remains achievable.

Table 6.2. RTA for FedAvg, Rand, FedProx, POC, and MOON reached in different levels of non-IID data of features as determined by FHD over CIFAR10 and Covtype using Hist Dirichlet for K = 30

Category	Method	Dataset	Aggregation algorithm	FHD = 0	FHD = 0.25	FHD = 0.50	FHD = 0.75	FHD = 0.9
Feature distribution skew	Hist Dirichlet	CIFAR10	FedAvg	5	5	5	5	4
			Rand	5	5	5	5	4
			FedProx	5	5	5	5	4
			POC	5	5	5	5	4
			MOON	8	7	7	8	7
		Covtype	FedAvg	3	3	3	3	4
			Rand	3	3	3	3	4
			FedProx	3	3	3	3	4
			POC	3	3	3	3	4
			MOON	4	4	4	4	5

In contrast to label skew, which directly influences class representation in local updates, feature skew mainly modifies input variations without significantly affecting the overall decision boundary. Consequently, the global model retains its ability to generalize well across clients, resulting in comparable convergence patterns regardless of the extent of non-IID data of features.

Chapter 7

Quantity Skew Results

This section explores the impact of quantity skew in client data on model performance.

7.1 Synthetic Partitioning Method

We utilize the MinSize-Dirichlet method from the FedArtML [31] tool, which defines the Dirichlet distribution parameter α and generates target participation proportions for each client. A minimum required size, termed the "minimum number of examples," is then set for each client. The minimum proportion size, $MinSize$, is computed as $MinSize = \frac{MinRequiredSize}{n}$, where n is the total number of examples in the centralized dataset. If any assigned proportion is less than $MinSize$, it is replaced with $MinSize$. Finally, the proportions are normalized to ensure they range between 0 and 1.

We evaluate the degree of non-IID data in quantity skew using the HD metric for quantity skew (QHD), within the range $0, 0.10, 0.17$. This limited range is due to the finite size of the dataset, which restricts the extent of quantity skew. Unlike other types of skew, the proportions based on quantity distribution cannot vary drastically because the total sample size limits how unevenly data can be allocated across clients.

Table 7.1. Mean and standard deviation Accuracy for each dataset for CL, FedAvg, Rand, FedProx, Power-Of-Choice, and MOON, considering different levels of non-IID data partitioning regarding record quantities measured by QHD for $K = 30$. Each model has undergone ten different trials.

Category	Method	Dataset	CL	QHD	FedAvg	Rand	FedProx	POC	MOON
Quantity distribution skew	Min-size Dirichlet	CIFAR10	70.50% ± 0.6%	0	65.77% ± 0.57%	65.92% ± 0.72%	**66.45% ± 0.61%**	66.04% ± 0.35%	64.01% ± 0.46%
				0.10	66.69% ± 0.72%	66.29% ± 0.67%	66.71% ± 0.76%	**68.82% ± 0.89%**	63.24% ± 0.37%
				0.17	66.04% ± 0.65%	67.91% ± 0.46%	68.07% ± 0.42%	**68.44% ± 0.51%**	63.49% ± 1.25%
		FMNIST	90.90% ± 0.02%	0	**90.72% ± 0.23%**	90.69% ± 0.25%	90.64% ± 0.15%	90.67% ± 0.11%	88.49% ± 0.24%
				0.10	90.37% ± 0.16%	90.39% ± 0.22%	90.30% ± 0.45%	**90.78% ± 0.13%**	88.04% ± 0.27%
				0.17	90.39% ± 0.32%	90.43% ± 0.19%	90.44% ± 0.27%	**90.65% ± 0.31%**	88.10% ± 0.26%
		Physionet	63.74% ± 1.24%	0	58.23% ± 0.68%	57.13% ± 0.56%	58.04% ± 0.29%	57.08% ± 0.53%	**61.29% ± 0.81%**
				0.10	59.48% ± 2.09%	59.06% ± 1.71%	59.42% ± 1.23%	**61.29% ± 0.50%**	58.67% ± 1.97%
				0.17	64.22% ± 1.27%	64.40% ± 0.55%	**64.92% ± 0.71%**	64.82% ± 0.85%	63.86% ± 0.40%
		Covtype	95.60% ± 0.1%	0	94.97% ± 0.04%	94.87% ± 0.06%	94.96% ± 0.07%	94.83% ± 0.05%	**95.15% ± 0.09%**
				0.10	95.67% ± 0.10%	95.65% ± 0.07%	95.70% ± 0.10%	**95.79% ± 0.10%**	95.13% ± 0.12%
				0.17	90.65% ± 1.57%	94.77% ± 0.43%	**95.27% ± 0.20%**	94.94% ± 0.44%	94.23% ± 0.47%
Number of times that performed the best					1	0	3	6	2

7.2 Classification Power:

In the subsequent paragraphs, we focus on the simulation results to assess different aggregation methods and datasets in terms of their classification accuracy, with a particular emphasis on the effects of quantity skew in client data.

Highlight 11: Quantity skew in client data does not impact the final model's performance [37].

An analysis of Table 7.1, evaluating each aggregation algorithm individually, reveals that the final model performance remains stable across different degrees of quantity skew in the clients' data. This consistency holds true regardless of the aggregation method used, reinforcing the conclusion that quantity skew does not influence model performance. These findings align with the results reported by Li et al. [37].

The consistency of model performance despite the presence of the quantity skew indicates that FL aggregation algorithms effectively balance client updates irrespective of differences in sample sizes. Clients with fewer samples still contribute meaningful gradients proportional to their data, without exerting undue influence on the training process. Moreover, common optimization strategies like weighted averaging help counteract potential biases arising from data imbalance, maintaining stable performance across varying degrees of quantity skew.

7.3 Convergence

This subsection concentrates on the results related to the learning behavior and training stability of various aggregation algorithms.

