Robust Federated Learning Framework for Defending Against Malicious Attacks

Ebtisaam Alharbi, BSc (Hons), MSc

School of Computing and Communications

Lancaster University

A thesis submitted for the degree of

Doctor of Philosophy

August, 2025

Robust Federated Learning Framework for Defending Against Malicious Attacks

Ebtisaam Alharbi, BSc (Hons), MSc.

School of Computing and Communications, Lancaster University

A thesis submitted for the degree of *Doctor of Philosophy*. .

Abstract

Federated Learning (FL) has emerged as a decentralized machine learning paradigm that enables collaborative model training while preserving data privacy. However, its reliance on distributed and unverified client updates makes it highly vulnerable to adversarial attacks such as data poisoning, model poisoning, and backdoor attacks. These threats can degrade performance, compromise integrity, and introduce hidden malicious behaviors, raising serious concerns for FL deployment in safety-critical domains such as healthcare, finance, and IoT. Addressing these challenges requires defense mechanisms that are both effective and privacy-preserving.

This thesis presents three novel defense frameworks that enhance the security and reliability of FL. First, we propose Robust Federated Clustering (RFCL), a multi-centre clustering-based aggregation strategy that groups client models by similarity to filter out adversarial updates. RFCL improves resilience to poisoning attacks under highly Non-IID (Non-independent and identically distributed) settings by isolating malicious updates while retaining benign diversity.

Second, we introduce Robust Knowledge Distillation (RKD) to mitigate backdoor threats. RKD integrates unsupervised clustering, median model selection, and knowledge distillation to suppress compromised client updates during global aggregation. This approach enables robust learning without requiring access to labeled reference data.

Third, we develop Synthetic Data-Driven Conformity Scoring for FL (SD-CSFL), an anomaly detection framework that uses synthetic calibration data, entropy-based nonconformity scoring, and adaptive thresholds to detect gradient manipulation and stealthy backdoors. SD-CSFL operates without accessing client data and remains effective in heterogeneous and adaptive attack scenarios.

The proposed methods are evaluated on diverse FL benchmarks—MNIST, Fashion-MNIST, EMNIST, CIFAR-10, and Birds—across a broad spectrum of adversarial settings. Results demonstrate that RFCL, RKD, and SD-CSFL consistently outperform existing defenses, significantly improving FL robustness while preserving model performance and data privacy.

Acknowledgements

I would like to express my sincere gratitude to my home country, Saudi Arabia, whose unwavering support and vision have been instrumental in bringing this work to fruition. This thesis is the result of the collective encouragement and assistance of many individuals and institutions, both at home and abroad, whose support has made this journey possible.

I am profoundly grateful to Dr. Leandro Soriano Marcolino for his consistent guidance, generous support, and enduring patience. His mentorship transformed challenges into opportunities for growth, and I feel incredibly fortunate to have benefited from his wisdom. I also extend my sincere thanks to Professor Qiang Ni and Dr. Antonios Gouglidis for their insightful advice and contributions.

I gratefully acknowledge the scholarship support from Umm Al-Qura University and the essential computing resources provided by the High-End Computing facility at Lancaster University. My appreciation also goes to the anonymous reviewers, whose constructive feedback significantly enhanced the quality of my work.

Finally, I owe a special debt of gratitude to my family. I am deeply thankful to my parents for their unwavering love and support, and to my sister Lamaa for looking after my daughter and son during my travels and conference commitments. A heartfelt thanks go to my loving husband, Sharaf, and our cherished children, Rafaa and Hashim, who have been my strength throughout this journey.

Declaration

I declare that the work presented in this thesis is, to the best of my knowledge and belief, original and my own work. The material has not been submitted, either in whole or in part, for a degree at this, or any other university. This thesis does not exceed the maximum permitted word length of 80,000 words including appendices and footnotes, but excluding the bibliography. A rough estimate of the word count is: 27369

Ebtisaam Alharbi

Publications

The following publications were generated as part of the research conducted for this thesis. These works have significantly contributed to the development of the thesis, guiding its direction and supporting its findings throughout my PhD:

- Alharbi, Ebtisaam and Marcolino, Leandro Soriano and Gouglidis, Antonios and Ni, Qiang. "Robust Federated Learning Method Against Data and Model Poisoning Attacks with Heterogeneous Data Distribution". In: *ECAI 2023*. IOS Press, 2023, pp. 85–92

- Alharbi, Ebtisaam and Marcolino, Leandro Soriano and Gouglidis, Antonios and Ni, Qiang. "Robust Knowledge Distillation in Federated Learning: Counteracting Backdoor Attacks". In: *SaTML 2025*. IEEE, 2025

- Alharbi, Ebtisaam and Kerim, Abdulrahman and Marcolino, Leandro Soriano and Ni, Qiang. "Synthetic Data-Driven Federated Learning Defense Against Gradient Manipulation and Backdoor Attacks". In: *ICLR 2026*. Under-review. 2026

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Federated Learning (FL) has emerged as a transformative paradigm in decentralized machine learning, enabling multiple clients to collaboratively train a shared global model without centralizing their private and sensitive data [46]. In FL, each client keeps its data locally and trains a model on this data, transmitting only model updates (e.g., parameter weights or gradients) to a central server. This server then aggregates these updates to build a global model. This setup allows FL to address key challenges in data privacy, communication efficiency, and scalability [73]. FL minimizes privacy risks associated with centralized data collection by keeping data on clients' devices, ensuring compliance with privacy regulations and data security requirements.

This approach also minimizes data transmission costs, as only model updates are sent to the server, which is particularly advantageous in bandwidth-constrained environments, such as mobile or IoT networks [47]. The reduction in data movement conserves network resources and minimizes latency, making FL suitable for time-sensitive applications. Furthermore, FL supports scalability across a large number of distributed devices, each contributing insights from its unique local data. This decentralized model can capture a wide range of data characteristics, resulting in a more robust and adaptive model that reflects diverse data sources. Due to these benefits, FL has found extensive applications in domains where data confidentiality is crucial, such as healthcare [70], finance [40], and mobile applications [36].

Despite these advantages, the decentralized nature of FL introduces significant challenges that impact the security, robustness, and reliability of the global model. Figure 1.1 provides an overview of the federated learning framework, highlighting how benign and malicious clients both contribute updates that influence the global model during aggregation. This illustration also shows how adversarial attacks, such as data and model poisoning, exploit FL's decentralized nature to compromise the model's integrity. FL's reliance on unverified client contributions leaves it particularly vulnerable to these threats [47]. Unlike centralized learning systems, where data and updates can be carefully monitored, FL assumes that each client behaves honestly, making it susceptible to malicious clients who may attempt to compromise the model. Adversarial attacks in FL are generally categorized into data poisoning and model poisoning attacks, each with distinct methods and impacts.

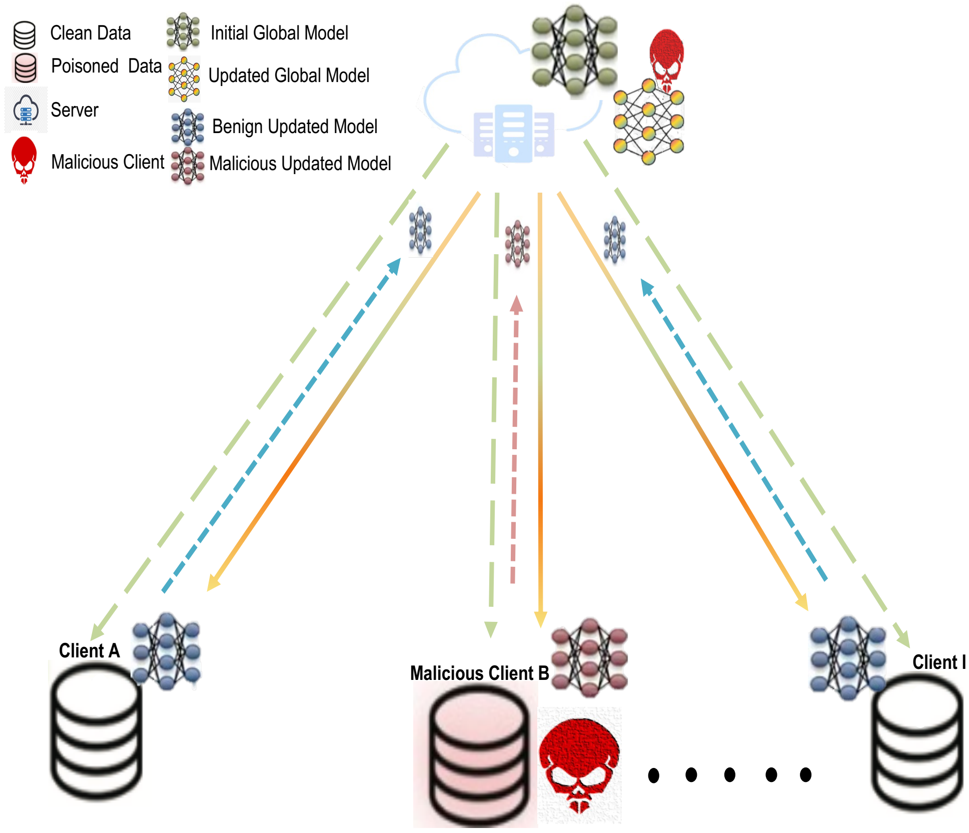


Figure 1.1: Overview of Vulnerabilities in Federated Learning.

Data poisoning attacks involve manipulating the local dataset used by a client to introduce harmful patterns or biases that degrade the global model's performance. A common type of data poisoning is the label flipping attack [9], where an adversarial client intentionally mislabels certain data points. For example, a client might change the labels of "cat" images to "dog." When these poisoned updates are aggregated, they lead to systematic misclassifications in the global model, reducing accuracy or inducing specific biases.

Model poisoning attacks go beyond data manipulation by directly altering the model parameters or gradients that clients send to the server. In these attacks, malicious clients craft updates that steer the global model's behavior in a harmful direction without changing the underlying data [7]. A common approach is to add random noise or subtle perturbations to model parameters, which can degrade the model's performance [2].

A particularly stealthy type of attack, backdoor attacks, can manifest as either data poisoning or model poisoning. In a data poisoning backdoor attack, the attacker introduces a specific "trigger" pattern within their dataset and associates it with an incorrect label. For instance, in an image classification task, a small, unique pattern might be added to "cat" images and labeled as "dog". As a result, the global model learns this association, leading to misclassifications whenever the trigger pattern is present [5]. In contrast, in a model poisoning backdoor attack, the attacker directly encodes the backdoor functionality into the model parameters. This approach allows the backdoor to remain hidden during standard validation, activating only when the specific trigger pattern appears [76]. This dual approach makes backdoor attacks particularly difficult to detect, as the global model behaves normally on clean inputs but exhibits malicious behavior under specific conditions defined by the trigger.

The difficulty in detecting these adversarial attacks is further compounded by the Non-Independent and Identically Distributed (Non-IID) nature of client data in FL [56]. In real-world applications, each client's data distribution often reflects unique demographic, geographic, or usage-based characteristics, introducing natural variability in their updates. This Non-IID characteristic exacerbates the challenge of distinguishing between legitimate variability and adversarial manipulations [27]. Current defense mechanisms in FL are often limited by their reliance on assumptions about data distribution and client behavior, resulting in significant challenges in handling complex adversarial strategies and Non-IID data. Techniques like Krum [11] or Median [74] aggregation reduce the impact of outliers but are less effective against sophisticated attacks, such as coordinated backdoor attacks or model poisoning strategies that closely resemble benign data variations. Furthermore, these defenses frequently require trade-offs between security and model performance; overly conservative defenses may exclude valuable client updates, reducing the model's ability to generalize, while lenient approaches may aggregate adversarial updates, compromising the integrity of the global model.

In summary, this thesis addresses the critical security and robustness challenges inherent in Federated Learning, with a particular focus on adversarial threats such as data poisoning, model poisoning, and backdoor attacks. These attacks exploit the decentralized architecture and Non-IID data environment of FL, introducing vulnerabilities that can severely compromise the integrity and reliability of the global model. This research aims to design, develop, and rigorously evaluate robust and adaptive defense frameworks capable of countering these adversarial threats while preserving the model's performance and generalization across diverse client data distributions. By proposing novel aggregation methods techniques, this thesis contributes to enhancing the security and resilience of FL, advancing its suitability for deployment in privacy-sensitive domains.

1.1 Research Questions and Contributions

The research questions aim to develop robust and adaptive defense mechanisms that enhance FL's resilience to adversarial attacks, including data poisoning, model poisoning, and backdoor attacks, even under Non-IID data conditions.

1.1.1 Defending against Data and Model Poisoning Attacks

Q1) How can FL defend against data and model poisoning attacks while maintaining robustness in Non-IID data environments?

Federated learning is vulnerable to data and model poisoning attacks, particularly in Non-IID settings where malicious updates can significantly degrade the global model's performance. Existing robust aggregation methods struggle in heterogeneous environments, as adversaries can exploit variations in client data distributions to inject harmful updates. To address these challenges, this work introduces **Robust Federated Clustering (RFCL)**, an innovative aggregation technique designed to enhance FL security against data and model poisoning attacks. RFCL employs clustering and cosine similarity to group client models based on similarity, forming high-quality clusters that represent groups of reliable client updates. The aggregation process prioritizes these clusters, reducing the impact of adversarial clients that attempt to poison the global model.

A key feature of RFCL is meta-learning phase, which consolidates models within each selected cluster. This ensures that the global model benefits from diverse yet trustworthy client updates, improving robustness in Non-IID environments. Additionally, RFCL integrates a personalization mechanism, selectively updating models for clients within similar clusters, allowing benign clients to receive tailored updates while excluding adversarial contributions. This approach strengthens FL's resistance to adversarial manipulations in Non-IID settings.

The contributions of RFCL are summarized as follows:

- **Clustering-based robust aggregation**: RFCL employs hierarchical clustering with cosine similarity to detect and exclude unreliable client updates, mitigating data and model poisoning attacks.

- **Meta-learning-driven model consolidation**: By integrating a meta-learning phase, RFCL improves global model robustness and ensures high-quality aggregation.

- **Personalized updates for benign clients**: RFCL's selective update mechanism enhances FL security while maintaining performance for benign clients, even in heterogeneous settings.

- **Comprehensive experimental validation**: Extensive evaluations across various attack scenarios—including Inner Product Masking (IPM), A Little Is Enough (ALIE), sign-flipping, random noise, and label-flipping—demonstrate that RFCL consistently outperforms state-of-the-art robust aggregation methods, maintaining high model integrity even in the presence of large numbers of malicious clients.

- **RFCL Implementation Repository:** The full implementation of the RFCL framework is available on GitHub at https://github.com/EbtisaamCS/RFCL.

The proposed RFCL framework was published in the proceedings of the European Conference on Artificial Intelligence (ECAI 2023) [2].

1.1.2 Countering Backdoor Attacks

Q2) What defense mechanism can be developed to counter backdoor attacks in FL, especially under Non-IID conditions?

Backdoor attacks in FL pose a significant threat, as malicious clients introduce hidden triggers into their model updates to manipulate predictions while remaining undetected. Traditional defense mechanisms rely on strong assumptions about data distribution and attack strategies, making them less effective in real-world Non-IID environments. To address these challenges, this thesis proposes **Robust Knowledge Distillation (RKD)**, a novel approach that enhances FL integrity by filtering malicious updates through clustering and model selection.

RKD employs clustering algorithms and cosine similarity to detect groups of benign client updates, isolating potential backdoor-injected updates as outliers. This ensures that only models with consistent, trustworthy updates contribute to the global model. Within the benign clusters, RKD refines model selection by choosing updates closest to the median, forming a reliable ensemble of client models. Knowledge distillation is then applied to transfer insights from this ensemble to the global model, ensuring that only benign update behavior is incorporated.

The key contributions of RKD are as follows:

- **Clustering-based malicious update isolation**: RKD effectively detects and removes backdoor-injected updates using hierarchical clustering and cosine similarity.

- **Median-based model selection**: By selecting updates closest to the median within benign clusters, RKD further minimizes the influence of potential adversarial outliers.

- **Knowledge distillation for secure aggregation**: RKD leverages knowledge distillation to ensure only benign model behaviors are retained, preventing backdoor propagation.

- **Extensive empirical validation**: RKD has been rigorously evaluated on diverse datasets, including CIFAR-10, EMNIST, and Fashion-MNIST, and tested against advanced backdoor threats—such as Adversarially Adaptive Backdoor Attacks (A3FL), Focused-Flip Federated Backdoor Attacks (F3BA), and Distributed Backdoor Attacks (DBA). It consistently reduces attack success rates to below **17%** while maintaining model accuracy above **80%**, outperforming existing defenses.

- **RKD Implementation Repository:** The full implementation of the RKD framework is available on GitHub at https://github.com/EbtisaamCS/RKD.

The proposed RKD framework was published in the proceedings of the 3rd IEEE Conference on Secure and Trustworthy Machine Learning (SaTML 25) [3].

1.1.3 Countering Various Adversarial Threats

Q3) How can a comprehensive framework be developed to effectively defend against multiple adversarial attack types, including model poisoning and backdoor attacks, while ensuring robustness in Non-IID data settings?

Federated learning faces diverse adversarial threats, ranging from gradient manipulation to backdoor attacks, requiring a unified defense strategy that handles both types of attacks while maintaining high model utility. To address this challenge, this thesis introduces **Synthetic Data-Driven Conformity Scoring for Federated Learning (SD-CSFL)**, a novel framework that utilizes synthetic data to compute entropy-based nonconformity scores for detecting malicious updates.

SD-CSFL addresses the limitations of traditional defense methods by using synthetic datasets rather than relying on potentially compromised client data. This allows for a consistent and controlled evaluation of model updates. The framework computes entropy-based nonconformity scores, measuring deviations in client updates from expected behavior. Additionally, SD-CSFL employs adaptive percentile thresholding, dynamically adjusting detection thresholds based on evolving client behaviors, and stratified sampling, which ensures balanced calibration sets, enhancing detection accuracy across diverse classes in Non-IID settings.

The contributions of SD-CSFL are as follows:

- **Synthetic data-based evaluation**: By leveraging synthetic datasets, SD-CSFL ensures privacy preservation while providing a stable reference for client update evaluation.

- **Entropy-based nonconformity scoring**: SD-CSFL introduces a novel entropy-driven approach to detect adversarial updates, improving detection accuracy for both gradient manipulation and backdoor attacks.

- **Adaptive thresholding for dynamic detection**: The framework dynamically adjusts thresholds based on percentile-based calibration, ensuring adaptability to evolving adversarial strategies.

- **Stratified sampling for balanced evaluation**: SD-CSFL incorporates stratified sampling techniques to create balanced calibration sets, enhancing its effectiveness across different data distributions.

- **Comprehensive experimental validation**: Evaluations on CIFAR-10 and Birds-525 [31], a large-scale dataset containing images from 525 bird species, demonstrate that SD-CSFL achieves a 35% improvement in detection accuracy for gradient manipulation attacks and an 80% reduction in backdoor attack success rate, while maintaining 61% accuracy under highly poisoned Non-IID conditions, outperforming existing defense methods.

- **SD-CSFL Implementation Repository:** The full implementation of the SD-CSFL framework is available on GitHub at https://github.com/EbtisaamCS/SD-CSFL.

The proposed SD-CSFL framework is currently under-review .

Together, these contributions represent a significant advancement in FL security and robustness. Each framework—RFCL, RKD, and SD-CSFL—addresses different aspects of adversarial threats. RFCL is designed to counter data and model poisoning in Non-IID settings, and RKD specifically tackles backdoor attacks by filtering out infected models. SD-CSFL, on the other hand, defends against a broader range of adversarial scenarios, including both poisoning and stealthy backdoor triggers. By integrating clustering, knowledge distillation, synthetic data, and conformal prediction, this thesis lays a foundation for FL systems that are resilient, adaptable, and well-suited for deployment in privacy-sensitive environments where data integrity is critical.