Highlight 12: *In the presence of quantity skew, all aggregation algorithms achieve convergence within the same number of communication rounds.*

Table 7.2. RTA for FedAvg, Rand, FedProx, POC, and MOON reached in different levels of quantity non-IID data cases as determined by QHD over CIFAR10 and Covtype using Min-size Dirichlet method for K = 30

Category	Method	Dataset	Aggregation algorithm	QHD = 0	QHD = 0.10	QHD = 0.17
Quantity distribution skew	Min-size Dirichlet	CIFAR10	FedAvg	5	3	2
			Rand	5	3	1
			FedProx	5	3	2
			POC	5	3	2
			MOON	6	3	2
		Covtype	FedAvg	3	2	1
			Rand	3	2	1
			FedProx	3	2	1
			POC	3	2	1
			MOON	4	2	1

As shown in Table 7.2, regardless of the level of quantity non-IID cases, all aggregation algorithms converge after a similar number of communication rounds.

The uniform convergence observed across aggregation algorithms is due to the redundancy in client datasets, where each client's data reflects the overall distribution. This redundancy enables the global model to capture similar patterns from any subset of clients, maintaining stable convergence despite the presence of quantity skew.

Chapter 8

Spatiotemporal Skew Results

This section explores the effect of different levels of non-IID data in space or time among clients on model performance.

8.1 Synthetic Partitioning Method

The key requirement for this partitioning method is that the dataset must include a categorical variable representing either space (such as locations, cities, latitude, longitude) or time (such as hours, months, years) to serve as the basis for partitioning. For example, Figure 8.1 illustrates the label distribution over time in the 5G NTF dataset. Here, the spatial variable used to generate the federated data is the flow's date, formatted categorically as year-month-day.

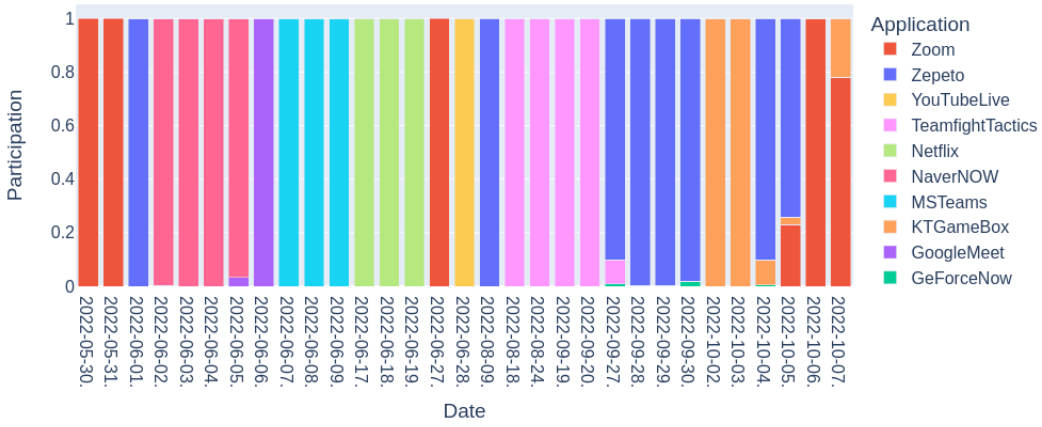

Figure 8.1. Distribution of 5G NTF applications (label) along date (spatiotemporal variable expressed in YYYY-MM-DD).

We apply the St-Dirichlet method from FedArtML [31], which utilizes the Dirichlet distribution to partition data according to spatial (SP skew) or temporal (TMP skew) categories for allocation among federated clients. The level of non-IID data in space or time is measured using the HD (STHD), evaluated in the range of {0, 0.25, 0.5, 0.75, 0.9}.

8.2 Classification Power

This section presents the simulation results used to assess various aggregation methods and datasets in terms of classification accuracy, with a specific emphasis on the effects of varying degrees of spatiotemporal skewness in clients' data.

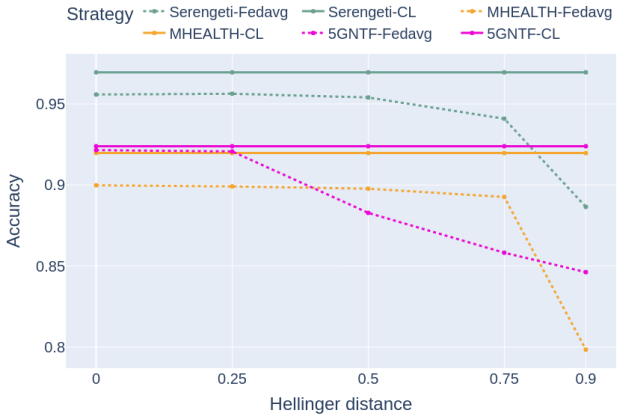

Figure 8.2. Changes in the models' accuracy considering different levels of non-IID data in space or time features measured by STHD for $K = 30$.

Table 8.1. Mean and standard deviation Accuracy for each dataset for CL, FedAvg, Rand, FedProx, Power-Of-Choice, and MOON, considering different levels of non-IID data partitioning of the records based on their space (SP) and time (TMP) measured by STHD for $K = 30$. Each model has undergone ten trials.

Category	Type	Method	Dataset	CL	STHD	FedAvg	Rand	FedProx	POC	MOON
SPT distribution skew	SP skew	St-Dirichlet	Serengeti	96.95% ± 0.11%	0	95.58% ± 0.08%	95.56% ± 0.09%	95.62% ± 0.14	95.48% ± 0.09	**96.69% ± 0.06**
					0.25	95.63% ± 0.05%	95.50% ± 0.10%	95.64% ± 0.03	95.51% ± 0.11	**96.76% ± 0.08**
					0.5	95.40% ± 0.08%	95.32% ± 0.07%	95.36% ± 0.06	95.32% ± 0.06	**96.51% ± 0.04**
					0.75	94.09% ± 0.25%	93.91% ± 0.36%	94.15% ± 0.21	94.06% ± 0.09	**95.80% ± 0.16**
					0.9	88.65% ± 0.18%	87.68% ± 0.58%	88.49% ± 0.13	88.20% ± 0.39	**91.74% ± 0.20**
			MHEALTH	91.97% ± 0.09	0	89.98% ± 0.04%	90.33% ± 0.03	**90.85% ± 0.04**	90.33% ± 0.05	90.56% ± 0.02
					0.25	89.91% ± 0.05	90.34% ± 0.04	**90.85% ± 0.01**	90.33% ± 0.05	90.57% ± 0.02
					0.5	89.77% ± 0.10	90.28% ± 0.04	**90.85% ± 0.02**	90.32% ± 0.04	90.48% ± 0.04
					0.75	89.26% ± 0.31	89.15% ± 0.24	**89.87% ± 0.31**	88.79% ± 0.48	89.69% ± 0.18
					0.90	79.84% ± 0.57	82.34% ± 1.36	82.48% ± 0.78	81.33% ± 0.74	**82.95% ± 1.04**
	TMP skew		5G NTF	92.39% ± 0.10%	0	92.15% ± 0.04%	92.15% ± 0.02%	92.14% ± 0.02%	92.15% ± 0.04%	**92.23% ± 0.02%**
					0.25	92.07% ± 0.12%	92.05% ± 0.15%	92.07% ± 0.18%	92.15% ± 0.05%	**92.28% ± 0.04%**
					0.5	88.28% ± 1.87%	87.92% ± 2.00%	89.19% ± 2.09%	**92.08% ± 0.08%**	89.24% ± 1.80%
					0.75	85.82% ± 0.06%	85.64% ± 0.08%	85.83% ± 0.08%	**91.77% ± 0.32%**	86.51% ± 1.79%
					0.9	84.62% ± 0.36%	84.50% ± 0.20%	84.55% ± 0.37%	**89.23% ± 2.13%**	83.94% ± 0.02%
Number of times that performed the best						0	0	4	3	8

Highlight 13: *The higher the level of non-IID data in space or time, the worse the model's performance.*

Table 8.1 shows that model performance declines across all aggregation algorithms as the data distribution among clients becomes more varied in terms of space or time. Figure 8.2 compares FedAvg with the CL model over different levels of STHD, revealing a consistent trend: accuracy decreases as the level of non-IID data in space or time increases among clients. The extent of this decline varies across datasets, with a marked drop in performance observed once STHD exceeds 0.75. This effect arises because increasing temporal and spatial disparities among clients also amplifies label distribution non-IID data.

This relationship is further illustrated in Table 8.2, which reports the HD of label distributions at different STHD levels across clients. Additionally, findings from the label skew analysis confirm that higher degrees of non-IID data in client label distributions adversely affect the final model's performance.

Table 8.2. HD among clients' label distributions at varying levels of non-IID partitioning in time and space

Dataset	STHD = 0	STHD = 0.25	STHD = 0.50	STHD = 0.75	STHD = 0.9
Serengeti	0.01	0.09	0.22	0.36	0.53
MHEALTH	0.01	0.01	0.01	0.03	0.07
5G NTF	0.03	0.20	0.29	0.30	0.49

8.3 Convergence

This section presents the results on model convergence in the presence of temporal and spatial data variations, based on experiments conducted with the Serengeti, MHEALTH, and 5G NTF datasets.

Table 8.3. RTA reached for different levels of non-IID data in space or time measured by STHD for K = 30.

Dataset	Aggregation algorithm	STHD = 0	STHD = 0.25	STHD = 0.50	STHD = 0.75	STHD = 0.9
Serengeti	FedAvg	6	7	7	10	14
	Rand	7	7	8	10	14
	FedProx	7	7	7	10	14
	POC	7	7	7	10	15
	MOON	8	8	9	12	19
5G NTF	FedAvg	1	1	1	1	1
	Rand	1	1	1	1	1
	FedProx	1	1	1	1	1
	POC	1	1	1	1	1
	MOON	1	1	1	1	1
MHEALTH	FedAvg	2	2	2	3	2
	Rand	2	2	2	3	2
	FedProx	2	2	2	3	2
	POC	2	2	2	4	2
	MOON	6	7	7	13	14

Highlight 14: *Non-IID data in space or time does not consistently affect the number of convergence rounds, with its impact differing depending on the dataset characteristics and the complexity of the task.*

Table 8.3 presents the *RTA* for five aggregation algorithms, indicating the number of rounds needed to achieve 90% maximum precision in the reference datasets as the level of non-IID data in space or time increases (STHD values $\in \{0, 0.25, 0.50, 0.75, 0.90\}$). The impact of a given non-IID data level can vary significantly depending on the characteristics of the underlying data:

- **Serengeti**: As STHD rises from 0 to 0.90, the RTA approximately doubles for all aggregation methods (for instance, FedAvg increases from 6 to 14 rounds,

and POC from 7 to 15 rounds). This indicates that as non-IID data across different locations intensifies, additional training rounds are needed for the models to achieve convergence.

- **5G NTF**: All aggregation methods achieve convergence within a single round regardless of the STHD level. This happens when the classes are easily separable, with a distinct separation between the records of one class and those of the others. Under these conditions, the task is straightforward, and temporal variations have little effect on the convergence timing.
- **MHEALTH**: For all aggregation methods, the RTA remains stable except for MOON, which experiences a significant increase from 6 to 14 rounds as the level of non-IID data in space or time rises from 0 to 0.90. Because the data originates from body-worn sensors and is partitioned by individual subjects, client-specific covariate shifts arise, which particularly challenge representation-based methods such as MOON.

For the MOON algorithm, the difference in RTA between STHD = 0 and STHD = 0.9 varies across datasets, showing no change for 5G NTF, an 8-round increase for MHEALTH, and an 11-round increase for Serengeti—while other aggregation methods exhibit no such variation. This suggests that there is not a direct, one-to-one correlation between STHD and convergence speed; rather, factors such as task complexity, temporal patterns, and feature diversity influence the results. These findings reinforce our earlier observation that non-IID data in space or time affects convergence in a dataset-dependent and problem-specific manner.

Chapter 9

General Results

This section presents a summary of the key findings from our experiments, integrating the observed behaviors related to label, feature, quantity, and spatiotemporal skews.

Highlight 15: *Label skew [71, 37] and spatiotemporal skew have a substantial effect on model performance.*

Our experimental results indicate that different types of non-IID data affect FL performance to varying degrees. Label skew and spatiotemporal skew have the most pronounced negative effects. Specifically, label skew leads to a 10–40% reduction in model accuracy compared to the CL baseline. In contrast, feature and quantity skews cause smaller accuracy decreases, typically between 1–5%. These findings are consistent with prior research [71, 37], which highlights that label skew disproportionately impairs aggregation due to local models overfitting to dominant classes.