1.2 Thesis Outline

The structure of this thesis and its related contributions are depicted in Figure 1.2. This thesis comprises six chapters, each focusing on different aspects of securing federated learning against adversarial threats.

Chapter 2 introduces the background and related work essential for understanding the challenges and methodologies discussed in this thesis. It provides an overview of the FL architecture, the adversarial attack types (data poisoning, model poisoning, and backdoor attacks), and existing defense mechanisms. The chapter also reviews the limitations of current approaches, particularly in handling Non-IID data and sophisticated attack strategies.

In Chapter 3, we discuss our first contribution, Robust Federated Clustering (RFCL) [2], as presented in the proceedings of ECAI 23. This framework is designed to defend against data and model poisoning attacks in FL. By employing advanced clustering techniques, such as HDBSCAN, and leveraging cosine similarity, RFCL identifies and aggregates trustworthy client updates. This chapter details how RFCL uses meta-learning to enhance model personalization and robustness. Extensive experiments demonstrate RFCL's effectiveness against diverse attacks, including ALIE, IPM, and label-flipping, across datasets such as MNIST, CIFAR-10, and Fashion-MNIST.

In Chapter 4, we introduce our second contribution, Robust Knowledge Distillation (RKD) [3], as published in the proceedings of SaTML 2025, which focuses on countering backdoor attacks in Federated Learning. RKD combines clustering with median-based selection to isolate malicious updates and employs knowledge distillation to aggregate benign client models into a robust global model. This chapter includes evaluations against sophisticated backdoor attack methods such as A3FL, F3BA, and DBA, showcasing RKD's ability to reduce attack success rates while maintaining high accuracy, even in Non-IID environments.

In Chapter 5, we present our third contribution, Synthetic Data-Driven Conformity Scoring for Federated Learning (SD-CSFL), which is currently under-review. This unified framework defends against both gradient manipulation and backdoor attacks by leveraging synthetic calibration datasets to compute entropy-based nonconformity scores. Adaptive thresholding and stratified sampling are introduced to enhance detection accuracy in Non-IID conditions. Experiments on CIFAR-10 and Birds demonstrate SD-CSFL's superior performance in detecting and mitigating adversarial behaviors.

In Chapter 6, we provide our concluding thoughts and summarize the key findings of this thesis. We reflect on the significance of RFCL, RKD, and SD-CSFL in enhancing the security and robustness of FL, emphasizing their effectiveness against diverse adversarial threats. Additionally, we explore directions for future work, including ways to improve the computational efficiency of FL security mechanisms and ensure that privacy-preserving protocols remain robust at large scales.

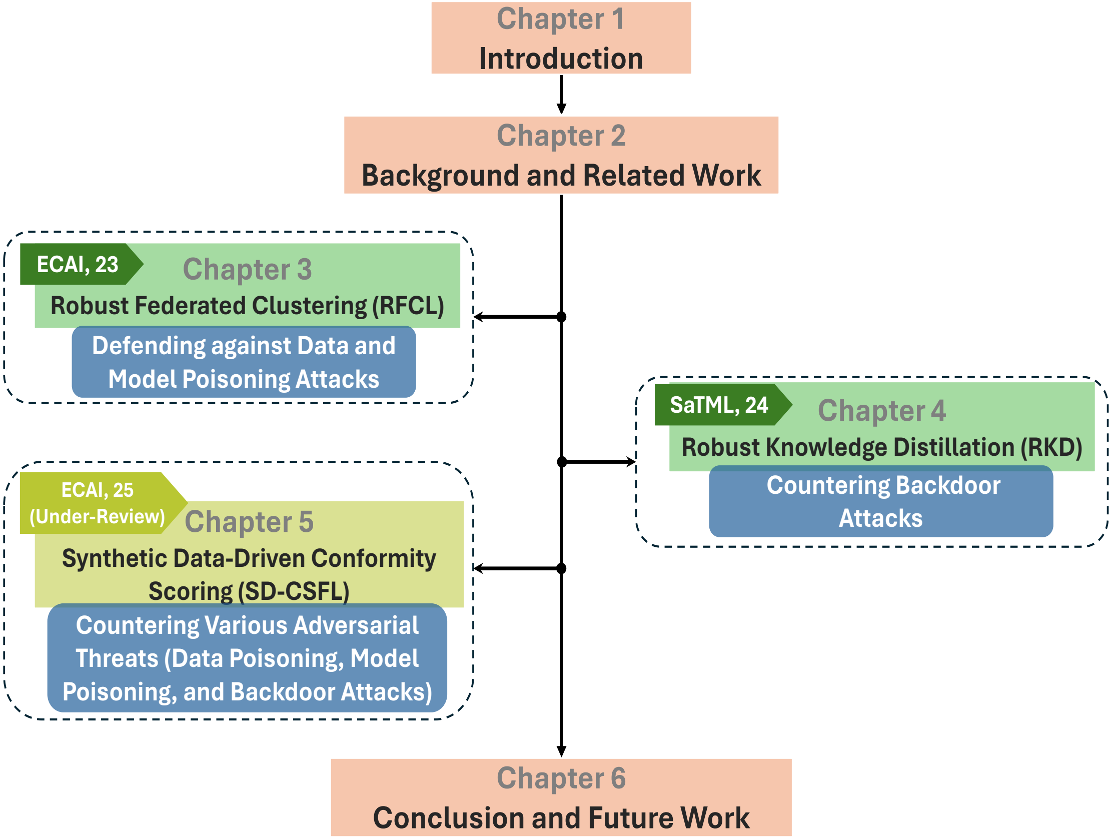

Chapter 1
Introduction
Chapter 2
Background and Related Work
ECAI, 23
Chapter 3
Robust Federated Clustering (RFCL)
Defending against Data and Model Poisoning Attacks
SaTML, 24
Chapter 4
Robust Knowledge Distillation (RKD)
Countering Backdoor Attacks
ECAI, 25 (Under-Review)
Chapter 5
Synthetic Data-Driven Conformity Scoring (SD-CSFL)
Countering Various Adversarial Threats (Data Poisoning, Model Poisoning, and Backdoor Attacks)
Chapter 6
Conclusion and Future Work

Figure 1.2: Outline of the Thesis.

Chapter 2

Background and Related Work

In this chapter, we provide the necessary background to ensure a clear understanding of the foundational concepts and challenges addressed in this thesis. Additionally, we review related work to highlight existing research and establish the context for the proposed frameworks and methodologies.

We begin by introducing the federated learning process in Section 2.1, where we outline its decentralized nature, collaborative training mechanism, and key advantages over traditional machine learning approaches. This section also delves into the iterative steps of FL, supported by mathematical formulations and illustrative diagrams.

Next, in Section 2.2, we explore the architectures and paradigms of FL, categorizing them into centralized architecture (Section 2.2.1) and decentralized architecture (Section 2.2.2). We further discuss their respective operational paradigms, including Cross-Silo Federated Learning and Cross-Device Federated Learning, while highlighting their use cases, benefits, and limitations. The section concludes with a comparison of these architectures to provide a clear understanding of their unique roles in FL.

Section 2.3 focuses on data composition in FL, emphasizing the challenges posed by non-IID data distributions across clients. Here, we classify FL into Horizontal Federated Learning (Section 2.3.1), Vertical Federated Learning (Section 2.3.2), and

Federated Transfer Learning (Section 2.3.3), explaining their specific applications and challenges. Additionally, we address the impact of data heterogeneity and the methods used to mitigate its effects on model training.

In Section 2.4, we address the security threats in FL, which include poisoning attacks and backdoor attacks. This section discusses how these adversarial strategies exploit the decentralized nature of FL to compromise model integrity and reliability. We provide a detailed review of data poisoning and model poisoning (Section 2.4.1) and elaborate on the intricacies of backdoor attacks (Section 2.4.2), highlighting their stealth and targeted nature.

Finally, Section 2.5 provides a related work of defense mechanisms in FL. This includes robust aggregation techniques (Section 2.5.1) and backdoor-specific defenses (Section 2.5.2). Each subsection explores the mechanisms, strengths, and limitations of various approaches, offering insights into their applicability in diverse FL scenarios.

2.1 Federated Learning Process

FL is a distributed machine learning paradigm that enables collaborative model training across multiple clients while ensuring that raw data remains localized. This decentralized approach not only preserves privacy but also leverages the computational capabilities of distributed clients to construct a global model [46]. By maintaining data privacy and facilitating collaborative learning, FL addresses many challenges associated with centralized machine learning systems.

Mathematically, FL is formulated as an optimization problem that aims to minimize a global loss function $F(\mathbf{w})$, defined as the weighted sum of local loss functions across C participating clients [46]. The global loss function is given by:

$$F(\mathbf{w}) = \sum_{i=1}^{C} \frac{n_i}{N} f_i(\mathbf{w}), \tag{2.1}$$

where $\mathbf{w}$ represents the global model parameters, $f_i(\mathbf{w})$ is the local loss function for client i, n_i denotes the number of data samples held by client i, and $N = \sum_{i=1}^{C} n_i$ is the total number of data samples across all clients.

This formulation ensures that each client's contribution to the global model update is weighted proportionally to its dataset size, making FL particularly useful for heterogeneous (Non-IID) data distributions.

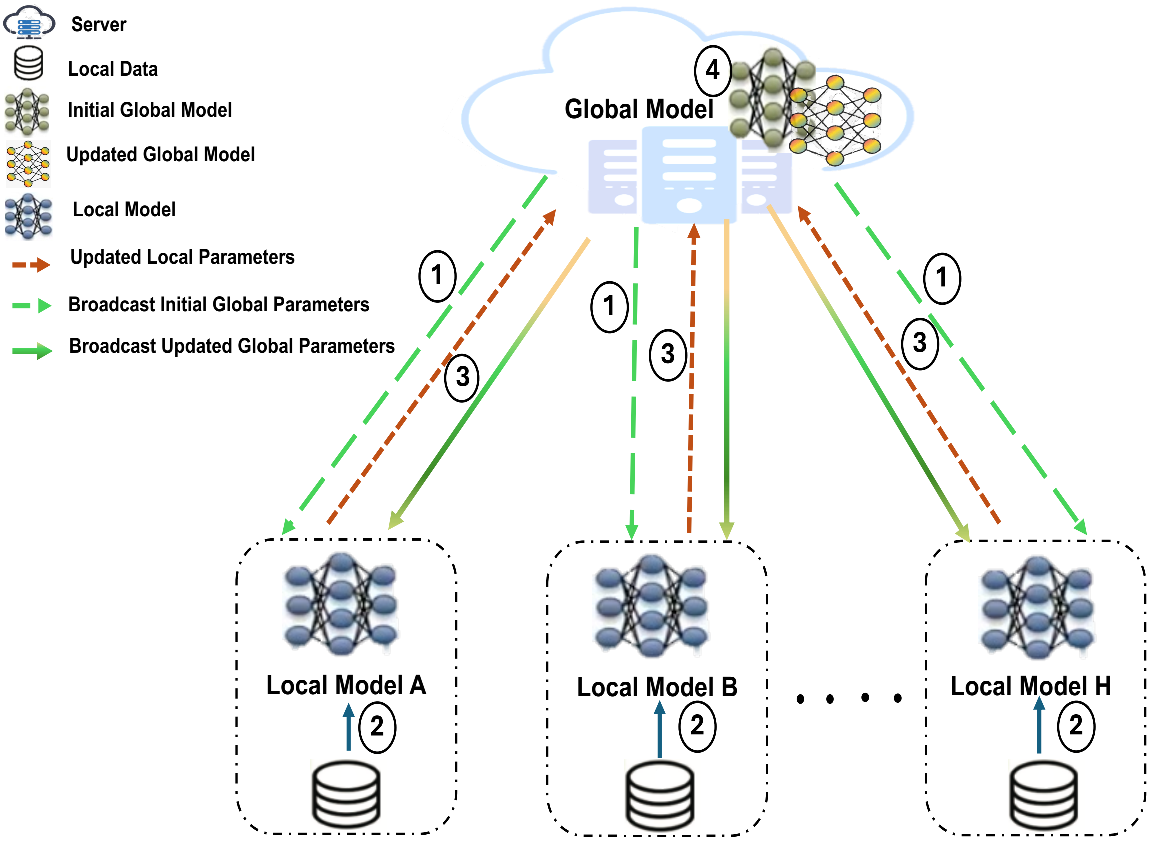


Figure 2.1: Federated learning process illustrating the iterative workflow: (1) Initialization and broadcast of the global model by the central server, (2) Local training at individual clients using private datasets, (3) Transmission of updated parameters from clients to the server, and (4) Aggregation of updates by the server to refine and redistribute the global model.

The process of FL is iterative and consists of the following steps [73], as illustrated in Figure 2.1:

1. **Broadcast of Global Model:** The central server initializes the global model

parameters $\mathbf{w}^{(0)}$ and broadcasts them to all participating clients. This ensures that all clients begin with the same model architecture and initial parameters, enabling synchronized training across the system.

2. **Local Training:** Each client trains its local model using its private dataset. The clients refine the global model parameters by minimizing their local objective $f_i(\mathbf{w})$ through gradient descent, as shown in Equation 2.2:

$$\mathbf{w}_i^{(t+1)} = \mathbf{w}^{(t)} - \eta \nabla f_i(\mathbf{w}^{(t)}), \tag{2.2}$$

   where η is the learning rate and t denotes the iteration index. This step captures the unique data characteristics of each client while preserving data privacy.

3. **Parameter Sharing:** After completing local training, clients send their updated model parameters or gradients to the central server. Only these updates are shared, ensuring that raw data remains securely stored on the clients. This step significantly reduces privacy risks and complies with data protection regulations.

4. **Global Aggregation:** The central server aggregates the updates received from all clients to refine the global model. The aggregation typically employs a weighted averaging strategy [46], as shown in Equation 2.3:

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \sum_{i=1}^{C} \frac{n_i}{N} \nabla f_i(\mathbf{w}^{(t)}), \tag{2.3}$$

   where the weights $\frac{n_i}{N}$ ensure that updates from clients with larger datasets have a proportionally greater impact on the global model. The refined model is then redistributed to the clients, completing one iteration of the FL process.

These four steps are repeated iteratively until the global loss function $F(\mathbf{w})$ converges to a desired threshold. This iterative process enables the global model to learn collaboratively from diverse and distributed datasets.

The iterative nature of FL facilitates robust learning despite data heterogeneity, communication constraints, and limited computational resources. The convergence of the global model depends on factors such as the diversity of client datasets, the efficiency of communication protocols, and the optimization techniques employed [56].

FL offers significant advantages over traditional machine learning methods. Its decentralized structure allows for efficient scaling, enabling the integration of numerous clients and large datasets. Additionally, the diverse data distributions across clients enhance the generalizability and robustness of the trained global model, making FL particularly suitable for applications in privacy-sensitive domains such as healthcare, finance, and IoT [47].

2.2 Federated Learning Architectures and Paradigms

FL is a transformative approach to machine learning that allows models to be trained across decentralized data sources while preserving data privacy. The architecture of FL dictates how participants, or clients, collaborate to build a global model. Two primary architectural types exist within FL: Centralized Federated Learning (CFL) and Decentralized Federated Learning (DFL) [73]. These architectures are further distinguished by two paradigms: Cross-Silo Federated Learning and Cross-Device Federated Learning, which address different scales and types of participants.

2.2.1 Centralized Federated Learning Aggregation

Centralized Federated Learning (CFL) aggregation is the most commonly implemented FL architecture. In CFL, a central server orchestrates the entire training process by distributing the initial global model to clients, collecting updates from them, and aggregating these updates to refine the model. The central server ensures synchronization across clients, making CFL particularly effective in managing large-scale collaborations. CFL is often used in scenarios where a reliable central server can be deployed to oversee the workflow [73].

2.2.1.1 Cross-Silo Federated Learning

Within CFL, Cross-Silo Federated Learning is a paradigm that involves collaboration among a relatively small number of clients, typically organizations or institutions referred to as silos. These silos, such as hospitals, banks, or universities, possess large and structured datasets that are used to train a shared model. The central server plays a pivotal role in coordinating this process, ensuring data privacy is maintained while leveraging the high-quality data from each silo. For example, hospitals can collaboratively train diagnostic models while adhering to strict data privacy regulations like banks can develop fraud detection models without sharing sensitive customer information [73].

2.2.1.2 Cross-Device Federated Learning

Another paradigm within CFL is Cross-Device Federated Learning, which operates at a much larger scale. Here, the central server manages millions of individual devices, such as smartphones, IoT devices, or wearables. Each device trains the global model on its local data, such as user interactions or sensor readings, and sends updates back to the server. These updates are aggregated to refine the global model. This paradigm is widely used for applications like predictive text input on mobile phones, where privacy and personalization are essential. Cross-Device CFL ensures that user data never leaves the device, making it a privacy-preserving solution for personalized applications [73].

2.2.2 Decentralized Federated Learning Aggregation

In contrast to CFL, Decentralized Federated Learning (DFL) aggregation eliminates the reliance on a central server. Instead, clients collaborate directly through peer-to-peer communication, sharing and aggregating model updates among themselves. DFL is particularly valuable in scenarios where a central server may introduce bottlenecks or security vulnerabilities, such as single-point failures [42]. This decentralized architecture is gaining popularity for its robustness and scalability in distributed environments [8].

2.2.2.1 Cross-Silo Federated Learning

Cross-Silo Decentralized Federated Learning adapts the DFL architecture for organizational collaborations. In this paradigm, silos communicate directly with each other to exchange and aggregate model updates, often using secure protocols. This approach allows organizations to train shared models collaboratively without relying on a central authority, offering enhanced privacy and security. For example, universities conducting joint research can train academic models while retaining full control over their individual datasets [8].

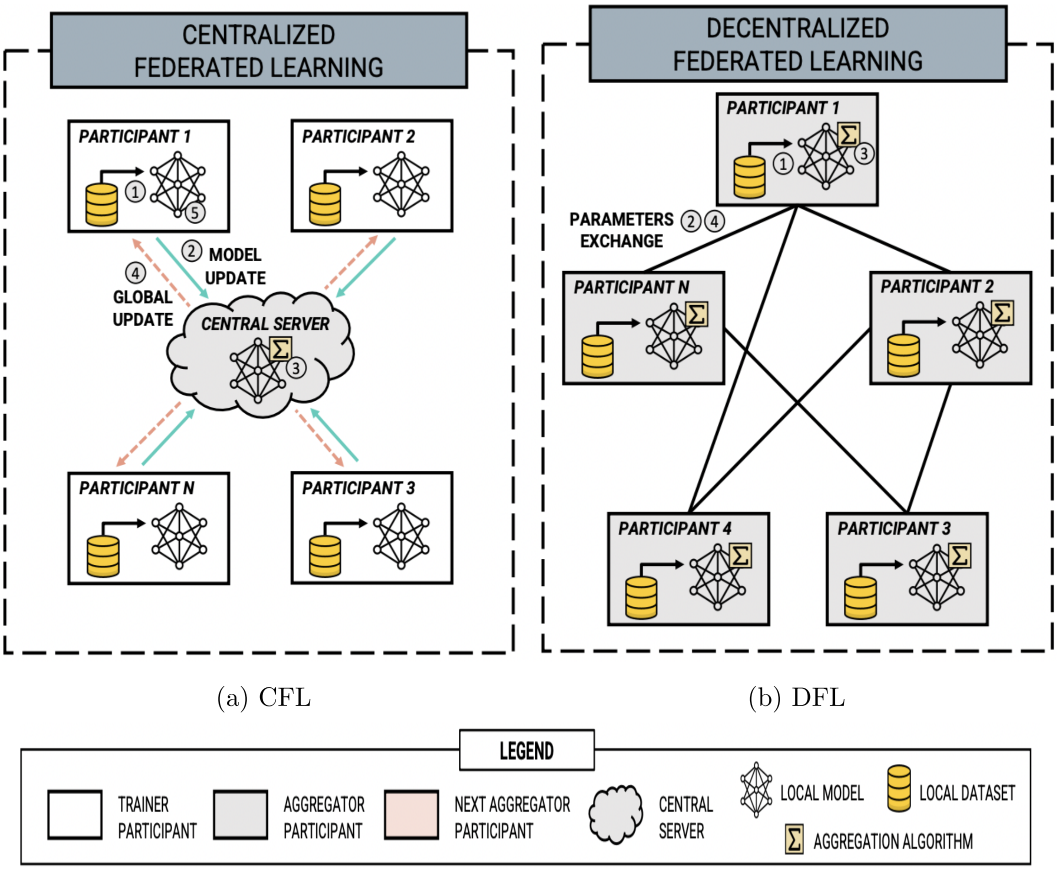
2.2.2.2 Cross-Device Federated Learning

Similarly, Cross-Device Decentralized Federated Learning applies DFL principles to large-scale networks of personal devices. Devices such as smart thermostats, wearables, or mobile phones collaborate to train a global model through direct communication. This paradigm is particularly useful in IoT networks, where devices interact locally to update shared models for tasks like energy management or network optimization. By avoiding central orchestration, Cross-Device DFL supports robust, flexible, and scalable learning in dynamic and resource-constrained environments [8].