Spatiotemporal skew causes contextual drift, such as variations in sensor data across different locations or times, that distorts the feature space. Like label skew, this issue cannot be resolved by straightforward aggregation alone. Our experiments demonstrate that FedAvg experiences a 10–12% greater accuracy decline under spatiotemporal skew. The global model finds it challenging to generalize well across varied contexts because the averaging process diminishes critical environmental patterns that are specific to certain locations or time frames.

Highlight 16: *In general FL scenarios, FedProx, POC, and MOON outperform FedAvg and Rand.*

Table 9.1 summarizes the number of times that each aggregation algorithm outperformed the others in the four types of skewness examined in this study. In most cases, FedProx, POC, and MOON achieved superior performance, surpassing the simpler FedAvg and Rand algorithms. This superior performance is attributable to the unique strategies each aggregation algorithm employs to address the effect of the non-IID data. FedProx stabilizes training by controlling the impact of the global model on local clients, POC improves personalization by selecting clients based

Table 9.1. The number of cases in which each specific algorithm achieved the best performance for each study.

Study	#Cases	FedAvg	Rand	FedProx	POC	MOON
Label Skew	25	4	1	12	2	6
Feature Skew	36	4	2	7	7	16
Quantity Skew	12	1	0	3	6	2
Spatio Temporal Skew	15	0	0	4	3	8
Total best performance		9	3	26	18	32

on their loss values, and MOON utilizes contrastive learning to enhance feature representation. These approaches facilitate better adaptation to heterogeneous client distributions, resulting in consistently improved performance across various types of skewness.

Although FedProx, POC, and MOON typically outperform FedAvg and Rand, the performance improvements are often modest. This suggests that although these methods provide better handling of non-IID data, they do not completely overcome the challenges associated with the non-IID data. The relatively limited gains highlight the need for more advanced aggregation algorithms that can more effectively adapt to diverse client distributions and improve model performance in FL contexts.

Highlight 17: *FedProx demonstrates greater effectiveness on image datasets, whereas MOON yields superior results with tabular datasets.*

Table 9.2. The number of cases in which each aggregation algorithm achieved the best performance for each type of dataset

Dataset type	#Cases	FedAvg	Rand	FedProx	POC	MOON
Image	39	8	3	19	9	0
Tabular	49	1	0	7	9	32
Total best performance		9	3	26	18	32

Table 9.2 analyzes the top-performing aggregation algorithms based on the type of dataset used for training. It reveals that FedProx outperforms all other methods on image datasets in nineteen out of thirty-nine instances. Conversely, MOON generally outperforms competing algorithms on tabular datasets in thirty-two out of forty-nine cases.

The superior performance of FedProx on image datasets and MOON on tabular datasets can be explained by their unique optimization approaches. FedProx reduces client drift by stabilizing model updates, which is especially advantageous for handling complex, high-dimensional image data. Conversely, MOON's contrastive learning framework improves feature representation, making it better suited for tabular datasets where capturing intricate feature relationships is essential. These methodological differences account for their distinct performance across different dataset types.

Chapter 10

Discussion of Experimental Findings

In this chapter, we summarize and evaluate the important empirical findings of our systematic assessment of the effects of non-IID data in FL. We formed our discussion around the four types of non-IID data scenarios, such as label, feature, quantity, and spatiotemporal skews, and finished with general trends, practical implications, and some recommendations for FL practitioners.

10.1 Label Skew

Impact of non-IID data. Our experiments demonstrated that label skew has a highly non-linear effect on FL model performance. As HD between client label distributions increases from 0.5 to 0.75, the performance of the model encounters two noticeable critical points, with a sharp degradation after HD = 0.5 and HD = 0.75 (Figure 5.2).

Simple vs Complex aggregation algorithms. This double-threshold pattern in model performance degradation gives the idea that for the cases where the level of non-IID data is not high (e.g. HD $\leq$ 0.5), using simple aggregation algorithms such as FedAvg and Rand can still give us a comparable result compared to other descent aggregation algorithms such as FedProx and POC that requires extra work as the performance change is not that noticeable(Table 5.1), however, for more severe levels of non-IID data (e.g. HD $\geq$ 0.75), even existing state-of-the-art aggregation methods continue to experience a significant performance reduction of up to 35% in accuracy when comparing HD=0 to HD=0.9.

Balanced vs. Imbalanced datasets. Table 5.2 compares the performance decrease between the highest level of non-IID data (HD = 0.9) and the IID case (HD = 0), revealing that balanced datasets (e.g., CIFAR10, FMNIST) suffer considerably less degradation than unbalanced ones such as Covtype. For example, CIFAR10's accuracy decreases by not more than 23%, whereas Covtype's accuracy decreases by more than 35%, suggesting that unbalanced datasets suffer significantly more performance loss in FL scenarios.

Convergence. As the level of non-IID data measured by HD increases, all FL aggregation algorithms require more communication rounds to achieve 90% of their highest accuracy throughout the training. Table 5.3 shows that for FedAvg, Rand, FedProx and POC the RTA increases from 5 rounds at HD = 0 to 14-15 rounds at HD = 0.9, whereas MOON's RTA increases from 8 to 20 rounds. This slower convergence emphasizes the additional effort required for the aggregation algorithms to integrate models trained on various client datasets into a single global model.

10.2 Feature Skew

Impact of non-IID data. Unlike label skew, feature skew causes small decreases in performance of 1-5 percentage points when comparing the most extreme level of non-IID data (FHD = 0.9) with the IID scenario (FHD = 0). Both the Gaussian noise and Hist-Dirichlet partitioning methods demonstrate that the differences between aggregation methods are insignificant (Table 6.1). Practically, this suggests that standard FL algorithms are robust to non-IID data at the feature level unless clients' feature distributions diverge dramatically, as seen at the highest level of non-IID data in features created when FHD=0.9 using the Gaussian noise method, which attempts to change the values directly to create such a scenario.

Convergence. Feature skew has minimal effect on the convergence of the model. Table 6.2 shows that through FHD levels from 0 to 0.9 on CIFAR10 (an image dataset) and Covtype (a tabular dataset), all aggregation algorithms consistently require a similar number of rounds to achieve 90% of their highest accuracy (with only small variations for MOON). This consistency comes from the fact that different feature distributions change input appearance but not the primary decision boundary; therefore, the global model converges in about the same number of iterations regardless of non-IID data in features.