Both CFL and DFL address specific challenges and opportunities in federated learning. CFL's reliance on a central server makes it suitable for structured collaborations in scenarios like healthcare, finance, and retail. On the other hand, DFL's decentralized nature enables autonomous and resilient learning, making it ideal for IoT and peer-to-peer collaborations.

The comparison between these two architectures is summarized in Table 2.1, detailing their core features, advantages, challenges, and use cases. Additionally, Figure 2.2 provides a visual representation of the CFL and DFL architectures, illustrating how centralized coordination contrasts with peer-to-peer collaboration.

Together, the table and figure offer a comprehensive overview of the distinctions between CFL and DFL, helping to contextualize their respective roles in federated learning systems.



(a) CFL

(b) DFL

Figure 2.2: Comparison of Centralized and Decentralized Federated Learning Architectures (from [8]).

Thesis Focus. Although both Centralized and Decentralized Federated Learning architectures offer distinct advantages, this thesis concentrates on the centralized, cross-device FL approach. Its reliance on a central server, straightforward orchestration, and large-scale device participation make CFL particularly suitable for the security challenges and solutions explored in the following chapters. Consequently, all proposed methods and experiments target adversarial threats within a centralized, cross-device FL setting.

Table 2.1: Comparison Between CFL and DFL

Feature	**Centralized Federated Learning (CFL)**	**Decentralized Federated Learning (DFL)**
Coordination	Orchestrated by a central server.	Peer-to-peer collaboration without a central server.
Reliability	Dependent on the central server; prone to single-point failures.	More robust; no single-point failure due to decentralized communication.
Scalability	Highly scalable for large-scale collaborations, especially in cross-device settings.	Limited by the efficiency of peer-to-peer communication and network structure.
Privacy	Central server aggregates updates, ensuring data privacy but may raise concerns about trust in the server.	Fully decentralized aggregation; no central entity, enhancing privacy and autonomy.
Communication	Clients communicate directly with the central server.	Clients communicate with peers, often requiring more complex communication protocols.
Resource Requirements	Relies on a central server with high computational power and stable connectivity.	No central server; relies on distributed client resources and robust communication.
Use Cases	Healthcare (hospitals), and finance (banks).	IoT networks, ad-hoc mobile networks, and autonomous distributed systems.
Challenges	Vulnerable to central server failures and bottlenecks.	Requires efficient peer-to-peer protocols; may face higher communication overhead.

2.3 Data Composition in Federated Learning

The composition of data in FL fundamentally influences how collaborative learning processes are designed and executed. FL is built on the principle of enabling distributed data owners, or clients, to collaboratively train a global machine learning model without sharing raw data. The diversity in data across clients introduces heterogeneity in terms of features, samples, and labels, which significantly affects the performance and robustness of FL systems [73]. Understanding data composition is crucial for tailoring FL frameworks to address these challenges effectively. Based on how data is distributed among the feature and sample spaces of participating clients, FL is categorized into three distinct frameworks: Horizontal Federated Learning (HFL), Vertical Federated Learning (VFL), and Federated Transfer Learning (FTL) [72]. The categorization of data composition in FL is presented in Table 2.2, which summarizes the distinct partitioning strategies, challenges, and typical applications of HFL, VFL, and FTL. This classification highlights how each framework addresses unique data distribution scenarios.

2.3.1 Horizontal Federated Learning

Horizontal Federated Learning (HFL) is designed for cases where clients share a common feature space but differ in their data samples. This setup, commonly referred to as sample-partitioned FL, involves datasets that are horizontally partitioned in a tabular format, where each row represents a unique sample, and columns represent shared features [73, 72]. HFL is particularly suited for organizations that operate in similar domains but cater to different user bases, such as regional banks or hospitals [73, 56].

In HFL, the alignment of feature spaces across clients allows the same machine learning model architecture to be deployed for local training. For example, multiple hospitals collaborating to develop a diagnostic model may share the same features, such as patient demographics and test results, but the data samples represent patients from different locations. HFL ensures that sensitive patient information remains private while enabling collective insights through a shared global model [73].

Despite its advantages, HFL often encounters challenges due to Non-IID (Non-independent and identically distributed) data distributions. Variations in data characteristics, such as imbalances in class distributions or disparities in dataset sizes across clients, can lead to client drift, where local updates deviate from the global model objective [73]. To address these issues, advanced aggregation methods such as FedAvg are commonly employed [46]. However, in highly heterogeneous data settings, modifications to FedAvg, such as FedProx, which introduces a regularization term, are often necessary to mitigate client divergence [35]. Additionally, personalized FL methods, which adapt the global model to individual client needs, and clustered FL, which groups clients based on similar data distributions, further enhance the performance of heterogeneous HFL in Non-IID scenarios [61, 36].

Simulations of Non-IID data are often conducted using a Dirichlet distribution parameterized by α, where smaller values indicate highly heterogeneous datasets. This approach allows researchers to systematically evaluate the robustness of FL under varying degrees of data heterogeneity [73, 27].

2.3.2 Vertical Federated Learning

Vertical Federated Learning (VFL) is applicable when clients share overlapping data samples but differ in their feature spaces. This setup, also referred to as feature-partitioned FL, addresses scenarios where organizations possess complementary feature sets for the same entities [73, 72]. For instance, a bank and an e-commerce platform may have a shared customer base but maintain distinct datasets—financial transactions for the bank and purchase histories for the e-commerce platform. VFL enables these organizations to collaboratively train a predictive model while preserving data privacy [73, 56].

In VFL, aligning shared samples is a critical first step. This process is typically achieved using secure protocols such as secure multi-party computation (SMPC) or homomorphic encryption, which ensure that overlapping samples are identified without revealing raw data [39]. Once aligned, clients collaboratively train the model by securely sharing partial computations, often leveraging techniques like split learning [72, 39]. Split learning divides the model into segments, with lower layers trained locally by each client and higher layers jointly optimized [39]. This approach allows deep learning on vertically partitioned data while minimizing the risk of privacy breaches [42, 39].

The computational complexity of VFL is often higher than that of HFL due to the need for secure alignment and joint optimization [39]. However, it offers unparalleled advantages in scenarios requiring the integration of complementary knowledge from diverse feature sets, such as healthcare research combining clinical and genomic data or cross-industry collaborations in finance and retail [73, 27].

2.3.3 Federated Transfer Learning

Federated Transfer Learning (FTL) addresses scenarios where clients have minimal or no overlap in both samples and features. This framework is particularly useful when clients operate in different domains with distinct datasets. FTL leverages the principles of transfer learning to enable collaboration between clients with heterogeneous data, making it a powerful solution for addressing data scarcity and domain mismatches [73, 72].

For example, a healthcare provider and a retail chain may collaborate to train a predictive model, even though their datasets differ entirely in samples and features. FTL bridges these gaps by employing techniques such as instance-based transfer, which reweights training samples to align domain distributions, and feature-based transfer, which learns a shared feature representation space to improve knowledge transfer [38]. Model-based transfer further enhances FTL by utilizing pre-trained models from resource-rich domains to bootstrap training in resource-scarce environments [73, 38].

FTL's flexibility allows it to address challenges that traditional HFL and VFL frameworks cannot. It is particularly valuable in applications such as cross-domain collaborations for sustainability, where organizations with disparate datasets share a common goal but lack sufficient overlap for direct model sharing [38].

Thesis Focus. While HFL, VFL, and FTL each address distinct data distribution scenarios, this thesis primarily concentrates on HFL. The sample-partitioned setting, where clients share a common feature space but hold different data samples, aligns with the security challenges explored in the following chapters. Consequently, all proposed methods and experiments target adversarial threats within an HFL context, utilizing Dirichlet distributions to simulate Non-IID data conditions.

2.3.4 Challenges of Non-IID Heterogeneity

Non-IID data distributions present a critical challenge in FL, stemming from variations in class distributions, feature representations, and dataset sizes across clients [41]. Unlike the assumptions of traditional machine learning, where data is often considered independent and identically distributed (IID), FL must contend with diverse and often conflicting local datasets [73]. This diversity reflects real-world scenarios, such as different demographic distributions across regions or varying user behaviors on personal devices. These disparities disrupt the alignment between local updates and the global optimization objective, complicating model convergence and reducing performance. Notable challenges include client drift, where local objectives deviate from the global goal, and imbalanced contributions, where clients with smaller datasets or unique distributions disproportionately affect the aggregated global model [72].

Client drift arises because clients independently optimize their local objectives, which are shaped by their specific data distributions. For example, a client with data skewed toward a particular class will generate gradients that emphasize that class, leading to divergence from the global model objective [73]. This phenomenon not only delays convergence but also results in a global model that may generalize poorly across all clients. Additionally, imbalanced contributions exacerbate these challenges. Clients with larger datasets dominate the aggregation process, overshadowing updates from clients with smaller or highly heterogeneous datasets [44]. This imbalance reduces model fairness and generalization, especially in cases where minority classes or underrepresented distributions are crucial.

To mitigate these challenges, advanced techniques have been developed. Scaffold introduces control variates to counteract client drift by correcting deviations in gradient directions, ensuring updates align more closely with the global objective [29]. This approach significantly improves convergence rates and robustness in non-IID environments. Similarly, FedNova addresses imbalances by normalizing local updates, accounting for differences in dataset sizes and the number of training epochs [35]. These normalization strategies ensure that all clients contribute equitably to the global model, regardless of their data volume or training duration.

Personalization strategies have also proven effective in addressing non-IID heterogeneity. Meta-learning frameworks optimize the global model to serve as a foundation for rapid adaptation to individual client distributions [71]. This approach is particularly valuable in applications like personalized healthcare or recommendation systems, where client-specific models are critical. Fine-tuning offers another layer of personalization, enabling clients to adapt the global model to their local data without compromising privacy [44]. Such strategies ensure that global models remain robust while meeting the unique requirements of individual clients.

Another promising solution is clustered FL, which groups clients with similar data characteristics into clusters. By training submodels for each cluster, this approach accommodates heterogeneity in data distributions without compromising the overall training process [71].

In addition to these approaches, synthetic data generation and adaptive learning strategies are gaining traction [41]. By generating synthetic datasets that mimic underrepresented distributions, researchers can balance contributions and improve global model generalization. Similarly, adaptive aggregation methods dynamically adjust learning rates or weighting schemes based on client-specific metrics, further enhancing robustness in non-IID settings [44].

Table 2.2: Data Composition Categories in Federated Learning

Framework	**Data Partitioning**	**Challenges**	**Common Use Cases**
Horizontal Federated Learning (HFL)	Shared features, distinct samples	Non-IID data leading to client drift, imbalances in dataset size and class distribution	Collaborations across organizations with similar domains, such as regional banks or hospitals
Vertical Federated Learning (VFL)	Shared samples, distinct features	Computational complexity due to secure sample alignment and joint optimization	Cross-industry collaborations (e.g., banks and e-commerce platforms), healthcare research combining clinical and genomic data
Federated Transfer Learning (FTL)	No overlap in samples or features	Domain mismatches, data scarcity, high heterogeneity in client data	Cross-domain collaborations in sustainability, education, and resource-scarce settings

2.4 Security Threats in Federated Learning

FL framework facilitates collaborative model training across distributed clients while preserving privacy by avoiding the need to exchange raw data of clients. This privacy-preserving nature makes FL a preferred solution in sensitive domains like healthcare, finance, and IoT [56]. However, FL's reliance on decentralized architectures and distributed computation introduces substantial security vulnerabilities. Adversaries can exploit these vulnerabilities to compromise the integrity of the global model performance, disrupt the training process, or embed malicious behaviors into system [42]. Key security concerns in FL revolve around poisoning attacks and backdoor attacks, both of which target the training process to undermine the FL system's robustness and reliability [56]. A deep analysis of these threats and their implications is essential to understand the resilience of FL systems.

2.4.1 Poisoning Attacks

Poisoning attacks in FL are among the most prevalent and harmful threats, leveraging the decentralized nature of training to introduce malicious updates. These attacks are aimed at manipulating the training process to degrade model performance or induce specific undesired behaviors. Based on the adversary's goals, poisoning attacks are classified as random attacks or targeted attacks [35].

Random poisoning attacks introduce noise or malicious updates into the training process, causing general degradation of the global model's performance. These attacks are relatively easier to detect as they often result in noticeable accuracy drops across tasks [56]. In contrast, targeted poisoning attacks are more sophisticated, focusing on achieving specific goals, such as causing the model to misclassify particular inputs [34]. Targeted attacks are harder to detect because they aim to maintain overall model performance while embedding malicious behaviors [56].

Poisoning attacks can be further classified based on the stage of the FL pipeline they target: data poisoning and model poisoning [35].

2.4.1.1 Data Poisoning

Data poisoning attacks exploit the decentralized nature of FL, where clients maintain control over their local datasets. These attacks target the data preparation phase of the FL pipeline, aiming to introduce biases or malicious patterns that degrade the performance or reliability of the global model [56]. By manipulating the training data, adversaries can influence the learning process to align with their objectives. Data poisoning attacks are broadly categorized into two types: clean-label attacks and dirty-label attacks, each with distinct methodologies and implications [47].

Clean-label attacks are characterized by their subtlety and stealth. In these attacks, adversaries imperceptibly alter the features of training samples while preserving their original labels [5]. The primary goal is to bias the model's decision-making process without raising suspicion [68]. For instance, an adversary might slightly modify pixel intensities in an image of a handwritten digit, ensuring that the changes are not noticeable to human observers or automated validation processes. Despite these imperceptible alterations, the global model's decision boundary can become skewed, leading to misclassifications or reduced accuracy [42].

The effectiveness of clean-label attacks lies in their ability to remain undetected. Since the poisoned samples retain their original labels, they appear consistent with the rest of the dataset, making anomaly detection challenging [47]. Such attacks are particularly dangerous in FL, where the server does not have direct access to client data and cannot inspect individual samples for tampering. Clean-label attacks are often employed as a vector for backdoor attacks, where a hidden trigger embedded in the data can activate adversary-defined behavior during inference [18]. For example, an image classification model might be trained to misclassify any input containing a specific watermark as a particular class, while functioning normally for other inputs.

In contrast, dirty-label attacks involve manipulating the labels of specific samples while leaving their features unchanged. These attacks typically follow straightforward strategies, such as flipping the labels of one class to another [62]. For instance, in a label-flipping attack, an adversary may reassign the labels of all images of the digit "1" to "7". This results in a model that misclassifies"1" as "7" during inference. Such attacks are relatively easy to execute because they do not require advanced technical skills or significant computational resources.

However, dirty-label attacks are generally more detectable compared to clean-label attacks. The mismatch between the features and labels creates an inconsistency that can be flagged by validation mechanisms or statistical analyses [49]. Despite their detectability, dirty-label attacks can still pose risks, especially when conducted by multiple colluding clients or in environments with limited model validation.

While both clean-label and dirty-label attacks aim to compromise the integrity of the global model, their approaches and challenges differ. Clean-label attacks are subtle, making them difficult to detect but technically more sophisticated to implement. They often require a deeper understanding of the model's decision boundaries to design effective feature modifications. Dirty-label attacks, on the other hand, are simpler and more accessible but carry a higher risk of detection due to the visible inconsistencies between features and labels.

The decentralized and privacy-preserving architecture of FL exacerbates the risks associated with data poisoning attacks. The lack of centralized oversight and the reliance on client-provided data create opportunities for adversaries to execute both clean-label and dirty-label attacks with minimal risk of immediate detection. Addressing these vulnerabilities requires robust defense mechanisms, including anomaly detection, validation protocols, and aggregation techniques designed to minimize the impact of poisoned data.

2.4.1.2 Model Poisoning

Model poisoning attacks are a sophisticated threat in FL that target the updates shared by clients with the central server. Unlike data poisoning, where the attack focuses on manipulating the training data, model poisoning directly intervenes in the training process by altering gradients or model parameters [9]. This strategic targeting of the aggregation process allows adversaries to exert greater influence on the global model's optimization while circumventing the challenges associated with data manipulation. The impact of model poisoning can range from degrading overall model performance (untargeted attacks) to embedding specific malicious behaviors (targeted attacks) [20].

In untargeted model poisoning attacks, the adversary introduces random noise or modifies updates to reduce the accuracy of the global model. The goal is to disrupt the training process, resulting in a model that performs poorly across all tasks. [9] This type of attack is typically easier to execute and harder to detect, as the noise appears random and does not follow a specific pattern [11].

Conversely, targeted model poisoning attacks aim to embed specific behaviors or biases into the global model. A common example is the insertion of backdoors, where an adversary modifies local updates to ensure that inputs with a predefined pattern or trigger are misclassified [18]. For instance, an image classification model may be manipulated to classify all images with a specific watermark as a particular class, regardless of their actual content. Targeted attacks are often stealthy, as they maintain normal model performance on clean inputs while activating malicious behaviors only under specific conditions [63].

Several sophisticated strategies are employed in model poisoning attacks, each designed to evade detection while achieving the attacker's objectives.

Inner Product Manipulation (IPM) Attack. The IPM attack is a stealthy model poisoning strategy that aligns the malicious update with the global model direction while preserving benign-like statistical properties [69]. This design allows the attacker to evade detection during aggregation by mimicking the orientation of benign updates. The adversary crafts the malicious gradient update $\Delta \mathbf{g}_t^i$ by projecting a crafted gradient vector onto the global model direction and scaling the result:

$$\Delta \mathbf{g}_t^i = \epsilon \, \frac{\langle \mathbf{g}_t^i, \mathbf{w} \rangle}{\|\mathbf{w}\|^2} \, \mathbf{w} \tag{2.4}$$

where $\langle \mathbf{g}_t^i, \mathbf{w} \rangle$ denotes the inner product, $\|\mathbf{w}\|$ is the ℓ_2-norm of the global model, and ϵ is a scalar controlling the attack strength. As a result, the update in Equation 2.4 retains the direction of the model vector, blending seamlessly with benign contributions while subtly degrading model performance.

A Little is Enough (ALIE) Attack. The ALIE attack injects carefully crafted noise to exploit the statistical variability among benign client updates, thereby deceiving the aggregation mechanism [7]. Specifically, the attacker estimates the empirical mean $\boldsymbol{\mu}_i$ and standard deviation $\boldsymbol{\delta}_i$ for each parameter coordinate i based on observed benign gradients. The malicious update is then constructed to lie within the interval defined in Equation 2.5:

$$(\boldsymbol{\mu}_i - z_{\max}\boldsymbol{\delta}_i,\ \boldsymbol{\mu}_i + z_{\max}\boldsymbol{\delta}_i)\,, \tag{2.5}$$

where $z_{\max}$ is a scalar threshold derived from the cumulative standard normal distribution. By remaining within this plausible statistical range, the attacker's update appears benign to the server, yet gradually degrades the global model's performance. This stealthy approach allows the attacker to bypass robust aggregation defenses.