10.3 Quantity Skew.

Impact of non-IID data Quantity skew has no significant impact on the final model's performance. As shown in Table 7.1, all the aggregation algorithms achieve practically the same performance at different levels of non-IID data with respect to the quantity of samples over clients quantified by QHD. The consistency of model performance despite the presence of quantity skew indicates that FL aggregation algorithms effectively balance client updates irrespective of differences in sample sizes.

Convergence. Even when clients' sample sizes are different (quantity skew), all FL aggregation algorithms achieve the 90% of their highest accuracy in roughly the same number of rounds. This uniform convergence across QHD levels is due to the redundancy in client datasets, where the data for each client reflects the overall distribution that allows the global model to learn consistent patterns regardless of the participating clients.

10.4 Spatiotemporal Skew

Impact of non-IID data. Our findings reveal that spatiotemporal skew decreases FL performance across all aggregation algorithms. As the level of non-IID data in space or time measured by STHD increases, the model accuracy slowly drops, with a significant decrease when STHD approaches 0.75 (Figure 8.2, Table 8.1). Table 8.1 shows that when STHD increases, so does the level of non-IID data on labels, which explains why this form of skew is causing degradation in model performance.

Convergence. Our experiments show that spatial non-IID data has a dataset-dependent effect on convergence speed as when STHD increases from 0 to 0.9, RTA nearly doubles for Serengeti (e.g., for FedAvg 6→14, and for POC 7→15), remains at 1 round for 5G NTF regardless of the presence of this type of skew, and stays constant for MHEALTH except for MOON (6→14 rounds), highlighting that task complexity and data dynamics, not STHD alone, drive convergence change.

10.5 Aggregation Algorithms

In most cases, the more advanced aggregation algorithms, such as FedProx and MOON, outperform simpler algorithms, such as FedAvg and Rand. FedProx leads in 26 cases while MOON in 32 of the 88 total experiments (Table 9.1).

- **FedProx** Based on the results, FedProx is well suited for situations involving label skew in the client's data distribution (Table 9.1). Furthermore, it appears to work better on image datasets (Table 9.2), reducing client data drift by stabilizing model updates using its proximal term, which is especially useful for dealing with, complex high-dimensional image data.

- **MOON** performed best on tabular data (Table 9.2), the use of contrastive learning enhances feature representation, making it especially effective for tabular data that demands modeling complex inter-feature relationships

10.6 Practical Recommendations

Measure before Training. Calculate the HD on labels (and, where applicable, spatiotemporal variables) of the client's data to predict performance declines and select the appropriate aggregation algorithm for your scenario.

Threshold-Aware Strategy. When the level of non-IID data among clients is less than HD=0.5, you may use simpler aggregation techniques at the expense of very little performance loss but faster training because of their simplicity and no additional computation other than averaging the model updates, while if it exceeds HD = 0.5, prefer a more advanced algorithm such as FedProx, POC or MOON.

Algorithm Selection by Data Type. Used FedProx when you are dealing with image datasets, while preferring MOON for structured or tabular datasets.

By providing a comprehensive empirical foundation and clear guidelines, this discussion empowers FL practitioners to diagnose and mitigate non-IID challenges in diverse deployment settings.

Chapter 11

Conclusions

This study presents a thorough empirical investigation of the effects of non-IID data in FL. We evaluated five state-of-the-art approaches for managing non-IID data distributions under controlled settings, including label, feature, quantity, and spatiotemporal skews, with particular emphasis on the latter. Our goal is to establish a standardized methodology for analyzing data heterogeneity in FL employing the HD to quantify differences in data distributions. The results highlight the substantial influence of label and spatiotemporal skew on FL model performance, while feature skew and quantity do not influence the model performance. We also identify a double-threshold effect, where performance degradation intensifies sharply once HD exceeds 0.5 and 0.75. Furthermore, our findings indicate that FL performance is most severely impacted under extreme non-IID data conditions. Based on these insights, we provide practical recommendations for addressing non-IID data in FL. This work constitutes the most comprehensive study of non-IID data in FL to date and lays a strong foundation for future research in the field.

Chapter 12

Future Work

We offer design insights and identify opportunities to guide researchers in addressing the challenges posed by non-IID data.

Quantifying non-IID data. Numerous studies have reported that non-IID data affect the performance of FL models [45, 46, 30]. However, for the first time, we show that the effect of non-IID data varies depending on the degree of non-IID data, as illustrated in Figure 5.2. Thus, accurately quantifying the degree of non-IID data in FL is crucial. This study used the HD metric to assess the non-IID level. However, we encourage researchers to explore alternative metrics, including JSD [54], EMD [16], and Total Variation distance [6], among others.

More effective methods to tackle high non-IID data. This study demonstrates the performance of state-of-the-art methods for addressing non-IID data, such as Rand, POC, FedProx, and MOON, compared to FedAvg. The results indicate that no single algorithm consistently outperforms FedAvg in all scenarios. Furthermore, even the more advanced methods offer only marginal improvements over FedAvg in highly non-IID data settings, with gains typically limited to around two percentage points.

This phenomenon has been observed and discussed in limited empirical studies on non-IID data and related methodologies [71, 2, 51, 37]. Consequently, developing effective strategies to mitigate the impact of high non-IID data is essential for the advancement and sustainability of FL. This need corresponds to the open challenges highlighted by Kairouz et al. [32].

Focusing on highly unbalanced data. Our experiments indicate that the most significant performance drop when comparing CL to FL occurs in unbalanced datasets, reflecting the difficulty of learning from highly imbalanced and less representative data (see Table 5.1).

Therefore, it is essential to develop solutions for addressing non-IID data that consider the extent of label imbalance across clients.

Studying spatiotemporal skew. When this study was conducted, there were no existing analyses or empirical investigations on how spatiotemporal skew influences

the performance of FL models. Therefore, we present the first set of experiments to understand the impact of varying degrees of spatiotemporal non-IID data on FL model accuracy. The findings (see Table 8.1) indicate that high spatial or temporal skew levels significantly reduce model performance, particularly when HD exceeds 0.75, representing severe non-IID data.