Sign Flipping (SF) Attack. The sign flipping (SF) attack is a simple yet effective model poisoning technique that reverses the direction of gradient updates to disrupt learning [9]. Unlike more sophisticated attacks such as IPM or ALIE, SF does not require knowledge of other clients' updates, making it accessible and easy to implement for adversaries. In practice, an attacker inverts the sign of each component in its gradient update:

$$\Delta\mathbf{g}^i_{\text{malicious}} = -\Delta\mathbf{g}^i_{\text{benign}}, \tag{2.6}$$

where $\Delta\mathbf{g}^i_{\text{benign}}$ is the original local gradient of client i, and the resulting update $\Delta\mathbf{g}^i_{\text{malicious}}$ pushes the global model in the opposite direction, effectively performing gradient ascent. This attack maximizes the local loss and significantly hinders model convergence (see Equation 2.6).

Random Noise (RN) Attack. The RN attack introduces unstructured noise into model updates to disrupt the learning process [73]. Unlike targeted attacks, it does not require knowledge of the model structure or other clients' updates, making it broadly accessible. The adversary generates noise from a zero-mean Gaussian distribution $\mathcal{N}(0, \sigma^2)$ and adds it to the model gradients, as shown in Equation 2.7:

$$\Delta \mathbf{g}_t^i = \mathbf{g}_t^i + \mathcal{N}(0, \sigma^2), \tag{2.7}$$

where $\mathbf{g}_t^i$ is the original gradient of client i, and σ controls the noise strength. The zero-mean ensures the perturbation is unbiased, while the variance σ^2 allows the attacker to adjust the noise level. Despite its simplicity, the RN attack can effectively degrade model performance, especially in settings with weak defenses or high heterogeneity.

Model poisoning attacks pose significant risks to the integrity and reliability of FL systems. By targeting the aggregation process, these attacks can compromise the global model in ways that are challenging to detect and mitigate. The adaptive nature of advanced attacks like IPM and ALIE, combined with the decentralized architecture of FL, exacerbates the difficulty of defense.

Mitigation strategies for model poisoning attacks include robust aggregation techniques, such as Byzantine-resilient algorithms that detect and exclude anomalous updates. Statistical analysis of updates, combined with secure aggregation protocols, can also help identify and mitigate malicious behavior. Additionally, techniques like differential privacy and cryptographic methods provide additional layers of defense by ensuring that individual updates remain secure and verifiable.

2.4.2 Backdoor Attacks

Backdoor attacks represent a sophisticated form of targeted poisoning attack, specifically designed to embed hidden triggers within the global model. These triggers activate malicious behaviors only under predefined conditions, such as a particular pattern, watermark, or other identifiable features in the input data [63]. For instance, in an image classification model, a backdoor attack might train the model to classify any image containing a specific pattern (e.g., a logo or a pixel arrangement) as a particular class, irrespective of the true content of the image. The insidious nature of backdoor attacks lies in their stealth; the model continues to perform normally under standard conditions, making the backdoor exceedingly difficult to detect without knowledge of the trigger.

Backdoor attacks can be executed via two main mechanisms: data poisoning and model poisoning. In data poisoning scenarios, adversaries inject training samples containing the trigger pattern into their local datasets and associate these samples with a target class. The poisoned samples influence the global model during aggregation, causing it to associate the trigger pattern with the adversary's chosen class [42]. During inference, inputs containing the trigger pattern are misclassified as the target class. In contrast, model poisoning directly targets the gradients or model parameters shared with the central server, embedding the backdoor more directly [43]. This method often bypasses the need for manipulated data, relying instead on adversarial gradient updates to achieve the desired behavior.

The success of backdoor attacks depends significantly on the adversary's ability to influence the FL training pipeline while evading detection. Even a single malicious client, if undetected, can introduce a backdoor that significantly compromises the reliability and integrity of the global model. This vulnerability underscores the importance of robust defense mechanisms in FL systems.

Several sophisticated strategies are employed in backdoor attacks, each carefully designed to embed malicious behaviors into the global model while evading detection. These strategies leverage both data and model poisoning techniques to ensure the backdoor remains stealthy and effective.

Adversarially Adaptive Backdoor Attack to Federated Learning (A3FL). The A3FL attack enhances the persistence and effectiveness of backdoors by dynamically adapting triggers to the evolving training dynamics of FL systems [76]. Unlike static backdoors, A3FL optimizes its triggers iteratively to ensure compatibility with both the current global model and adversarially crafted variants. By employing adversarial adaptation loss and Projected Gradient Descent (PGD), A3FL continuously refines the backdoor, ensuring its robustness across multiple training updates and iterations.

Focused-Flip Federated Backdoor Attack (F3BA). The F3BA strategy narrows its manipulation to a subset of critical model parameters, altering them selectively to embed a backdoor while minimizing disruption to overall model performance [18]. Each parameter's importance is quantified using a sensitivity score defined in Equation 2.8:

$$\mathbf{S}_j = -\left(\frac{\partial L_g}{\partial \mathbf{w}_j}\right) \odot \mathbf{w}_j, \tag{2.8}$$

where $\mathbf{S}_j$ captures the sensitivity of the global loss function L_g to the parameter $\mathbf{w}_j$, and $\odot$ denotes element-wise multiplication. The attacker identifies the most sensitive parameters based on $\mathbf{S}_j$ and flips their signs to embed the trigger, achieving a balance between stealth and backdoor effectiveness.

Cerberus Poisoning (CerP) Attack. The CerP introduces a stealthy and distributed backdoor attack by fine-tuning backdoor triggers, controlling local model parameter biases, and maximizing diversity among malicious updates. By exploiting defense assumptions, CerP minimizes deviations between poisoned and benign models, achieving high attack success rates while preserving the main learning task's accuracy [43].

Distributed Backdoor Attack (DBA). The DBA spreads a trigger pattern across multiple adversarial clients, enhancing stealth and making detection more difficult. Each compromised client injects a portion of the full trigger into its local training data. When these local models are aggregated, the global model inadvertently learns to associate the combined trigger pattern with the target class [68]. As a result, inputs containing the full trigger pattern are misclassified, effectively executing the backdoor attack without any single client contributing a suspiciously large modification.

Characteristics of Backdoor Attacks

Backdoor attacks often exhibit distinctive characteristics in the model updates sent by malicious clients, which can be exploited for detection. Key characteristics include:

- **Angular Deviation**: Malicious model updates may have a different direction in the parameter space compared to updates from benign clients [5]. This directional difference can be quantified using the cosine similarity (or angular deviation) between the parameter vectors of malicious updates $\mathbf{w}_{\text{attack}}$ and benign updates $\mathbf{w}_{\text{benign}}$:

$$\Delta\mathbf{w}_{\text{angular}} = \cos^{-1}\left(\frac{\mathbf{w}_{\text{attack}} \cdot \mathbf{w}_{\text{benign}}}{\|\mathbf{w}_{\text{attack}}\| \, \|\mathbf{w}_{\text{benign}}\|}\right) \tag{2.9}$$

- **Magnitude Deviation**: Malicious updates may have a significantly different norm magnitude compared to benign updates [68]. The magnitude deviation is observed when:

$$\|\mathbf{w}_{\text{attack}}\| \gg \|\mathbf{w}_{\text{benign}}\| \tag{2.10}$$

- **Subtle Deviations**: Some attackers design their updates to closely resemble those of benign clients, keeping the deviation within a small threshold ϵ to avoid detection [76]:

$$\|\mathbf{w}_{\text{attack}} - \mathbf{w}_{\text{benign}}\| < \epsilon \tag{2.11}$$

Equations 2.9, 2.10, and 2.11 describe distinct attack patterns that can be leveraged to detect backdoor behaviors without access to raw training data.

2.4.3 Interplay Between Poisoning and Backdoor Attacks

The relationship between poisoning and backdoor attacks is both intricate and significant, as these two strategies often overlap and amplify each other's impact. While poisoning attacks primarily aim to degrade the performance or usability of the global model by altering either training data or model updates [7], backdoor attacks exploit these alterations to embed hidden triggers into the model [9]. The triggers remain dormant under normal conditions but activate specific adversarial behaviors when encountering predefined inputs [18]. Understanding the interplay between these attack types is crucial for developing comprehensive defenses in FL.

Both poisoning and backdoor attacks exploit the decentralized and distributed nature of FL. In data poisoning, adversaries manipulate local training datasets to bias the model's learning process [63]. This manipulation can range from straightforward label flipping to more subtle clean-label attacks, where seemingly benign samples are crafted to alter the decision boundaries of the global model. When data poisoning is tailored to inject specific patterns or triggers, it seamlessly transitions into a backdoor attack [5].

Model poisoning, on the other hand, operates at the gradient or parameter level, directly tampering with the updates sent to the central server. By modifying the gradients or parameters, adversaries can steer the global model towards a malicious objective, such as embedding a backdoor [68]. Model poisoning can also enhance the stealth and effectiveness of backdoor attacks by bypassing the need for visible data manipulation [18]. In practice, poisoned datasets often result in compromised model updates, illustrating how model poisoning subsumes data poisoning under certain conditions.

Clean-Label data poisoning as a backdoor vector. A clean-label data poisoning attack serves as a prime example of the interplay between these strategies. Adversaries introduce training samples with imperceptible modifications—such as slight pixel adjustments in an image—that embed hidden triggers without altering the sample labels [68]. During the training process, these poisoned samples bias the model's parameters to associate the trigger pattern with a specific target class. Upon aggregation, the global model inherits this backdoor behavior, effectively combining the mechanisms of data poisoning and backdoor implantation.

Gradient manipulation in backdoor model poisoning. Model poisoning attacks offer another pathway for embedding backdoors. By directly manipulating the gradients or parameters shared during training, adversaries can embed triggers without relying on poisoned data. This approach is particularly effective in scenarios where direct data manipulation is infeasible or detectable. For instance, adversaries can craft gradient updates that align with the global optimization objective while embedding malicious behaviors. Techniques like A3FL [76] attack or scaling malicious updates amplify the backdoor effect while evading detection during aggregation.

Amplification through combined strategies. The interplay between poisoning and backdoor attacks can amplify their individual impacts. A hybrid approach, where adversaries poison both the data and model updates, creates a synergistic effect [18]. For example, a backdoor attack may start with data poisoning to introduce triggers in local datasets. Concurrently, model poisoning fine-tunes the gradients to reinforce the trigger's impact during aggregation. This dual strategy not only increases the likelihood of successful backdoor embedding but also enhances its persistence and stealth.

Implications for FL security. The convergence of poisoning and backdoor attacks underscores the complexity of securing FL systems. Defense mechanisms must address both data integrity and model update authenticity, recognizing that seemingly independent attack vectors can reinforce each other. Robust aggregation techniques, anomaly detection based on update similarity, and adversarial training are critical to mitigating the compounded risks posed by these interrelated threats.

In summary, poisoning and backdoor attacks are not isolated phenomena but interdependent strategies that exploit the vulnerabilities of FL framework. Their interplay highlights the need for defenses that consider the full spectrum of adversarial tactics, ensuring the robustness of collaborative learning frameworks.

Thesis Focus. The security threats in FL encompass data poisoning, model poisoning, and stealthy backdoor attacks. In this thesis, we develop defense mechanisms to counter these threats in a centralized, cross-device FL setting under Horizontal Federated Learning assumptions. Our aim is to detect and mitigate adversarial behaviors without compromising model performance.

2.5 Related Work

FL introduces a transformative approach to collaborative machine learning, allowing decentralized training while preserving data privacy. This paradigm mitigates the need for data centralization, aligning with privacy regulations and user trust. However, the decentralized nature of FL presents significant security challenges, exposing the system to various adversarial threats. Among these, poisoning and backdoor attacks stand out as critical threats that can degrade the integrity, utility, and reliability of the global model [38].

Poisoning attacks manipulate the training process by introducing malicious updates or data, disrupting the global model's performance and effectiveness [20]. Backdoor attacks, on the other hand, embed hidden triggers into the model, causing it to exhibit specific malicious behaviors under predefined conditions [18]. These adversarial strategies exploit the inherent characteristics of FL, such as non-IID data distributions, limited visibility into client operations, and the absence of centralized monitoring mechanisms [47].

Robust defense mechanisms are crucial to counter these threats while upholding the core principles of FL, including data privacy and decentralization. Unlike traditional machine learning systems that benefit from centralized oversight and comprehensive monitoring, FL requires innovative and decentralized solutions tailored to its unique operational challenges.

This section delves into the defense mechanisms designed to mitigate the impact of poisoning and backdoor attacks in FL. The discussion is organized into two primary categories: Robust Aggregation Defenses, which focus on safeguarding the aggregation process against malicious contributions, and Backdoor-Specific Defenses, which aim to detect and neutralize hidden triggers embedded in the global model. By addressing these adversarial strategies, the defenses discussed in this section highlight the ongoing efforts to secure the FL ecosystem without compromising its privacy-preserving capabilities or model performance.

2.5.1 Robust Aggregation Defenses

Robust aggregation mechanisms play a pivotal role in mitigating the impact of adversarial updates in FL. These defenses aim to ensure that the global model remains resilient to malicious or outlier contributions, which could otherwise compromise its integrity and utility. Given the decentralized and privacy-preserving nature of FL, where raw data cannot be directly inspected, robust aggregation must operate effectively under constraints of limited visibility and diverse, often non-IID, data distributions.

This subsection delves into three primary categories of robust aggregation defenses: Distance-Based Filtering, Statistical Distribution-Based Aggregation, and Proxy Dataset-Based Validation. Each approach tackles adversarial threats from a distinct perspective, offering insights into their mechanisms, strengths, and limitations.

2.5.1.1 Distance-Based Filtering

Distance-based filtering approaches identify and exclude malicious updates by measuring their deviation from the expected distribution of client updates [48, 59, 66, 6]. The fundamental assumption is that benign updates cluster closely around a central trend, while adversarial updates deviate significantly [9].

Multi-Krum selects a subset of client updates that are most similar to their neighbors, measured by Euclidean distance [11]. By prioritizing updates that align with the majority, this method minimizes the influence of outliers, which are often indicative of adversarial behavior. Multi-Krum is particularly effective when the majority of clients are benign. However, its reliance on the dominance of benign updates reduces its effectiveness in scenarios with high adversarial participation or heterogeneous data distributions.

FoolsGold employs cosine similarity to evaluate the alignment of updates across clients [19]. By identifying updates that frequently align too closely—potentially indicative of collusion—it assigns lower weights to such updates during aggregation.

This approach is highly effective against collusion-based attacks but struggles in non-IID settings, where legitimate similarities in updates could be misinterpreted as adversarial.

FABA (Fast Aggregation Against Byzantine Attacks) iteratively removes updates that deviate significantly from the mean before computing the global model. This iterative filtering effectively isolates outliers [66]. However, its reliance on mean-based criteria makes it less robust in non-IID settings, where benign updates naturally exhibit higher variability.

2.5.1.2 Statistical Distribution-Based Aggregation

Statistical approaches leverage robust statistical techniques to aggregate updates in a manner that minimizes the influence of outliers, without explicitly identifying or excluding them.

The median is a robust statistical measure that minimizes the impact of extreme values. In FL, median aggregation involves computing the median of each parameter across client updates [74]. This approach ensures that outliers, whether malicious or resulting from data variability, have limited influence on the global model. Median aggregation is computationally efficient and performs well in scenarios where adversarial updates are sparse. However, it assumes that updates are symmetrically distributed around the central trend, which may not hold in highly non-IID settings.

RFA (Robust Federated Aggregation) calculates the geometric median of client updates, offering greater resistance to outliers compared to traditional mean-based aggregation [54]. By employing alternating minimization, RFA achieves stable convergence while maintaining robustness. However, its computational overhead can be significant in large-scale FL systems. Additionally, RFA assumes that client updates are bounded within a specific range, which may not hold in highly heterogeneous data scenarios.

Bulyan combines geometric median and trimmed median [74] techniques in a two-step process. Initially, it identifies a subset of updates close to the geometric median, excluding outliers [22]. It then computes a trimmed mean from the selected updates to finalize the aggregation. While this dual-layer approach enhances robustness, it is computationally intensive and sensitive to extreme heterogeneity in data distributions.

Building on the Krum algorithm, Dim-Krum identifies abnormal client updates by examining a small subset of dimensions with higher backdoor strengths [77]. This selective approach allows the server to isolate and remove malicious contributions. Dim-Krum is particularly effective in settings with a high adversarial presence but assumes that benign clients dominate numerically.

RLR (Robust Learning Rate) adjusts the server's learning rate dynamically based on the sign information of client updates [52]. By analyzing updates across dimensions and training rounds, RLR identifies anomalies indicative of adversarial behavior. This approach enhances model robustness against both backdoor and malicious attacks.

Clipping methods restrict the magnitude of client updates to a predefined threshold, reducing the influence of outliers introduced by adversarial updates [23]. By limiting the norm of each client update, clipping prevents disproportionately large updates from dominating the aggregation process.

Centered clipping enhances this approach by centering updates around a reference point, often the mean or median of all updates, before applying the clipping threshold [28]. This adjustment ensures that updates are normalized relative to a central trend, further mitigating the impact of malicious contributions. While both methods are computationally efficient, their effectiveness depends on carefully chosen thresholds that balance robustness and utility. Excessive clipping can overly suppress legitimate variations in updates, particularly in heterogeneous FL settings.

2.5.1.3 Proxy Dataset-Based Validation

Proxy dataset-based methods incorporate a trusted, auxiliary dataset to evaluate the reliability of client updates. This dataset serves as a benchmark to assess how closely client updates align with the expected model behavior.

FLTrust uses a small, clean proxy dataset to evaluate client updates. By calculating a reference gradient on the proxy data, it measures the cosine similarity of client updates with this reference, reweighting updates accordingly [13]. While FLTrust effectively identifies and suppresses malicious contributions, its dependence on a high-quality proxy dataset limits its applicability in privacy-sensitive or heterogeneous domains where such datasets may not be available.

SageFlow evaluates client updates based on their impact on model entropy when applied to a proxy dataset [53]. Updates that cause high entropy are assigned lower weights, reflecting their potential unreliability. While this entropy-based filtering provides a nuanced evaluation of update reliability, it shares FLTrust's limitation of requiring a clean and representative proxy dataset, which may not always align with the diversity of FL client data.

The effectiveness of robust aggregation defenses is influenced by underlying assumptions about data distributions and adversarial behavior. Distance-based and statistical methods are well-suited for IID settings but often struggle with non-IID data, where natural variability among benign updates can mimic adversarial patterns. Proxy dataset-based methods address some of these limitations by providing an external benchmark, but their reliance on auxiliary data raises concerns about feasibility and privacy.

In practice, the choice of robust aggregation defense depends on the specific characteristics of the FL deployment. Systems with relatively homogeneous data and low adversarial risk may benefit from simpler, distance-based methods. Conversely, highly heterogeneous and adversarial environments demand more sophisticated approaches, potentially combining statistical robustness with adaptive evaluation mechanisms to ensure both security and model performance.

2.5.2 Backdoor Defense Mechanisms

Backdoor attacks pose a significant threat to FL systems by embedding hidden triggers into the global model. These attacks exploit the decentralized nature of FL to introduce adversarial behaviors that remain dormant under normal conditions but activate malicious outputs when specific triggers are present. Unlike broader poisoning attacks, backdoor attacks are often more insidious, as they aim to preserve the overall accuracy of the model while introducing targeted vulnerabilities. Addressing these threats requires specialized defense mechanisms that go beyond traditional aggregation strategies.

2.5.2.1 Model Refinement Approaches

Model refinement techniques focus on post-aggregation interventions to cleanse the global model of potential backdoor triggers.

Fine-tuning involves retraining the global model on a small, trusted dataset to mitigate backdoor effects [55]. By exposing the model to clean data, the backdoor association between the trigger and the adversarial target label can be weakened. However, fine-tuning relies heavily on the availability of an auxiliary dataset that closely aligns with the original training data, which may conflict with FL's privacy-preserving principles.

Pruning aims to identify and remove specific neurons or layers associated with the backdoor's activation [65]. By systematically reducing the model's complexity, pruning can suppress malicious behaviors. However, excessive pruning risks degrading the model's overall performance, particularly in non-IID settings.

Knowledge distillation transfers the knowledge of the global model to a new model by training it on the outputs of the original model, typically using a clean proxy dataset [60]. This approach assumes that the distilled model will inherit only benign behaviors, effectively erasing backdoors. Distillation's effectiveness is constrained by the availability of high-quality auxiliary data.

Model refinement approaches are often limited by their reliance on clean datasets and the risk of overfitting or underperforming if the auxiliary dataset does not adequately represent the diversity of client data.

2.5.2.2 Dynamic Client Trust Scoring

Dynamic trust scoring mechanisms assign trust levels to clients based on their historical behavior and alignment with global model. These scores are then used to weigh client updates during aggregation, reducing the impact of malicious contributions and behavior.

Trust scoring based on update consistency ensures that clients that consistently submit updates aligned with the global optimization direction are assigned higher trust scores [48]. Conversely, clients with erratic or anomalous updates receive lower scores. This dynamic approach ensures that the aggregation process prioritizes reliable contributors.

Reputation systems track client behavior over multiple rounds of training, identifying patterns indicative of malicious intent [48]. Clients with a history of submitting suspicious updates are flagged and excluded from future aggregation.

Behavioral adaptation ensures that trust scores can adapt based on the observed behavior of clients across different training rounds [17]. This approach ensures that previously benign clients that turn malicious are identified and penalized.

Dynamic trust scoring offers a adaptive defense against backdoor attacks, particularly in environments with heterogeneous data distributions [64]. However, its reliance on historical data may delay the detection of newly compromised clients.

Backdoor defense mechanisms form a critical component of FL's security architecture, complementing robust aggregation strategies. While model refinement techniques cleanse the global model post-aggregation and dynamic trust scoring address threats during the training process. Each approach has its strengths and limitations, and their effectiveness depends on factors such as data availability, computational resources, and the sophistication of the adversarial strategy.

2.5.3 Overall Summary and Thesis Contributions

Existing defenses against poisoning and backdoor attacks in FL generally fall into two categories. Some approaches focus on robust aggregation, using methods such as distance based filtering, statistical distribution based strategies, or proxy dataset validation to counter malicious updates. Others target hidden triggers through post aggregation model refinement and adaptive trust scoring. Although these methods have improved FL security, many still struggle with highly diverse Non-IID data, require external datasets, or entail high computational costs. These limitations complicate the deployment of secure and scalable FL systems. Table 2.3 summarizes major obstacles that persist in FL defense mechanisms.

Thesis Contributions. To address these challenges, this thesis proposes a framework that adapts to Non-IID data by introducing specialized aggregation and anomaly detection methods suited for heterogeneous client distributions. It also minimizes auxiliary data reliance by reducing or eliminating the need for clean external clients datasets. Finally, it strengthens detection and mitigation to effectively counter both poisoning and backdoor threats without severely degrading model performance. The following chapters detail each component of this framework, present rigorous experimental evaluations, and demonstrate how the proposed methods significantly improve FL security in adversarial settings while preserving user privacy.

Table 2.3: Key Challenges and Limitations in FL Defenses

Aspect	**Challenges**	**Limitations**
Data Heterogeneity	Difficulty distinguishing malicious updates from natural variations *(Addressed in Chapter 3, Chapter 4, and Chapter 5)*	Most methods assume IID data, limiting applicability in Non-IID scenarios
Computational Overhead	High resource demands for robust aggregation and anomaly detection *(Addressed in Chapters 3)*	May not scale well in large systems
Auxiliary Data Needs	Reliance on proxy datasets for tuning and validation *(Mitigated in Chapter 5 via synthetic calibration data)*	Conflicts with FL's privacy principles
Adaptive Attacks	Evolving backdoor techniques that evade traditional detection *(Tackled in Chapter 4 and Chapter 5)*	Existing defenses lag behind new attack tactics
Performance Balance	Aggressive clipping or pruning risks suppressing valid updates *(Addressed in Chapters 3, 4, and 5)*	Potential degradation of model accuracy

Chapter 3

Defending Against Data and Model Poisoning Attacks

Federated learning is vulnerable to adversarial attacks, both model and data poisoning, that degrade global model performance. Traditional defenses often fail in high-adversary or heterogeneous data settings.

This chapter introduces Robust Federated Clustering Learning (RFCL), a clustering-based approach that groups similar client updates to isolate malicious contributions, applies adaptive federated averaging to weigh reliable inputs, and utilizes personalized model sharing for heterogeneous data.

Section 5.2 outlines the methodology. It begins with a formal problem definition and the adversarial threat model and then describes RFCL's key components, including clustering-based filtering, multi-centre aggregation, similarity analysis, and meta-learning for aggregation.

Section 5.3 presents an experimental evaluation comparing RFCL with state-of-the-art robust aggregation techniques under various adversarial conditions and Non-IID data distributions.

Section 4.4 concludes the chapter by summarizing the key findings and emphasizing RFCL's effectiveness in mitigating both model and data poisoning attacks while preserving contributions from benign clients.

3.1 Introduction

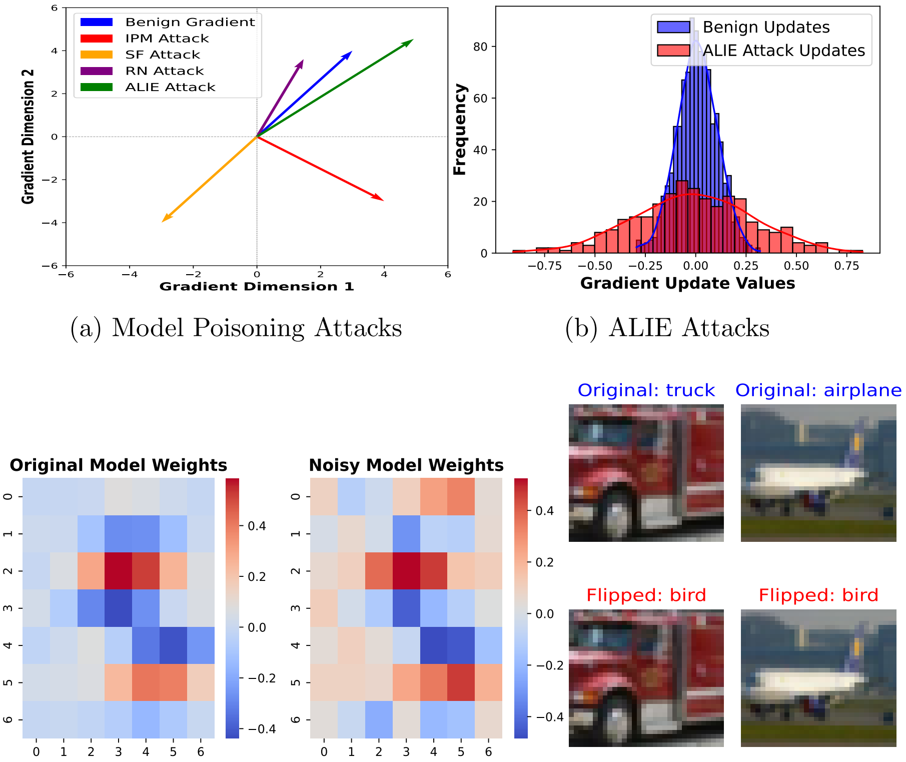
As highlighted in Chapter 2, federated learning presents security vulnerabilities due to its decentralized nature, particularly against poisoning attacks.

This chapter builds on that background by introducing RFCL, a defense designed to counter these vulnerabilities.

The threat posed by well-crafted poisoning attacks has been widely studied in recent works. One of the most effective attacks is the Inner Product Manipulation (IPM) attack, which alters gradient directions while maintaining the same norm as benign updates, making it undetectable by norm-based anomaly detection methods [69]. Similarly, the A Little is Enough (ALIE) attack perturbs gradients within a controlled variance range, ensuring that adversarial updates remain statistically similar to honest updates, thereby bypassing robust aggregation rules [7]. Other attack strategies include the Sign Flipping (SF) attack [28], the Random Noise (RN) attack [48], and the Label Flipping (LF) attack [62], each of which exploits different vulnerabilities to compromise model integrity.

While techniques such as Krum, Median, and Trimmed Mean offer defense, they fail under high adversarial participation [18]. ALIE and IPM are especially difficult for these methods to detect. Furthermore, these defenses often underperform in Non-IID settings. Recent defenses, including Bulyan, AFA [48], FedMGDA+ [54], and Centered Clipping, attempt to address these challenges but face limitations such as high computational cost, fixed thresholds, or poor adaptability to Non-IID data. A detailed comparison with existing methods is shown in Table 3.1.

To overcome these issues, we propose **Robust Federated Clustering Learning (RFCL)**, a novel clustering-based aggregation framework designed to enhance FL security against both data and model poisoning attacks. RFCL applies unsupervised clustering (e.g., HDBSCAN) to identify dense groups of similar client updates, assuming malicious updates form outliers. By measuring cosine similarity and filtering outliers before aggregation, RFCL ensures that only reliable updates influence the global model, improving both security and robustness under



(a) Model Poisoning Attacks

(b) ALIE Attacks

(c) RN Attack

(d) LF Attack

Figure 3.1: FL Under Adversarial Attacks of Model Poisoning and Data Poisoning. (a) Visualization of adversarial gradient manipulations in FL. IPM perturbs gradient directions while maintaining their norm, evading norm-based anomaly detection. SF flips gradient signs, forcing divergence. RN injects stochastic noise, disrupting model convergence. ALIE shifts the mean of gradients while maintaining expected variance. (b) Histogram showing how ALIE blends with benign updates. (c) Heatmap showing instability in RN attack. (d) LF flips true labels to poison the learning process.

heterogeneous data conditions.

Table 3.1: Comparison with existing robust FL aggregation methods.

Method	**Resilience**	**IPM**	**ALIE**	**Non-IID Compatible**
Krum [11]	Fails >30% Attackers	✗	✗	✗
Median [74]	Fails >40% Attackers	✗	✓	✗
Trimmed Mean [74]	Fails >40% Attackers	✗	✓	✗
AFA [48]	Fails >35% Attackers	✗	✗	✓
FedMGDA+ [54]	Fails >50% Attackers	✓	✗	✓
RFCL (Proposed)	**Stable up to 50% Attackers**	✓	✓	✓

The remainder of this chapter describes the RFCL framework in detail, presents empirical results, and discusses potential limitations.

3.2 Methodology

This section presents the Robust Federated Clustering Learning (RFCL) framework, designed to enhance FL security against both data and model poisoning attacks. RFCL integrates dimensionality reduction, clustering, similarity filtering, meta-learning-based aggregation, and personalized model sharing.

3.2.1 Overview of the RFCL Framework

Figure 3.2 illustrates the overall RFCL framework. RFCL introduces a novel, multi-layered defense strategy that fundamentally differs from existing robust aggregation methods. It integrates unsupervised clustering (HDBSCAN), adaptive trust-weighted internal aggregation (ModiAFA), similarity-guided meta-aggregation, and personalized model sharing. Unlike traditional defenses that rely on static thresholds or simple distance-based outlier detection, RFCL performs progressive, structure-aware filtering and aggregation. This modular design enables the framework to accurately isolate and suppress malicious updates, including those that are stealthy or statistically indistinguishable, thereby enhancing robustness in highly adversarial and heterogeneous federated learning environments.

Filtering Layer: High-dimensional client model updates are first preprocessed using Principal Component Analysis (PCA) to reduce dimensionality and noise. The PCA-transformed updates are then clustered using HDBSCAN. This step groups similar updates together while isolating outliers.

Eliminating Layer: Within each cluster, internal aggregation is performed using the Modified Adaptive Federated Averaging (ModiAFA) method to compute a representative cluster center. Cosine similarity analysis is subsequently applied to compare the similarity between the computed cluster centers.

Aggregation Layer: The refined cluster centers are then combined using a meta-learning-based aggregation strategy to produce a robust, concentrated global model. Finally, this global model is personalized by selectively sharing it with clients associated with trusted clusters, ensuring that the final model reflects the underlying data distributions of reliable client groups.

3.2.2 RFCL Process

The RFCL process integrates filtering, eliminating, and aggregation into a unified workflow. The key steps and high-level view of the RFCL process, as outlined in Algorithm 1. At the beginning of each round, the server checks whether it is the first round ($r = 0$). If so, the initial model $\mathbf{M}_0$ is shared with all clients; for subsequent rounds, the server distributes the current cluster centers $\mathbf{M}_{cc}$ to the respective client clusters. After local training, the server collects the updated

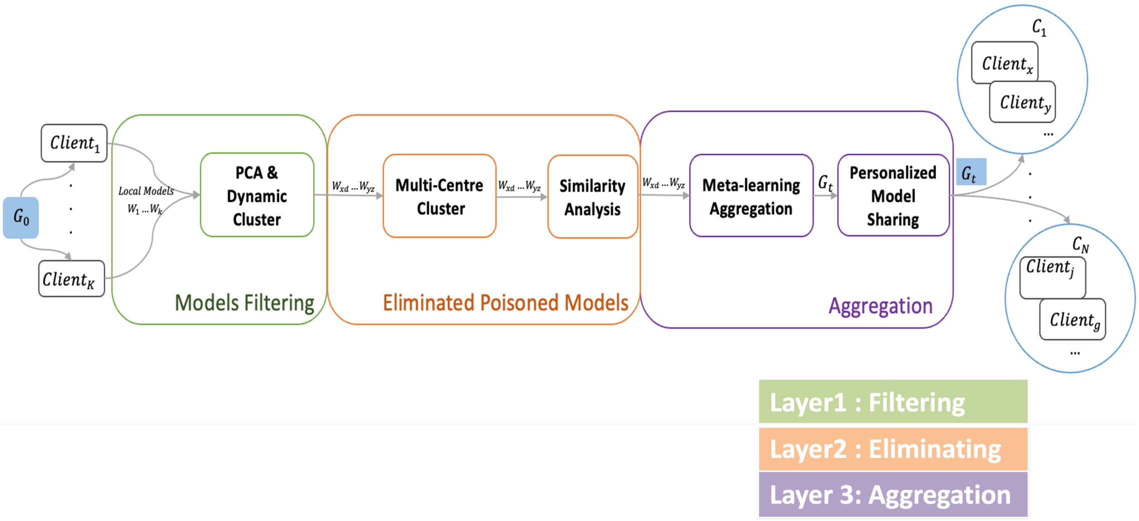


Figure 3.2: Overview of the RFCL framework. The framework comprises the Filtering Layer (PCA-based dimensionality reduction and HDBSCAN clustering), the Eliminating Layer (internal aggregation using ModiAFA with cosine similarity-based refinement and outlier suppression), and the Aggregation Layer (meta-learning-based aggregation and personalized model sharing).

client models $\mathbf{M}_i$ and performs PCA followed by HDBSCAN clustering. An internal aggregation is then conducted within each cluster (using ModiAFA) to generate cluster centers. Next, cosine similarity analysis is applied to select the most similar cluster centers $\mathbf{M}_{best}$, which are aggregated using a meta-learning-based strategy to compute a concentrated model $\mathbf{M}_c$. This concentrated model is used to update the corresponding clusters, and $\mathbf{M}_c$ is set as the new global model $\mathbf{M}_g$. Finally, the server evaluates $\mathbf{M}_g$ on a test dataset D_{test} and records the performance.

3.2.3 Clustering-Based Filtering

Clustering-based filtering is a crucial step in RFCL, enabling the detection and elimination of adversarial updates prior to aggregation. This phase employs both dimensionality reduction and unsupervised clustering.

Dimensionality Reduction via PCA. Deep learning models often contain

Algorithm 1 RFCL Process

```
Require: $M_0$, $N$, $D_{test}$ return $E$, $M_g$
for $r = 0$ to $R - 1$ do
    if $r = 0$ then
        Share $M_0$ with all clients and perform local training.
    else
        Share $M_{cc}$ with associated clients and perform local training.
    end if
    Collect all client models $M_i$.
    Apply PCA and HDBSCAN on $M_i$ to perform clustering.
    Conduct internal aggregation within each cluster (using ModiAFA) to generate cluster centers $M_{cc}$.
    Select the most similar cluster centers $M_{best}$ based on cosine similarity.
    Aggregate $M_{best}$ to compute the concentrated model $M_c$.
    Update the cluster centers in $M_{cc}$ for the selected clusters with $M_c$.
    Set the global model $M_g \leftarrow M_c$.
    Evaluate $M_g$ on $D_{test}$ and record the error $E[r]$.
end for
return $E$, $M_g$
```

millions of parameters, leading to high-dimensional weight vectors that are challenging for clustering. To address this, RFCL applies Principal Component Analysis (PCA), implemented with Singular Value Decomposition (SVD) [26, 10]. Given a dataset of n client updates, each represented by a flattened weight vector $\mathbf{w}_i \in \mathbb{R}^d$, PCA finds the transformation

$$\mathbf{Z} = \mathbf{X}\mathbf{V}_{1:k}, \tag{3.1}$$

where $\mathbf{X} \in \mathbb{R}^{n\times d}$ stacks the weight vectors, $\mathbf{V}_{1:k}$ holds the top k principal components, and $\mathbf{Z} \in \mathbb{R}^{n\times k}$ is the low-dimensional representation. The dimensionality k is chosen so that the cumulative explained variance

$$\rho_k = \frac{\sum_{i=1}^{k} \sigma_i^2}{\sum_{j=1}^{d} \sigma_j^2}, \tag{3.2}$$

satisfies $\rho_k \geq 0.95$.

Unsupervised Clustering via HDBSCAN. Once the updates are projected to $\mathbb{R}^k$, RFCL employs Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) [12, 45] to discover clusters and flag outliers. Unlike K-Means, HDBSCAN chooses the number of clusters automatically and supports variable-density clusters—both critical for heterogeneous FL scenarios. In our implementation we set:

- **min_cluster_size** = 9, preventing over-fragmentation;
- **min_samples** to control outlier sensitivity;
- **metric** = `euclidean`.

After clustering, the algorithm counts the number of clusters, assigns each model to its respective cluster, and computes the cluster centers using the Modified AFA (ModiAFA) method, which adjusts each client's contribution based on cosine similarity. The algorithm 2 illustrate the clustering-based filtering process.

Algorithm 2 Clustering Method

```
Require: $models$ return $M_{cc}$
$X \leftarrow$ extract_weights(models)                      ▷ Extract weights from models
$X \leftarrow$ PCA($X$)                                     ▷ Apply PCA for dimensionality reduction
cluster $\leftarrow$ HDBSCAN($X$, )                         ▷ Apply dynamic cluster
$cluster\_labels \leftarrow$ cluster.labels                 ▷ Retrieve cluster labels
$cluster\_count \leftarrow \max(cluster\_labels) + 1$       ▷ Count the number of clusters
$indices \leftarrow$ [[] for _ in range($cluster\_count$)]
for $i, l \in$ enumerate($cluster\_labels$) do
    if $l \neq -1$ then
        Append index $i$ to $indices[l]$
    end if
end for
for $i, ins \in$ enumerate($indices$) do
    $M_{cc}[i] \leftarrow$ ModiAFA($ins, models$)           ▷ Compute the cluster center using ModiAFA
end for
return $M_{cc}$
```

3.2.4 Multi-Centres Internal Aggregation Method

RFCL leverages a multi-centres internal aggregation approach to address the heterogeneity inherent in federated learning. Rather than computing a single global model by averaging all client updates, RFCL first clusters the model updates into several groups—each representing a subset of clients with similar data distributions. For instance, while a typical federated learning system might produce one global model (as shown on the left in Figure 3.3), RFCL can generate multiple cluster centers (as depicted on the right in Figure 3.3), such as $\boldsymbol{M}_{cc}^{(1)}, \boldsymbol{M}_{cc}^{(2)}$, and $\boldsymbol{M}_{cc}^{(3)}$.