Therefore, researchers can evaluate methods for addressing spatial and temporal skew in FL [67, 19, 88] better to understand their performance under conditions of high non-IID data.

Methods to compare mixed non-IID data types. Existing tools and techniques for synthetically partitioning centralized data into federated datasets [31, 83, 36, 56, 26] typically simulate a single type of non-IID data, such as label, feature, quantity, or spatiotemporal skew. However, more realistic scenarios involve combinations of multiple types of non-IID data, which can significantly impact the performance of the FL model. Therefore, to advance FL research, it would be valuable to develop partitioning methods that enable simultaneous control over multiple non-IID data factors when distributing centralized data across federated clients.

Bibliography

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: Large-scale machine learning on heterogeneous systems. arXiv preprint arXiv:1603.04467, 2016.

[2] Ahmed M Abdelmoniem, Chen-Yu Ho, Pantelis Papageorgiou, and Marco Canini. Empirical analysis of federated learning in heterogeneous environments. In *Proceedings of the 2nd European Workshop on Machine Learning and Systems*, pages 1–9, 2022.

[3] Niklas Babendererde, Moritz Fuchs, Camila Gonzalez, Yuri Tolkach, and Anirban Mukhopadhyay. Jointly exploring client drift and catastrophic forgetting in dynamic learning. *Scientific Reports*, 15(1):5857, 2025.

[4] Oresti Banos, Rafael Garcia, and Alejandro Saez. MHEALTH. UCI Machine Learning Repository, 2014. DOI: https://doi.org/10.24432/C5TW22.

[5] Daniel J Beutel, Taner Topal, Akhil Mathur, Xinchi Qiu, Javier Fernandez-Marques, Yan Gao, Lorenzo Sani, Kwing Hei Li, Titouan Parcollet, Pedro Porto Buarque de Gusmão, et al. Flower: A friendly federated learning research framework. *arXiv preprint arXiv:2007.14390*, 2020.

[6] Arnab Bhattacharyya, Sutanu Gayen, Kuldeep S Meel, Dimitrios Myrisiotis, Aduri Pavan, and NV Vinodchandran. On approximating total variation distance. *arXiv preprint arXiv:2206.07209*, 2022.

[7] Jock Blackard. Covertype. UCI Machine Learning Repository, 1998. DOI: https://doi.org/10.24432/C50K5N.

[8] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Byzantine-tolerant machine learning. *arXiv preprint arXiv:1703.02757*, 2017.

[9] Rahul Chauhan, Kamal Kumar Ghanshala, and RC Joshi. Convolutional neural network (cnn) for image detection and recognition. In *2018 first international conference on secure cyber computing and communication (ICSCCC)*, pages 278–282, IEEE, 2018. IEEE, IEEE.

[10] Yiqiang Chen, Xin Qin, Jindong Wang, Chaohui Yu, and Wen Gao. Fedhealth: A federated transfer learning framework for wearable healthcare. *IEEE Intelligent Systems*, 35(4):83–93, 2020.

[11] Yae Jee Cho, Jianyu Wang, and Gauri Joshi. Client selection in federated learning: Convergence analysis and power-of-choice selection strategies. *arXiv preprint arXiv:2010.01243*, 2020.

[12] Yae Jee Cho, Jianyu Wang, and Gauri Joshi. Towards understanding biased client selection in federated learning. In *International Conference on Artificial Intelligence and Statistics*, pages 10351–10375. PMLR, 2022.

[13] Yong-Hoon Choi, Daegyeom Kim, Myeongjin Ko, Kyung-yul Cheon, Seungkeun Park, Yunbae Kim, and Hyungoo Yoon. Ml-based 5g traffic generation for practical simulations using open datasets. *IEEE Communications Magazine*, 61(9):130–136, 2023.

[14] Marcos F Criado, Fernando E Casado, Roberto Iglesias, Carlos V Regueiro, and Senén Barro. Non-iid data and continual learning processes in federated learning: A long road ahead. *Information Fusion*, 88:263–280, 2022.

[15] Daily Dose of DS. Federated learning: A critical step towards privacy-preserving machine learning, 2021. Accessed: 2025-07-06.

[16] Adam Davis, Tony Menzo, Ahmed Youssef, and Jure Zupan. Earth mover's distance as a measure of cp violation. *Journal of High Energy Physics*, 2023(6):1–42, 2023.

[17] Haya Elayan, Moayad Aloqaily, and Mohsen Guizani. Deep federated learning for iot-based decentralized healthcare systems. In *2021 International Wireless Communications and Mobile Computing (IWCMC)*, pages 105–109. IEEE, 2021.

[18] Ahmed Elhussein and Gamze Gursoy. A universal metric of dataset similarity for cross-silo federated learning. *arXiv preprint arXiv:2404.18773*, 2024.

[19] Wenjie Fu, Xudong Zhang, Junlong Wang, Di Yang, Yuntong Lv, Yuqing Wang, Zhao Zhen, and Fei Wang. A spatiotemporal federated learning based distributed photovoltaic ultra-short-term power forecasting method. In *2023 IEEE/IAS 59th Industrial and Commercial Power Systems Technical Conference (I&CPS)*, pages 1–7. IEEE, 2023.

[20] Roma Goussakov. *Hellinger Distance-based Similarity Measures for Recommender Systems*. PhD thesis, Umea University, 2020.

[21] Mackenzie Graham, Richard Milne, Paige Fitzsimmons, and Mark Sheehan. Trust and the goldacre review: why trusted research environments are not about trust. *Journal of Medical Ethics*, 49(10):670–673, 2023.

[22] Shunxin Guo, Hongsong Wang, Shuxia Lin, Xu Yang, and Xin Geng. Sthfl: Spatio-temporal heterogeneous federated learning. *arXiv preprint arXiv:2501.05775*, 2025.

[23] Daniel Mauricio Jimenez Gutierrez, Hafiz Muuhammad Hassan, Lorella Landi, Andrea Vitaletti, and Ioannis Chatzigiannakis. Application of federated learning techniques for arrhythmia classification using 12-lead ecg signals. *arXiv preprint arXiv:2208.10993*, 2022.

[24] Charles R Harris, K Jarrod Millman, Stéfan J Van Der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, et al. Array programming with numpy. *Nature*, 585(7825):357–362, 2020.