To compute these cluster centers, RFCL employs the **Modified Adaptive Federated Averaging (ModiAFA)** method, which robustly aggregates client updates within each cluster even when some updates are noisy or adversarial. ModiAFA is built upon the foundational principles of the Adaptive Federated Averaging (AFA) approach [48]; however, several key modifications have been introduced. Unlike the AFA, which blocks clients with outlier updates, ModiAFA down-weights the contributions of such updates instead of discarding them entirely.

The computation unfolds in two key phases:

Phase 1: Initial Aggregation and Cosine Similarity Computation. Within each cluster, an initial weighted average is computed to form a temporary cluster center. The weights for each client's update are derived from trust scores, which are computed using parameters α and β. Each client is initially assigned $\alpha = 1$ and $\beta = 1$. Then, over rounds: If a client's update is consistent with benign behavior, its α is incremented. If a client's update is flagged as outlier or adversarial, its β is incremented. The trust score is then computed using Eq. 3.3:

$$\text{TrustScore} = \frac{\alpha}{\beta}, \tag{3.3}$$

A higher trust score indicates a more reliable client. These trust scores are used as weights in the aggregation, so the temporary cluster center is given by Eq. (3.4):

$$\mathbf{C}_{\text{temporary}} = \frac{\sum_{i \in \text{cluster}} \text{TrustScore}_i \cdot \Delta \mathbf{w}_i}{\sum_{i \in \text{cluster}} \text{TrustScore}_i}, \tag{3.4}$$

Once this temporary center is obtained, the cosine similarity between each client update and the temporary center is calculated. Cosine similarity, which ranges from -1 to 1, quantifies the alignment between the update and the center; a higher value indicates that the update is consistent with the cluster consensus.

Phase 2: Statistical Refinement and Threshold-Based Weight Adjustment. In this phase, the similarity scores from Phase 1 are statistically analyzed. For each cluster, key statistics—mean ($\hat{\mu}_s$), median ($\bar{\mu}_s$), and standard deviation (σ_s)—of the cosine similarity scores are computed. These statistics capture the central tendency and variability of the alignment within the cluster.

A dynamic threshold (Eq. 3.5) is then determined using a parameter ξ:

$$\text{Threshold} = \begin{cases} \bar{\mu}_s - \xi\, \sigma_s, & \text{if } \hat{\mu}_s < \bar{\mu}_s, \\ \\ \bar{\mu}_s + \xi\, \sigma_s, & \text{otherwise.} \end{cases} \tag{3.5}$$

Here, ξ is a parameter that can be adjusted to regulate the sensitivity of the outlier detection process.

Any client update whose cosine similarity s_i deviates significantly from the central tendency (i.e., falls below this threshold) is considered an outlier. Rather than completely blocking such updates, ModiAFA down-weights their contributions—often setting their weight to zero—so that the internal aggregation relies predominantly on updates that align with the cluster consensus.

After this refinement, the remaining trusted updates are re-normalized and aggregated to produce the robust cluster center $\mathbf{M}_{cc}$. At the end of the process, the trust parameters α and β are updated based on the outcome, ensuring that future rounds reflect the reliability of each client's updates.

3.2.5 Similarity Analysis Method

To further refine the clustered updates, RFCL employs a similarity analysis method to select the most representative cluster centers from the set $\mathbf{M}_{cc}$. The goal is to identify clusters that are highly similar to each other under the assumption that

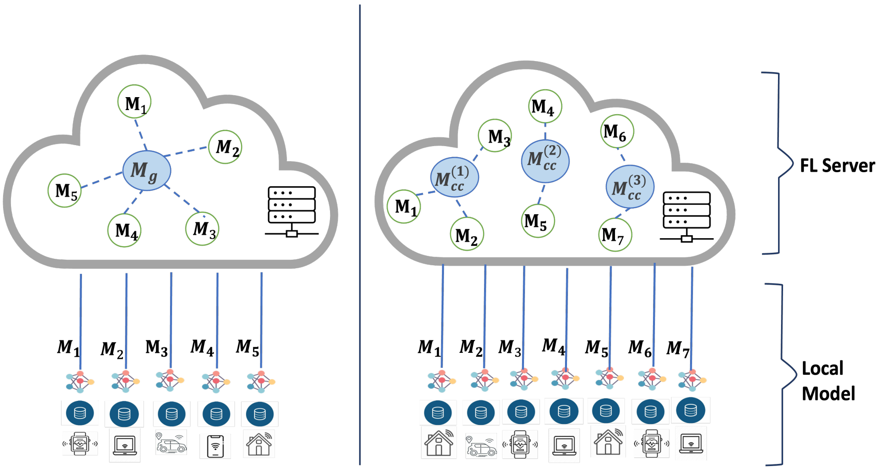


Figure 3.3: Comparison of typical FL single-centre aggregation (left) versus RFCL's multi-centre aggregation (right). RFCL clusters similar updates before aggregation, thereby enhancing robustness against adversarial manipulation.

such clusters are more likely to contain benign and consistent updates. This process is parameterized by K, which determines the number of center clusters to select, and uses cosine similarity as the measure of alignment between cluster centers.

A weight matrix $\mathbf{X}$ is generated from the cluster centers $\mathbf{M}_{cc}$ by flattening each model's parameters into a vector. Then, using cosine similarity, the similarity between every pair of these vectors is computed. The results are stored in a similarity matrix `sims`. A high cosine similarity value (close to 1) indicates that the corresponding cluster centers are well-aligned. For example, if we have five cluster centers, the algorithm computes the cosine similarity among all five pairs to form a 5×5 matrix.

For each cluster center, the algorithm identifies the indices corresponding to the top K largest similarity scores. The similarity scores at these indices are summed to produce a total similarity value for that cluster. The cluster (or clusters) with the highest total similarity is then chosen. These indices are stored as `indices_best`, and the corresponding cluster centers form the set $\mathbf{M}_{\text{best}}$.

Once the best clusters have been selected, their similarity scores are normalized to produce selection probabilities $\mathbf{p}_s$. This is done by dividing each similarity score by the total similarity of the selected clusters. Furthermore, these probabilities are adjusted based on the size of each cluster (i.e., the number of client updates in that cluster) so that larger clusters have a proportionally greater influence in the final aggregation. The method then returns the selected cluster centers $\mathbf{M}_{\text{best}}$, their normalized probabilities $\mathbf{p}_s$, and the corresponding indices `indices_best`. The explanation of this similarity analysis procedure is provided in Algorithm 3.

3.2.6 Meta-Learning for External Aggregation Method

After identifying the most representative cluster centers from the set $\mathbf{M}_{cc}$ through similarity analysis, RFCL employs a meta-learning-based external aggregation method to compute a concentrated global model. This method combines the selected cluster centers, each weighted according to its reliability, to form the global model.

First, the similarity analysis returns the set of best cluster centers $\mathbf{M}_{\text{best}}$ together with raw probabilities $\mathbf{p}_s$ and their indices. These probabilities reflect both the similarity among cluster centers and the size of each cluster; consequently, larger clusters receive higher weights. Before aggregation, the probabilities are normalized so that they sum to one, ensuring that every selected cluster center's contribution is properly scaled.

The concentrated global model is then obtained as a reliability-weighted mean of the selected cluster centers:

$$\mathbf{M}_c = \sum_{i \in \texttt{indices}_{\text{best}}} \texttt{conc\_p}_s[i]\, \mathbf{M}_{\text{best}}[i], \tag{3.6}$$

where each weight $\texttt{conc\_p}_s[i]$ satisfies $\sum_i \texttt{conc\_p}_s[i] = 1$.

Equation 3.6 mirrors the standard FedAvg [46] strategy, which forms a global model by averaging client updates in proportion to their local data volumes. RFCL adopts the same rationale but substitutes those data-size weights with reliability-aware probabilities $\texttt{conc\_p}_s[i]$; cluster centers that are both larger and more trustworthy therefore exert greater influence on the aggregated model.

Algorithm 3 Similarity Analysis

```
Require: $M_{cc}$, $K$ return $M_{best}, p_s, indices_{best}$
$X \leftarrow$ _generate_weights($M_{cc}$)    ▷ Flatten each cluster center into a vector
Initialize an empty list $sims = []$
for each $m_1$ in $X$ do
    Initialize an empty list $sim = []$
    for each $m_2$ in $X$ do
        Compute $sim \leftarrow \text{cosine\_similarity}(m_1, m_2)$
        Append $sim$ to $sim$
    end for
    Append $sim$ to $sims$
end for
Initialize $best\_indices = []$ and $best\_val = 0$
for each set of similarity scores $s$ in $sims$ with index $i$ do
    Determine $indices$: indices of the $K$ largest values in $s$
    Compute $val \leftarrow \sum_{j \in indices} s[j]$
    if $val > best\_val$ then
        $best\_val \leftarrow val$
        $indices_{best} \leftarrow indices$
    end if
end for
Normalize the similarity scores for the selected indices:
        $p_{s_i} = \frac{s_i}{\sum_{j \in indices_{best}} s_j} \quad \text{for each } i \in indices_{best}$
$M_{best} \leftarrow [M_{cc}[i] \text{ for each } i \in indices_{best}]$
Adjust the probabilities based on the size of each cluster:
        $p_{s_i} = \frac{p_{s_i} \cdot \text{len}(cluster\_centre_i)}{\sum_{j \in indices_{best}} p_{s_j} \cdot \text{len}(cluster\_centre_j)}$
return $M_{best}, p_s, indices_{best}$
```

3.2.7 Personalized Model Sharing

RFCL further refines the aggregation process by employing personalized model sharing. Instead of distributing the aggregated model $\mathbf{M}_c$ uniformly to all clients, the server selectively shares $\mathbf{M}_c$ only with those clients that are associated with the most trustworthy clusters (i.e., the selected cluster centers $\mathbf{M}_{\text{best}}$). This selective distribution ensures that each client receives a model that is better tailored to its specific data distribution. In contrast, clients that belong to unselected clusters retain their local cluster centers $\mathbf{M}_{cc}$, thereby preserving their unique characteristics and limiting the influence of potentially unreliable updates.

Algorithm 4 Personalization (Cluster-based Model Sharing)

```
for $i = 0$ to len($M_{cc}$) $- 1$ do
    if $i \in indices_{best}$ then
        $M_{cc}[i] \leftarrow M_c$
    end if
end for
```

In summary, RFCL integrates several techniques—filtering via PCA and HDBSCAN, outlier elimination and similarity analysis with ModiAFA, meta-learning-based external aggregation, and personalized model sharing—into a comprehensive framework. This approach may ensures that only trustworthy client updates contribute to the final global model, significantly enhancing the security and reliability of federated learning in heterogeneous, adversarial environments. Notably, the aggregated model $\mathbf{M}_c$ is shared exclusively with clients in the selected best clusters, while those in other clusters maintain their original cluster centers.

3.3 Experiments

In this section, we evaluate the performance of the proposed RFCL method on image classification tasks using three public datasets. The effectiveness of

RFCL is compared against six baseline robust aggregation methods. The RFCL implementation is available on GitHub[1].

3.3.1 Datasets and Models

We evaluate our defence on three standard image-classification benchmarks: **MNIST** [33], **Fashion-MNIST** [67], and **CIFAR-10** [32]. Together they span a broad range of visual complexity:

- **MNIST** (28×28, grayscale digits): a lightweight, low-noise dataset that serves as a sanity check; results here establish a lower bound on robustness.
- **Fashion-MNIST** (same resolution, clothing images): introduces richer textures and intra-class variation while retaining MNIST's modest computational footprint.
- **CIFAR-10** (32×32, RGB): real-world objects with colour channels and background clutter, providing a more demanding and practically relevant setting.

We use small, fully connected networks for all three datasets (Table 3.2) rather than heavyweight CNNs so that (i) training remains feasible on all 30 clients without GPU acceleration and (ii) any performance difference can be attributed to the aggregation rule, not to model capacity or architectural tweaks. Keeping the architecture identical across baselines eliminates confounding variables and makes robustness comparisons fair and transparent.

3.3.2 Non-IID Degree

Non-IID data distributions play a crucial role in our evaluation as they mimic the realistic variations found across clients in FL environments. In our experiments, we

[1] https://github.com/EbtisaamCS/RFCL

MNIST and Fashion-MNIST Model Architecture
DNN (784 × 512 × 256 × 10) with 2 hidden layers
Activation functions: Leaky ReLU
Batch size: 64, Loss: Cross-Entropy
Optimizer: SGD (learning rate = 0.1), Dropout: $p = 0.5$
CIFAR-10 Model Architecture
DNN (3072 × 256 × 128 × 10) with 2 hidden layers
Activation functions: Leaky ReLU
Batch size: 128, Loss: Cross-Entropy
Optimizer: SGD (learning rate = 0.5)

Table 3.2: Models and training parameters for the experiments.

simulate Non-IID conditions using a Dirichlet distribution to partition the training data among 30 clients. A higher Dirichlet parameter value, such as $\alpha = 0.9$, results in a scenario where class distributions and data quantities are relatively similar across clients, producing a slightly Non-IID split characterized by low variance. In contrast, a lower parameter value, such as $\alpha = 0.1$, generates a highly Non-IID environment where the variance among client data distributions is significantly increased. This simulation allows us to rigorously examine the performance of aggregation methods when confronted with the challenges posed by heterogeneous data distributions.

3.3.3 Number of Attackers

The impact of adversarial interference is critical to understanding the robustness of federated learning systems. All experiments assume 30 clients per communication round. We vary the number of adversarial participants across six scenarios (3, 6, 9, 12, 15, and 18 attackers per round), corresponding to attacker ratios of 10%, 20%, 30%, 40%, 50%, and 60% of the total client set, respectively. This systematic sweep pinpoints where conventional aggregation begins to degrade and shows how RFCL preserves stability even as the attacker ratio increases.

3.3.4 Threat Model

We consider a practical federated learning setup where some clients may behave maliciously by launching model or data poisoning attacks. The server is honest but untrusted; it does not know which clients are adversarial and relies entirely on the defense mechanism to identify and mitigate threats. Attackers operate independently and have no knowledge of other clients' updates, but they can craft updates that closely resemble benign behavior to avoid detection.

To test RFCL under diverse adversarial conditions, we include five representative attack types:

- **IPM** [69] and **ALIE** [7] are stealthy model poisoning attacks that mimic the statistical patterns of benign gradients to evade norm- or distance-based defenses.
- **SF** [28] and **RN** [48] are aggressive attacks that introduce large gradient deviations to disrupt training.
- **LF** [62] is a data poisoning attack that corrupts labels during local training, degrading the global model without modifying the update mechanics.

These threat models were chosen to reflect both subtle and disruptive attack strategies, ensuring RFCL is evaluated under realistic and challenging scenarios.

3.3.5 Comparison Methods

A thorough evaluation requires a comprehensive comparison with existing robust aggregation techniques. In our experimental study, RFCL is benchmarked against several prominent methods to assess its effectiveness in mitigating adversarial attacks in federated learning. The comparison includes MKrum [11], Median [74], Adaptive Federated Averaging (AFA) [48], FedMGDA+ [25], and Centered Clipping (CC) [28].

Additionally, we include the standard Federated Averaging (FedAvg) [46] as a baseline to illustrate the performance improvements achieved by robust aggregation techniques. This multi-faceted comparison highlights the relative strengths and weaknesses of each method, providing a clear context for the advances introduced by RFCL in terms of robustness, efficiency, and overall model utility in diverse and adversarial federated learning environments.

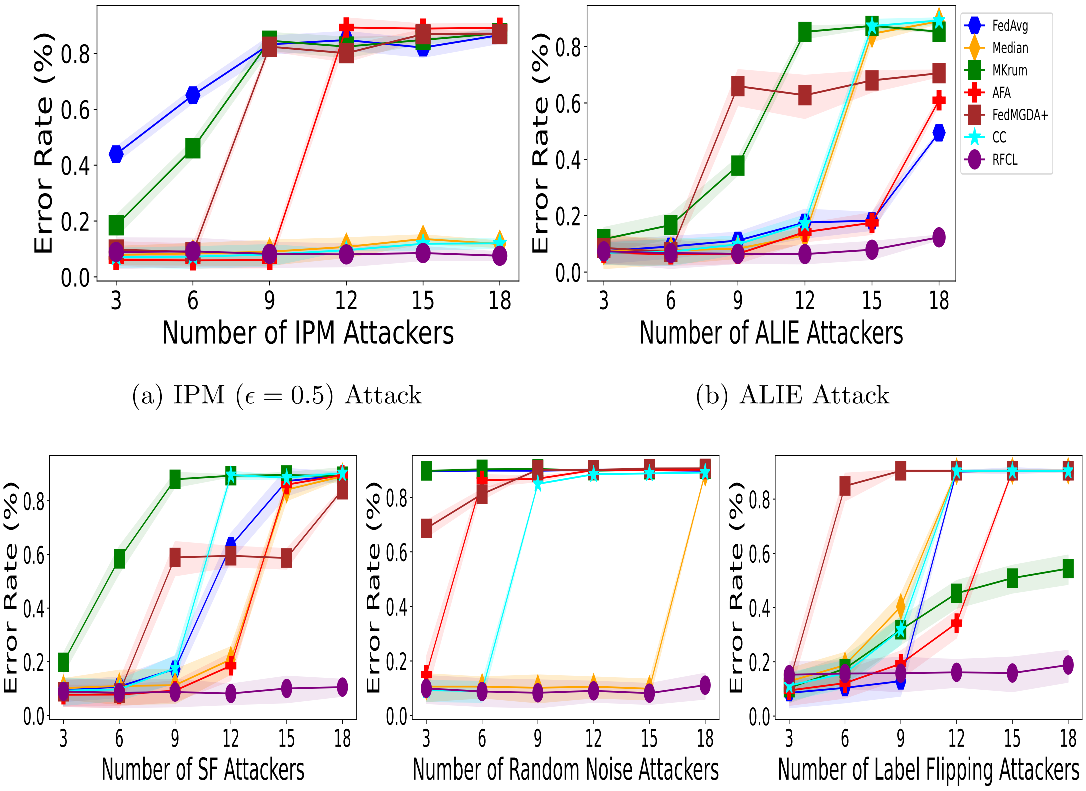
3.3.6 Experiment Results

We evaluate the performance of RFCL and the baseline methods on each dataset under various attack scenarios and degrees of Non-IID data. All experiments are repeated over five independent runs, and the average results are reported. In all plots, error bars represent a confidence interval of $\rho = 0.01$.

Figure 3.4 shows the performance of different methods on the MNIST dataset under a Non-IID setting with $\alpha = 0.5$. Under IPM, ALIE, SF, RN, and LF attacks, RFCL consistently achieves a lower error rate compared to the other methods. For example, when the number of IPM attackers is 3 or 6, RFCL, Median, AFA, FedMGDA+, and CC maintain low error rates, while FedAvg and MKrum experience a significant accuracy drop. As the number of attackers increases, RFCL remains robust.

Figure 3.5 illustrates the impact of increasing the perturbation magnitude ϵ in IPM attacks. As ϵ increases from 0.5 to 100.0, the perturbations applied to the gradients become more severe. This results in more drastic changes in the model parameters, negatively affecting the performance of most aggregation methods. RFCL, however, exhibits robust performance even under higher perturbation magnitudes.

Figure 3.6 shows the performance of the aggregation methods on the CIFAR-10 dataset under a Non-IID setting with $\alpha = 0.5$. RFCL achieves the lowest error rate under various attack types, including IPM, ALIE, SF, RN, and LF, particularly when the number of attackers increases.



(a) IPM ($\epsilon = 0.5$) Attack

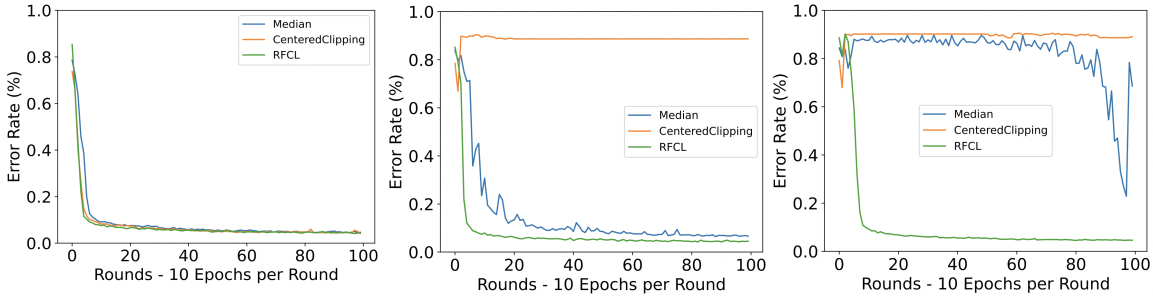
(b) ALIE Attack

(c) SF Attack

(d) RN Attack

(e) LF Attack

Figure 3.4: Performance comparison of FedAvg, Median, MKrum, AFA, FedMGDA+, CC, and RFCL on the MNIST dataset under a Non-IID ($\alpha = 0.5$) scenario. Each method is evaluated with 3, 6, 9, 12, 15, and 18 malicious clients per round, corresponding to attacker ratios from 10% to 60% of the 30 total clients, across five attack types (IPM, ALIE, SF, RN, and LF).



(a) IPM Attackers (b) 12 IPM Attackers (c) 18 IPM Attackers

Figure 3.5: Comparison of each round's performance of Median, CC, and RFCL on MNIST under a Non-IID ($\alpha = 0.5$) scenario with varying numbers of IPM ($\epsilon = 100.0$) attackers.

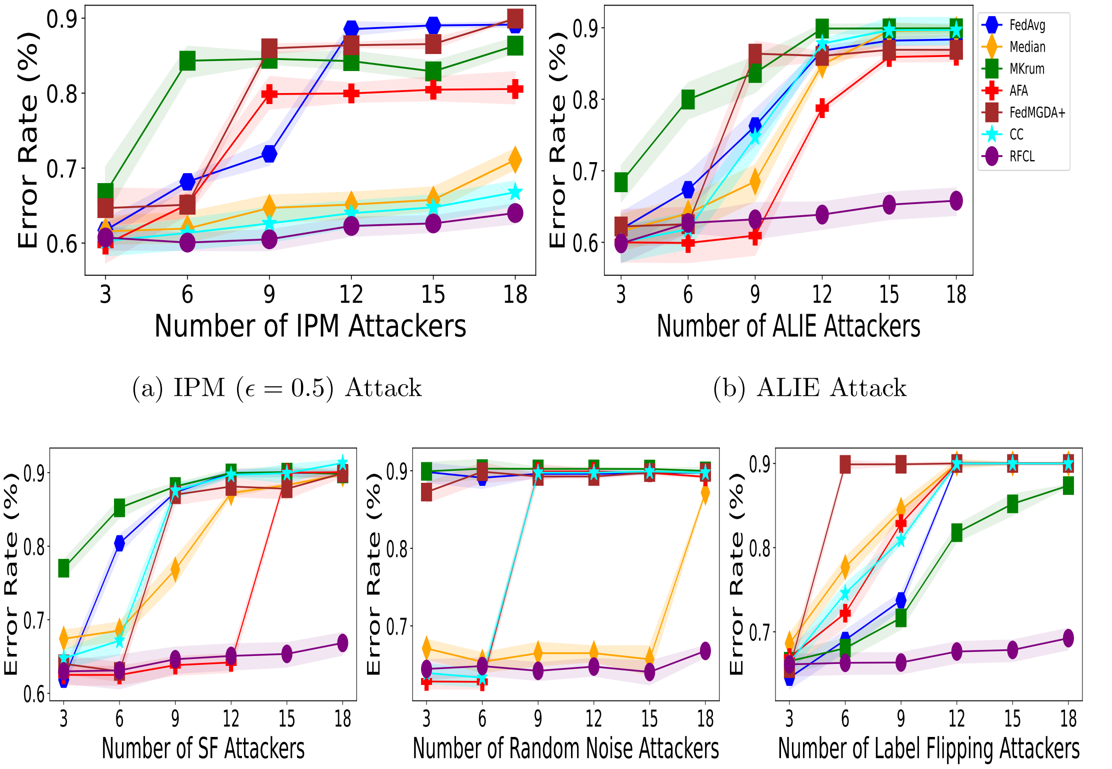
Figure 3.7 provides a round-by-round performance comparison of the methods on CIFAR-10 under scenarios with 6, 12, and 18 ALIE and LF attackers. RFCL maintains the lowest error rate, indicating that its clustering and personalized model-sharing mechanisms help reduce the impact of adversarial attacks even as their number increases.

Figures 3.8 and 3.9 demonstrate the performance of the methods on the Fashion-MNIST dataset under slightly Non-IID ($\alpha = 0.5$) and extremely Non-IID ($\alpha = 0.1$) scenarios, respectively. In both cases, RFCL achieves the lowest test error rate compared to the baseline robust aggregation methods, indicating its effectiveness in handling heterogeneous data distributions and varying numbers of attackers.

3.3.7 Different Clustering Methods

In addition to our primary configuration using HDBSCAN, we further evaluated the robustness of RFCL by exploring alternative clustering methods, such as Agglomerative clustering and K-Means. Figure 3.10 illustrates a comparative analysis of these methods under IPM and ALIE attacks on the CIFAR-10 dataset.

K-Means clustering, while widely used, requires predefining the number of clusters. In our experiments, we set the number of K-Means clusters to be five



(a) IPM ($\epsilon = 0.5$) Attack

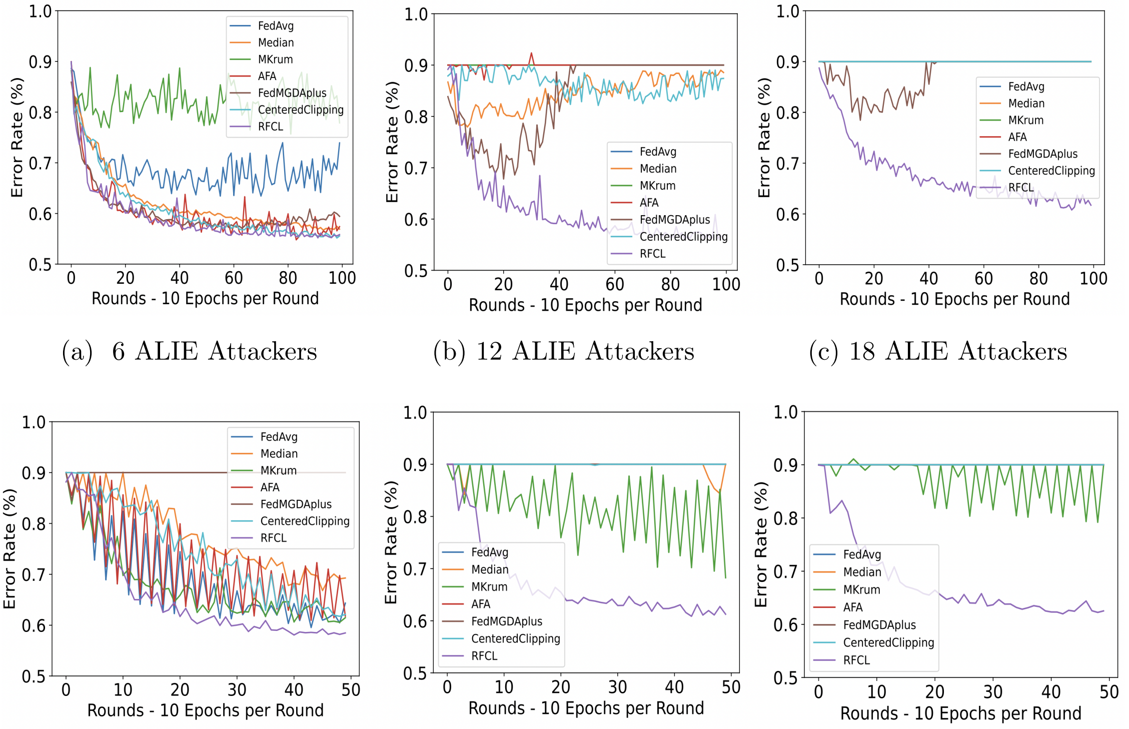
(b) ALIE Attack

(c) SF Attack

(d) RN Attack

(e) LF Attack

Figure 3.6: Performance comparison of FedAvg, Median, MKrum, AFA, FedMGDA+, CC, and RFCL on the CIFAR-10 dataset under a Non-IID ($\alpha = 0.5$) scenario, evaluated against various numbers of IPM, ALIE, SF, RN, and LF attackers.



(a) 6 ALIE Attackers

(b) 12 ALIE Attackers

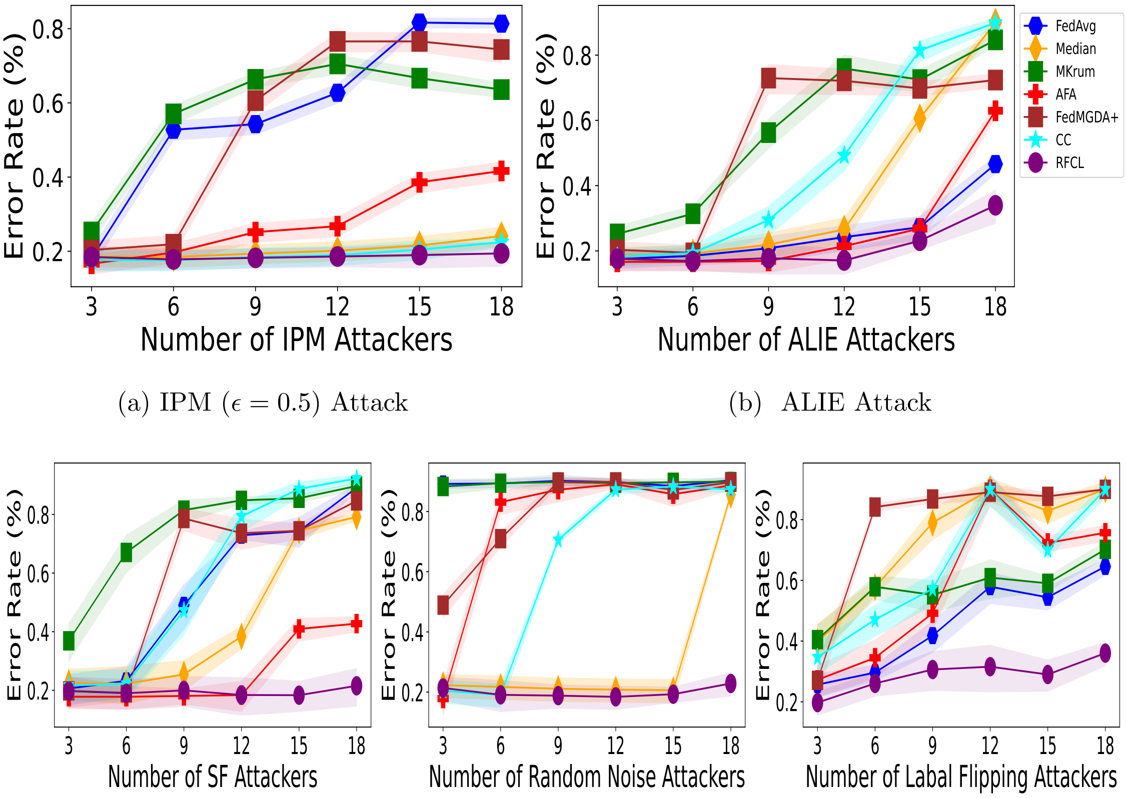
(c) 18 ALIE Attackers

(d) 6 LF Attackers

(e) 12 LF Attackers

(f) 18 LF Attackers

Figure 3.7: Comparison of round-by-round performance for FedAvg, Median, MKrum, AFA, FedMGDA+, CC, and RFCL on CIFAR-10 under a Non-IID ($\alpha = 0.5$) scenario with different numbers of ALIE and LF attackers.



(a) IPM ($\epsilon = 0.5$) Attack

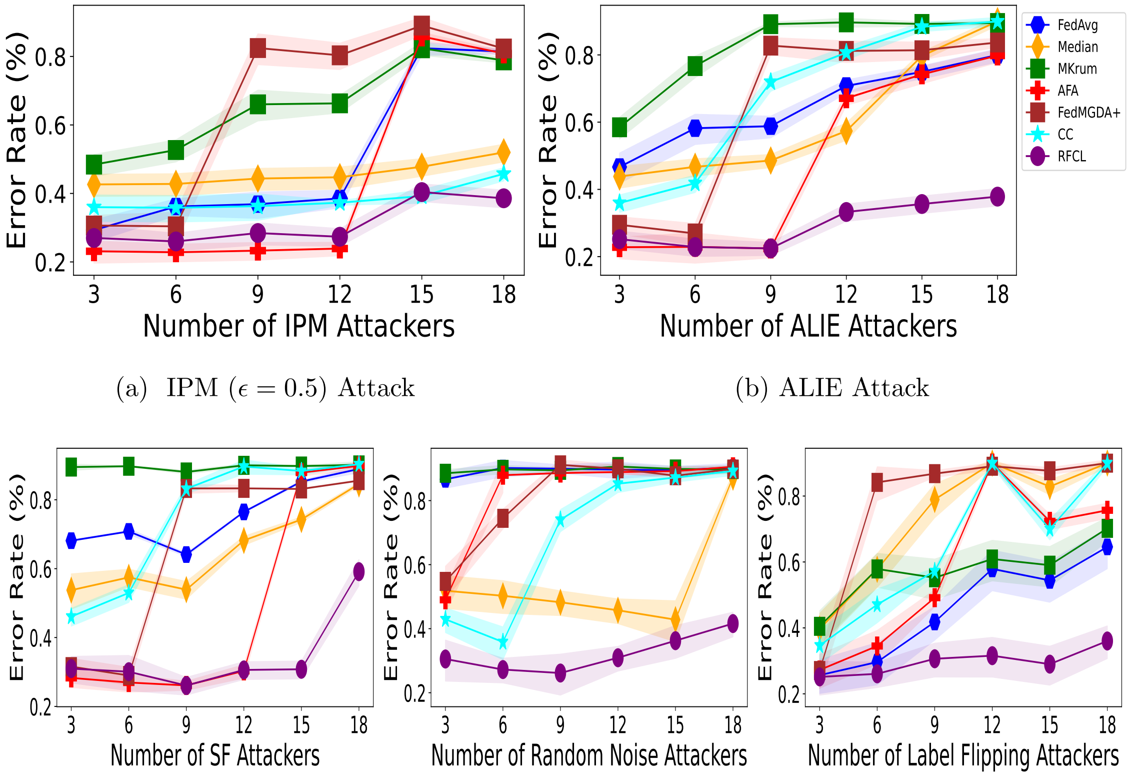
(b) ALIE Attack

(c) SF Attack

(d) RN Attack

(e) LF Attack

Figure 3.8: Performance comparison of FedAvg, Median, MKrum, AFA, FedMGDA+, CC, and RFCL on Fashion-MNIST under a Non-IID ($\alpha = 0.5$) scenario.



(a) IPM ($\epsilon = 0.5$) Attack

(b) ALIE Attack

(c) SF Attack

(d) RN Attack

(e) LF Attack

Figure 3.9: Performance comparison of methods on Fashion-MNIST under an extremely Non-IID ($\alpha = 0.1$) scenario.

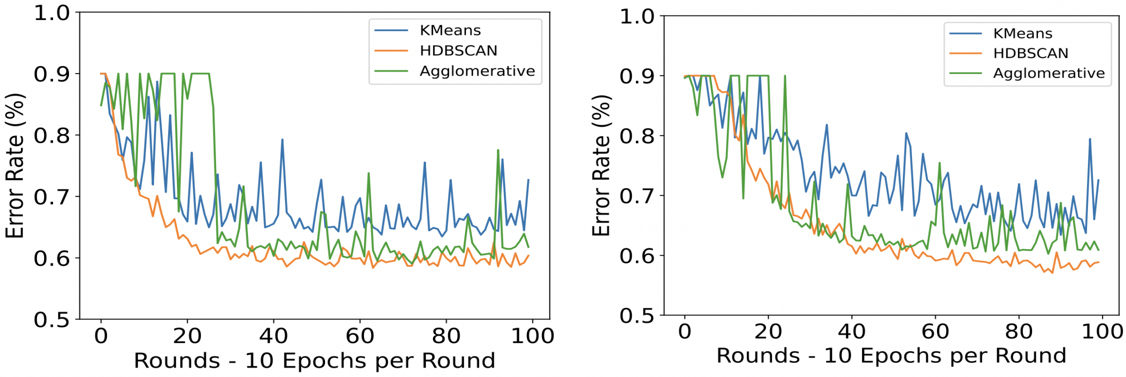
more than the number of attackers (i.e., $C = M + 5$) to ensure comprehensive data analysis. However, this constraint may limit its flexibility compared to density-based methods.

Agglomerative clustering operates by treating each data point as an individual cluster and gradually merging them based on similarity [12]. Although this hierarchical method provides useful visual insights into the data structure, its computational complexity can make it less suitable for larger datasets.

HDBSCAN, a density-based method, is particularly well-suited for our application because it does not require pre-specifying the number of clusters and can automatically detect clusters of varying densities. Parameter tuning for HDBSCAN, such as setting `min_cluster_size` and `min_samples`, relies on domain knowledge and empirical observations. For example, for scenarios with fewer than 12 attackers, we found that setting `min_cluster_size` between 24 and 18 and `min_samples` between 6 and 12 works well. In cases with a higher proportion of attackers (e.g., 15 or 18), adjusting `min_cluster_size` to between 12 and 9 and `min_samples` to between 18 and 21 yields better results. Based on our experiments, HDBSCAN provided the most reliable clustering performance.

3.3.8 Ablation Study

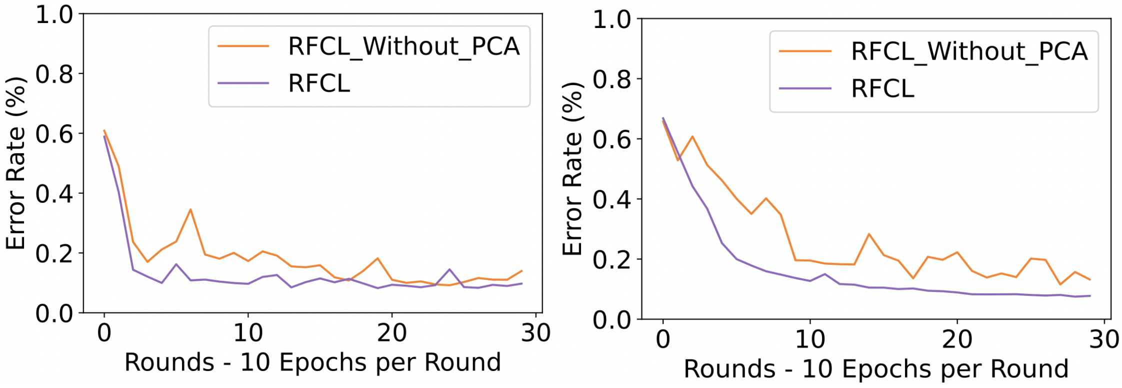
To evaluate the contribution of each component of RFCL, we conducted ablation studies focusing on the impact of the PCA step. Figure 3.11 compares the performance of RFCL with and without PCA under Random Noise (RN) and Label Flipping (LF) attacks. The results indicate that while RFCL maintains a degree of robustness even without PCA, the inclusion of PCA slightly improves overall performance by reducing noise and enhancing clustering quality. This suggests that, although not critical, PCA contributes positively to the resilience and effectiveness of the RFCL method.



(a) 18 IPM Attackers

(b) 18 ALIE Attackers

Figure 3.10: CIFAR-10 error rates under 18 IPM (left) and 18 ALIE (right) attackers. The proposed method—RFCL with HDBSCAN (orange curve)—consistently achieves the lowest error, outperforming K-Means and Agglomerative.