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[26] Kevin Hsieh, Amar Phanishayee, Onur Mutlu, and Phillip Gibbons. The non-iid data quagmire of decentralized machine learning. In *International Conference on Machine Learning*, pages 4387–4398, unknown, 2020. PMLR, PMLR.

[27] Chao Huang, Jianwei Huang, and Xin Liu. Cross-silo federated learning: Challenges and opportunities, 2022.

[28] Wei Huang, Tianrui Li, Dexian Wang, Shengdong Du, Junbo Zhang, and Tianqiang Huang. Fairness and accuracy in horizontal federated learning. *Information Sciences*, 589:170–185, 2022.

[29] Yutao Huang, Lingyang Chu, Zirui Zhou, Lanjun Wang, Jiangchuan Liu, Jian Pei, and Yong Zhang. Personalized cross-silo federated learning on non-iid data. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 7865–7873, 2021.

[30] Hadi Jamali-Rad, Mohammad Abdizadeh, and Anuj Singh. Federated learning with taskonomy for non-iid data. *IEEE transactions on neural networks and learning systems*, 2022.

[31] G Daniel Mauricio Jimenez, Aris Anagnostopoulos, Ioannis Chatzigiannakis, and Andrea Vitaletti. Fedartml: A tool to facilitate the generation of non-iid datasets in a controlled way to support federated learning research. *IEEE Access*, 2024.

[32] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and trends® in machine learning*, 14(1–2):1–210, 2021.

[33] Sai Praneeth Karimireddy, Martin Jaggi, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian U Stich, and Ananda Theertha Suresh. Breaking the centralized barrier for cross-device federated learning. *Advances in Neural Information Processing Systems*, 34:28663–28676, 2021.

[34] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[35] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). *unknown*, 0(0):0, 2009.

[36] Fan Lai, Yinwei Dai, Sanjay Singapuram, Jiachen Liu, Xiangfeng Zhu, Harsha Madhyastha, and Mosharaf Chowdhury. Fedscale: Benchmarking model and system performance of federated learning at scale. In *International conference on machine learning*, pages 11814–11827. PMLR, 2022.

[37] Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. Federated learning on non-iid data silos: An experimental study. In *2022 IEEE 38th international conference on data engineering (ICDE)*, pages 965–978. IEEE, 2022.

[38] Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. Federated learning on non-iid data silos: An experimental study. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 965–978, IEEE, 2022. IEEE, IEEE.

[39] Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10713–10722, 2021.

[40] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020.

[41] Jiayu Lin. On the dirichlet distribution. *Department of Mathematics and Statistics, Queens University*, 40, 2016.

[42] Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. *Advances in Neural Information Processing Systems*, 33:2351–2363, 2020.

[43] Yang Liu, Yan Kang, Chaoping Xing, Tianjian Chen, and Qiang Yang. A secure federated transfer learning framework. *IEEE Intelligent Systems*, 35(4):70–82, 2020.

[44] Yang Liu, Yan Kang, Tianyuan Zou, Yanhong Pu, Yuanqin He, Xiaozhou Ye, Ye Ouyang, Ya-Qin Zhang, and Qiang Yang. Vertical federated learning: Concepts, advances, and challenges. *IEEE Transactions on Knowledge and Data Engineering*, 36(7):3615–3634, 2024.

[45] Zili Lu, Heng Pan, Yueyue Dai, Xueming Si, and Yan Zhang. Federated learning with non-iid data: A survey. *IEEE Internet of Things Journal*, 2024.

[46] Xiaodong Ma, Jia Zhu, Zhihao Lin, Shanxuan Chen, and Yangjie Qin. A state-of-the-art survey on solving non-iid data in federated learning. *Future Generation Computer Systems*, 135:244–258, 2022.

[47] Xiaodong Ma, Jia Zhu, Zhihao Lin, Shanxuan Chen, and Yangjie Qin. A state-of-the-art survey on solving non-iid data in federated learning. *Future Generation Computer Systems*, 135:244–258, 2022.

[48] Wes McKinney et al. pandas: a foundational python library for data analysis and statistics. *Python for high performance and scientific computing*, 14(9):1–9, 2011.

[49] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[50] Matias Mendieta, Taojiannan Yang, Pu Wang, Minwoo Lee, Zhengming Ding, and Chen Chen. Local learning matters: Rethinking data heterogeneity in federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8397–8406, 2022.

[51] Alessio Mora, Davide Fantini, and Paolo Bellavista. Federated learning algorithms with heterogeneous data distributions: An empirical evaluation. In *2022 IEEE/ACM 7th Symposium on Edge Computing (SEC)*, pages 336–341. IEEE, 2022.

[52] Kristina Müller and Freimut Bodendorf. Cross-silo federated learning in enterprise networks with cooperative and competing actors. *Hum. Side Serv. Eng*, 108:244–253, 2023.

[53] Fatma Murat, Ozal Yildirim, Muhammed Talo, Ulas Baran Baloglu, Yakup Demir, and U. Rajendra Acharya. Application of deep learning techniques for heartbeats detection using ecg signals-analysis and review. *Computers in Biology and Medicine*, 120:103726, 2020.

[54] Frank Nielsen. On the jensen–shannon symmetrization of distances relying on abstract means. *Entropy*, 21(5):485, 2019.

[55] Evgenija S Novikova, Yang Chen, and Aleksej V Meleshko. Evaluation of data heterogeneity in fl environment. In *2024 XXVII International Conference on Soft Computing and Measurements (SCM)*, pages 344–347. IEEE, 2024.

[56] Jean Ogier du Terrail, Samy-Safwan Ayed, Edwige Cyffers, Felix Grimberg, Chaoyang He, Regis Loeb, Paul Mangold, Tanguy Marchand, Othmane Marfoq, Erum Mushtaq, et al. Flamby: Datasets and benchmarks for cross-silo federated learning in realistic healthcare settings. *Advances in Neural Information Processing Systems*, 35:5315–5334, 2022.

[57] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[58] Debidutta Pattnaik, Sougata Ray, and Raghu Raman. Applications of artificial intelligence and machine learning in the financial services industry: A bibliometric review. *Heliyon*, 2024.

[59] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[60] Jiaming Pei, Wenxuan Liu, Jinhai Li, Lukun Wang, and Chao Liu. A review of federated learning methods in heterogeneous scenarios. *IEEE Transactions on Consumer Electronics*, 2024.

[61] Plotly Technologies Inc. Collaborative data science with plotly. `https://plotly.com`, 2015.

[62] Anichur Rahman, Tanoy Debnath, Dipanjali Kundu, Md Saikat Islam Khan, Airin Afroj Aishi, Sadia Sazzad, Mohammad Sayduzzaman, and Shahab S Band. Machine learning and deep learning-based approach in smart healthcare: Recent advances, applications, challenges and opportunities. *AIMS Public Health*, 11(1):58–109, 2024.

[63] Maryam Saeed, Olev Märtens, Benoit Larras, Antoine Frappé, Deepu John, and Barry Cardiff. Ecg classification with event-driven sampling. *IEEE Access*, 2024.

[64] Sudipan Saha and Tahir Ahmad. Federated transfer learning: Concept and applications. *Intelligenza Artificiale*, 15(1):35–44, 2021.

[65] Sadman Sakib, Mostafa M Fouda, Zubair Md Fadlullah, Khalid Abualsaud, Elias Yaacoub, and Mohsen Guizani. Asynchronous federated learning-based ecg analysis for arrhythmia detection. In *2021 IEEE International Mediterranean Conference on Communications and Networking (MeditCom)*, pages 277–282. IEEE, 2021.

[66] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[67] Xiuyu Shen, Jingxu Chen, Siying Zhu, and Ran Yan. A decentralized federated learning-based spatial–temporal model for freight traffic speed forecasting. *Expert Systems with Applications*, 238:122302, 2024.

[68] Bassel Soudan, Sohail Abbas, Ahmed Kubba, Omnia Abu Waraga, Manar Abu Talib, and Qassim Nasir. Scalability and performance evaluation of federated learning frameworks: a comparative analysis. *International Journal of Machine Learning and Cybernetics*, pages 1–15, 2025.

[69] Alexandra Swanson, Margaret Kosmala, Chris Lintott, Robert Simpson, Arfon Smith, and Craig Packer. Snapshot serengeti, high-frequency annotated camera trap images of 40 mammalian species in an african savanna. *Scientific data*, 2(1):1–14, 2015.

[70] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[71] Saeed Vahidian, Mahdi Morafah, Mubarak Shah, and Bill Lin. Rethinking data heterogeneity in federated learning: Introducing a new notion and standard benchmarks. *IEEE Transactions on Artificial Intelligence*, 2023.

[72] Yanmeng Wang, Qingjiang Shi, and Tsung-Hui Chang. Why batch normalization damage federated learning on non-iid data? *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[73] Yuanli Wang, Joel Wolfrath, Nikhil Sreekumar, Dhruv Kumar, and Abhishek Chandra. Accelerated training via device similarity in federated learning. In *Proceedings of the 4th International Workshop on Edge Systems, Analytics and Networking*, pages 31–36, 2021.

[74] Kok-Seng Wong, Manh Nguyen-Duc, Khiem Le-Huy, Long Ho-Tuan, Cuong Do-Danh, and Danh Le-Phuoc. An empirical study of federated learning on iot-edge devices: Resource allocation and heterogeneity. *arXiv preprint arXiv:2305.19831*, 2023.

[75] Feijie Wu, Song Guo, Zhihao Qu, Shiqi He, Ziming Liu, and Jing Gao. Anchor sampling for federated learning with partial client participation. In *International Conference on Machine Learning*, pages 37379–37416. PMLR, 2023.

[76] Hongda Wu and Ping Wang. Fast-convergent federated learning with adaptive weighting. *IEEE Transactions on Cognitive Communications and Networking*, 7(4):1078–1088, 2021.

[77] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[78] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

[79] Xin Yang, Hao Yu, Xin Gao, Hao Wang, Junbo Zhang, and Tianrui Li. Federated continual learning via knowledge fusion: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 36(8):3832–3850, 2024.

[80] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International conference on machine learning*, pages 5650–5659. Pmlr, 2018.

[81] Hao Yu, Xin Yang, Xin Gao, Yihui Feng, Hao Wang, Yan Kang, and Tianrui Li. Overcoming spatial-temporal catastrophic forgetting for federated class-incremental learning. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 5280–5288, 2024.

[82] Hao Yu, Xin Yang, Le Zhang, Hanlin Gu, Tianrui Li, Lixin Fan, and Qiang Yang. Addressing spatial-temporal data heterogeneity in federated continual learning via tail anchor. *arXiv preprint arXiv:2412.18355*, 2024.

[83] Dun Zeng, Siqi Liang, Xiangjing Hu, Hui Wang, and Zenglin Xu. Fedlab: A flexible federated learning framework. *Journal of Machine Learning Research*, 24(100):1–7, 2023.

[84] Chen Zhang, Yu Xie, Hang Bai, Bin Yu, Weihong Li, and Yuan Gao. A survey on federated learning. *Knowledge-Based Systems*, 216:106775, 2021.

[85] Mufeng Zhang, Yining Wang, and Tao Luo. Federated learning for arrhythmia detection of non-iid ecg. In *2020 IEEE 6th International Conference on Computer and Communications (ICCC)*, pages 1176–1180. IEEE, 2020.

[86] Peng Zhang and Maged N Kamel Boulos. Privacy-by-design environments for large-scale health research and federated learning from data. *International Journal of Environmental Research and Public Health*, 19(19):11876, 2022.

[87] Shuyao Zhang, Jordan Tay, and Pedro Baiz. The effects of data imbalance under a federated learning approach for credit risk forecasting. *arXiv preprint arXiv:2401.07234*, 2024.

[88] Xuehan Zhou, Ruimin Ke, Zhiyong Cui, Qiang Liu, and Wenxing Qian. Stfl: Spatio-temporal federated learning for vehicle trajectory prediction. In *2022 IEEE 2nd International Conference on Digital Twins and Parallel Intelligence (DTPI)*, pages 1–6. IEEE, 2022.

[89] Hangyu Zhu, Jinjin Xu, Shiqing Liu, and Yaochu Jin. Federated learning on non-iid data: A survey. *Neurocomputing*, 465:371–390, 2021.