(a) RN Attack

(b) LF Attack

Figure 3.11: Ablation study comparing RFCL performance with and without the PCA step under RN and LF attacks.

3.3.9 Discussion and Limitations

In this work, we proposed RFCL, a robust federated learning framework that leverages clustering-based filtering, similarity analysis, and meta-learning-based aggregation with personalized model sharing. Our experiments on MNIST, CIFAR-10, and Fashion-MNIST demonstrate that RFCL consistently outperforms several baseline robust aggregation methods under various adversarial attack scenarios and Non-IID data distributions.

While RFCL shows significant improvements, there are several limitations that warrant further investigation. First, the clustering phase—which uses PCA for dimensionality reduction followed by HDBSCAN—can be computationally intensive, especially as the number of clients increases. The performance of RFCL is sensitive to the parameter settings of HDBSCAN (e.g., `min_cluster_size` and `min_samples`), which require careful tuning based on the specific data distribution and expected attack scenarios. Additionally, although the personalized model sharing mechanism ensures that clients receive models aligned with their local data distributions, it also introduces additional complexity in maintaining consistency across different clusters. In extremely heterogeneous environments or when the proportion of adversarial clients becomes very high, the effectiveness of the current trust score updating and similarity-based selection may be further challenged.

Future work should consider exploring adaptive parameter tuning and more computationally efficient clustering algorithms, as well as investigating the scalability of RFCL in larger and more diverse federated settings.

3.4 Conclusion

In this chapter, we introduced RFCL, a novel robust federated learning framework designed to mitigate data and model poisoning attacks in heterogeneous environments. RFCL combines multiple techniques: it uses PCA and HDBSCAN for filtering client updates, employs the Modified Adaptive Federated Averaging

(ModiAFA) method—with cosine similarity and statistical refinement—to compute robust cluster centers, and integrates a meta-learning-based external aggregation strategy along with personalized model sharing to form a concentrated global model.

Experimental results on MNIST, CIFAR-10, and Fashion-MNIST datasets indicate that RFCL achieves lower error rates and improved robustness compared to state-of-the-art aggregation methods, even in the presence of various adversarial attacks and under Non-IID data distributions. While the proposed framework demonstrates significant improvements, challenges such as computational complexity, parameter sensitivity, and scalability in extreme heterogeneous settings remain. Future work will focus on adaptive parameter tuning, optimizing the clustering process, and exploring more advanced personalization strategies to further enhance RFCL's performance.

Overall, RFCL represents a promising step towards more secure and reliable federated learning systems, providing a comprehensive solution that integrates robust aggregation with tailored model sharing.

Chapter 4

Counteracting Backdoor Attacks

This chapter introduces Robust Knowledge Distillation (RKD), a three-stage defence consisting of automated clustering, median-based selection and knowledge distillation to filter out poisoned client updates and neutralize backdoor attacks in federated learning.

Section 5.2 formalizes the backdoor problem and threat model, then details each RKD stage: HDBSCAN clustering to spot outliers, coordinate-wise median selection to resist extremes and distillation into a clean global model.

Section 5.3 benchmarks RKD against A3FL, F3BA, DBA, ADBA and TSBA under varying data heterogeneity; analyses the impact of HDBSCAN's cluster-size; measures runtime scalability; and conducts an ablation study on each RKD module.

Section 4.4 summarizes that RKD effectively suppresses backdoors without degrading main-task accuracy and outlines future research directions.

4.1 Introduction

The previous chapter tackled model-poisoning threats in federated learning. Here, we turn to the more subtle problem of backdoor attacks, in which adversarial clients embed hidden trigger patterns so that the global model behaves normally on clean data yet misclassifies inputs containing the trigger.

Recent work shows that even a minority of colluding clients can insert highly effective backdoors. Distributed Backdoor Attack (DBA) [68] splits a trigger across several participants; Adversarially Adaptive Backdoor Attacks (A3FL) [76] adapts its trigger with projected-gradient steps; and Focused-Flip Federated Backdoor Attacks (F3BA) [18] flips a targeted subset of weights. Detecting such distributed, low-magnitude changes is extremely challenging.

Figure 4.1 illustrates the characteristic effects of these backdoor attacks on data and model activations, highlighting the subtle yet distinctive patterns produced by DBA, A3FL, and F3BA. These visualizations underscore the complexity of detecting covert triggers in a heterogeneous FL environment.

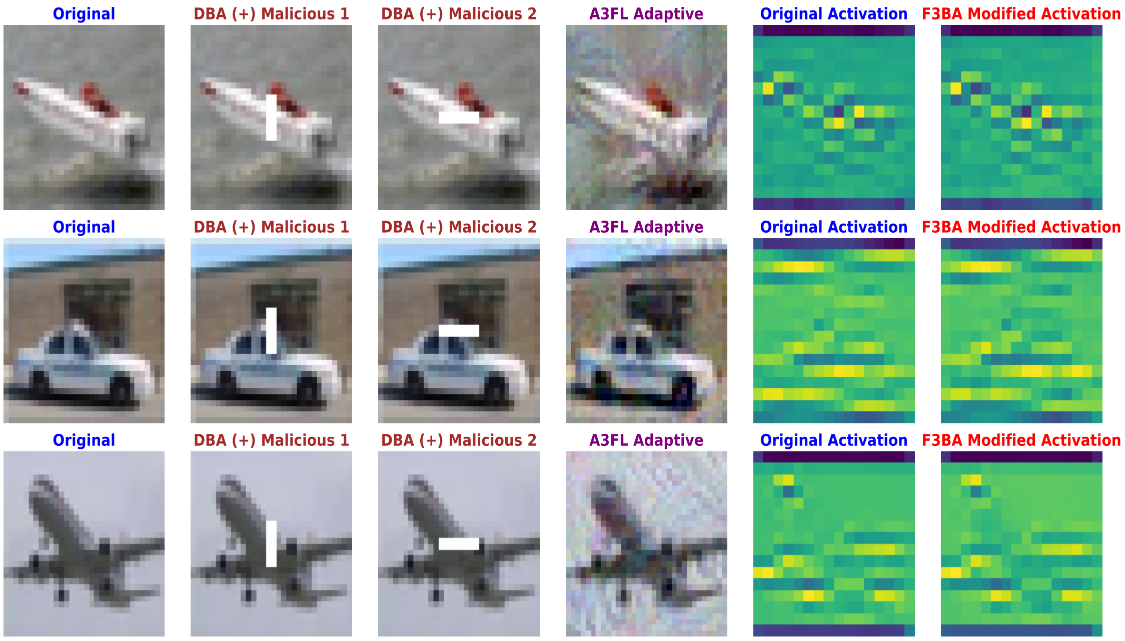


Figure 4.1: Visualizations of backdoor attack effects. From left to right: (1) Original input image (clean, no attack), (2–3) Images with DBA malicious triggers, (4) Image with A3FL adaptive trigger, (5) Activation map for clean input (benign behavior), and (6) Activation map under F3BA (malicious manipulation).

Traditional robust aggregators (Krum, Median, Trimmed Mean) assume IID data or a small adversary fraction and miss stealth triggers. More recent methods—e.g. RLR [52], FoolsGold [20], FLAME [50]—either penalize benign clients in heterogeneous settings or require heavy hyper-parameter tuning in high-dimensional spaces. Table 4.1 offers a comparative overview of existing backdoor defense approaches along with the proposed RKD method in FL.

Table 4.1: Comparison with existing backdoor defense approaches in FL.

Method	**Key Assumptions**	**Backdoor Robustness**	**Works on Non-IID**
FLTrust [52]	Trusted data available; low adversary fraction	Moderate; vulnerable to adaptive attacks	Limited
Foolsgold [20]	Assumes adversaries produce nearly identical gradients	Moderate; may penalize benign clients in heterogeneous settings	Poor
FedRAD [60]	Majority of clients are honest	Moderate (stable up to 40% attackers)	Moderate
FedDF [37]	Primarily addresses data heterogeneity	Limited backdoor defense capability	High
FedBE [15]	Focuses on mitigating heterogeneity effects	Limited backdoor defense capability	High
RLR [52]	Assumes significant deviation in malicious updates	Limited; struggles with adaptive attacks	Poor
RKD	**No strict IID or low adversary fraction**	**High (stable up to 50% attackers)**	**Robust**

To address these challenges, we propose **Robust Knowledge Distillation (RKD)**, a novel defense mechanism that specifically targets backdoor attacks in FL without relying on strict IID or low adversary assumptions. RKD integrates clustering and median model selection techniques to filter out malicious updates.

By computing cosine similarity scores between client updates and the global model, RKD transforms the high-dimensional parameter space into a one-dimensional representation that captures directional alignment. This simplified representation enables efficient anomaly detection using HDBSCAN. A robust median model selection process subsequently identifies a representative ensemble of benign models, and knowledge is distilled from this ensemble to securely update the global model. In doing so, RKD effectively mitigates backdoor attacks even in scenarios with up to 50% adversarial participation, while also works on Non-IID data distributions.

The remainder of this chapter presents the RKD methodology, analyzes its empirical evaluation results, and discusses its limitations.

4.2 Methodology

In this section, we introduce the Robust Knowledge Distillation **(RKD)** framework, designed to secure federated learning against backdoor attacks by identifying and mitigating malicious model updates. RKD consists of three core components—Automated Clustering, Model Selection, and a Knowledge Distillation Module—that work in concert to detect and eliminate backdoor attacks while preserving the performance and integrity of the global model.

4.2.1 Overview of the RKD Framework

The proposed RKD framework employs a multi-tiered strategy to enhance the robustness of federated learning systems. Initially, the central server initializes the global model $\boldsymbol{M}^0_{\text{global}}$ and broadcasts it to all participating clients. Each client i then trains its local model $\boldsymbol{M}^r_i$ on its private dataset D_i, starting from the current global model $\boldsymbol{M}^r_{\text{global}}$, and returns its updated model weights $\mathbf{w}^r_i$ to the server.

At the server, the first step is to identify potential malicious updates. To do so, the server computes the cosine similarity between each client's update $\mathbf{w}^r_i$ and the current global model $\boldsymbol{M}^r_{\text{global}}$. This metric captures the angular alignment between local updates and the global model, enabling the detection of updates that significantly deviate in direction—a phenomenon we refer to as *Angular Deviation.* Using these similarity scores, the server applies the HDBSCAN algorithm to cluster the client updates. This clustering process distinguishes benign updates, which tend to form dense clusters due to their similarity, from malicious updates, which appear as outliers due to their dissimilarity. Importantly, by using scalar similarity scores rather than full high-dimensional parameter vectors, the clustering process remains scalable and efficient.

Within the benign cluster, the server computes the median of the model weights by taking the median value of each parameter across the models. This procedure mitigates the impact of extreme values, addressing the issue of *Magnitude Deviation* introduced by malicious updates. Subsequently, the server selects the models closest to this median to form a representative ensemble, filtering out residual outliers.

The selected ensemble is aggregated to form an initial distilled model. To further refine this model and improve its resilience against subtle backdoor triggers—especially under Non-IID settings—the framework applies a Knowledge Distillation (KD) process. During this process, the ensemble of benign models guides the refinement of the distilled model $\boldsymbol{M}^{r+1}_{\text{global}}$, ensuring that the updated global model reflects the collective benign behavior.

This step mitigates the risk of *Subtle Deviations*, where attackers mimic benign updates in both magnitude and direction.

Finally, the refined global model $\boldsymbol{M}^{r+1}_{\text{global}}$ is broadcast to benign clients. For clients identified as malicious, RKD supports two strategies:

- **Exclusion Strategy:** Withholds the updated global model, forcing malicious clients to continue training on their previous model.
- **Perturbation Strategy:** Supplies a perturbed version of the global model:

$$\boldsymbol{M}^{r+1}_{\text{pert}} = \boldsymbol{M}^{r+1}_{\text{global}} + \boldsymbol{\eta}, \tag{4.1}$$

where $\boldsymbol{\eta}$ is a noise vector with small magnitude (e.g., $\|\boldsymbol{\eta}\| \approx 1 \times 10^{-4}$). This approach, referenced in Equation 4.1, limits the adversary's ability to infer its classification status.

If a previously flagged client is later reclassified as benign, it resumes receiving the standard global model $\boldsymbol{M}_{\text{global}}^{r+1}$. The RKD process is summarized in Algorithm 5.

4.2.2 Automated Clustering

This component identifies and excludes potentially malicious models to safeguard the integrity of the federated learning process. We leverage cosine similarity and the HDBSCAN clustering algorithm to differentiate benign from malicious model updates. Let $\mathbf{w}_i^r$ be the local model parameters from client i at iteration r, and let $\mathbf{w}_{\text{global}}^r$ represent the current global model. The server computes the cosine similarity between each client's local model and the current global model as:

$$s_i = \frac{(\mathbf{w}_i^r)^\top \, \mathbf{w}_{\text{global}}^r}{\|\mathbf{w}_i^r\| \, \|\mathbf{w}_{\text{global}}^r\|}, \quad i = 1, \ldots, N, \tag{4.2}$$

where higher similarity scores s_i imply stronger alignment with the global model (i.e., likely benign), whereas malicious updates tend to deviate more. Equation 4.2 is used to inform the clustering mechanism for anomaly detection.

The resulting similarity scores $\{s_i\}$ are then clustered using HDBSCAN. Operating on these scalar values, rather than the full high-dimensional parameter vectors, significantly reduces computational overhead while retaining enough information to distinguish between benign and adversarial updates. A key requirement in HDBSCAN is the minimum cluster size Q, which we set adaptively at each training round r:

$$Q = \max\left(2, \, \lceil 0.2N - r \rceil\right), \tag{4.3}$$

where N is the number of participating clients, and $\lceil \cdot \rceil$ denotes the ceiling function. This formulation is a heuristic designed to reflect two main observations: First, in early rounds ($r \approx 0$), local models can exhibit substantial variance before

Algorithm 5 RKD Framework Methodology

```
Require: Clients $\mathcal{A}$, number of iterations $R$, malicious client strategy $S \in$ {Exclusion, Perturbation} return Final global model $M^R_{\text{global}}$
Initialize global model $M^0_{\text{global}}$
for $r = 0$ to $R - 1$ do
    if $r = 0$ then
        Send $M^0_{\text{global}}$ to all clients in $\mathcal{A}$
    else
        Send $M^r_{\text{global}}$ to benign clients $\mathcal{A}^{r-1}_{\text{benign}}$
        if $S =$ Exclusion then
            for each malicious client $i \in \mathcal{A} \setminus \mathcal{A}^{r-1}_{\text{benign}}$ do
                Send the current local model $M^r_i$ to client $i$
            end for
        else[Otherwise, using Perturbation]
            Compute perturbed model $M^r_{\text{pert}} = M^r_{\text{global}} + \eta$
            Send $M^r_{\text{pert}}$ to malicious clients $\mathcal{A} \setminus \mathcal{A}^{r-1}_{\text{benign}}$
        end if
    end if
    Collect models $\{M^r\} = \{M^r_i \mid i \in \mathcal{A}\}$
    Identify $\{M^r_{\text{benign}}\}$ and $\mathcal{A}^r_{\text{benign}}$ via clustering        ▷ See Algorithm 6
    Select ensemble models $\mathcal{E}^r$ from $\{M^r_{\text{benign}}\}$
    Compute aggregated model $M^r_{\text{distill}}$ from $\mathcal{E}^r$
    Update $M^{r+1}_{\text{global}} = \text{KD}(M^r_{\text{distill}}, \mathcal{E}^r)$        ▷ See Algorithm 7
end for
return Final global model $M^R_{\text{global}}$
```

converging, so a larger $Q \approx 0.2N$ helps avoid prematurely breaking up the main benign cluster. Second, as training progresses, benign client models converge toward $\mathbf{w}^r_{\text{global}}$, and their updates become more homogeneous. Gradually decreasing Q by about 1 per round makes HDBSCAN more sensitive to small outlier clusters, thus better isolating subtle malicious deviations.

After assigning cluster labels $\{L_i\}$ via HDBSCAN, each identified cluster C_k has a mean cosine similarity defined as:

$$\mu_k = \frac{1}{|C_k|} \sum_{i \in C_k} s_i. \tag{4.4}$$

Letting $\mu_{\max} = \max(\{\mu_k\})$, we designate the cluster(s) that satisfy $\mu_k = \mu_{\max}$ as benign, while all others are considered malicious:

$$\text{Cluster } C_k = \begin{cases} \text{benign}, & \text{if } \mu_k = \mu_{\max}, \\ \text{malicious}, & \text{otherwise.} \end{cases} \tag{4.5}$$

Models in the benign cluster continue to participate in subsequent training rounds with the updated global model, whereas malicious updates are handled differently based on the chosen strategy. Under the **Exclusion Strategy**, malicious clients are excluded from receiving the refined global model altogether, minimizing their ability to adapt. Alternatively, under the **Perturbation Strategy**, they receive a perturbed version of the global model that limits their capacity to refine an ongoing attack while also masking the fact that they have been flagged as malicious. Algorithm 6 summarizes the automated clustering procedure.

4.2.3 Model Selection

This component refines the set of benign models by selecting the most representative among them for aggregation, thereby mitigating the influence of outliers and enhancing the robustness of the global model.

Algorithm 6 Automated Clustering Algorithm

```
Require: $\{\mathbf{w}_i^r\}_{i=1}^N$: Client models at iteration $r$
Ensure: $\{\mathbf{w}_i^r\}_{i \in \mathcal{A}_{\text{benign}}^r}$, $\mathcal{A}_{\text{benign}}^r$: Benign models and client indices
for each client $i = 1$ to $N$ do
    Compute cosine similarity
        $$s_i = \frac{(\mathbf{w}_i^r)^\top \mathbf{w}_{\text{global}}^r}{\|\mathbf{w}_i^r\| \, \|\mathbf{w}_{\text{global}}^r\|}$$
end for
Apply HDBSCAN to the similarity scores $\{s_i\}$
for each cluster $C_k$ do
    Compute mean similarity
        $$\mu_k = \frac{1}{\|C_k\|} \sum_{i \in C_k} s_i$$
end for
Identify benign cluster:
    $$\mu_{\max} = \max(\{\mu_k\})$$
Identify benign clients:
    $$\mathcal{A}_{\text{benign}}^r = \{i \mid L_i = k \ \wedge \ \mu_k = \mu_{\max}\}$$
Collect benign models:
    $$\{\mathbf{w}_i^r\}_{i \in \mathcal{A}_{\text{benign}}^r}$$
return $\{\mathbf{w}_i^r\}_{i \in \mathcal{A}_{\text{benign}}^r}$, $\mathcal{A}_{\text{benign}}^r$
```

The set of benign models selected after clustering is defined as:

$$\{\mathbf{w}_i^r\}_{i \in \mathcal{A}_{\text{benign}}^r}, \tag{4.6}$$

where $\mathcal{A}_{\text{benign}}^r$ denotes the indices of benign clients at round r.

From the benign model set in Equation 4.6, we compute a median model vector by taking the element-wise median:

$$\mathbf{w}_{\text{median}}^r = \text{median}\left(\{\mathbf{w}_i^r \mid i \in \mathcal{A}_{\text{benign}}^r\}\right), \tag{4.7}$$

which minimizes the sum of absolute deviations across model dimensions.

Then, we compute the L_1 distance between each benign model and the median model:

$$d_i = \|\mathbf{w}_i^r - \mathbf{w}_{\text{median}}^r\|_1, \quad \forall i \in \mathcal{A}_{\text{benign}}^r. \tag{4.8}$$

We define an adaptive threshold based on the empirical distribution of distances:

$$\epsilon = \mu_d + k\sigma_d, \tag{4.9}$$

where μ_d and σ_d denote the mean and standard deviation of the distances $\{d_i\}$, and k is a tunable hyperparameter.

The final ensemble $\mathcal{E}^r$ is then formed by selecting only those models whose distances to the median do not exceed the threshold:

$$\mathcal{E}^r = \left\{\mathbf{w}_i^r \in \{\mathbf{w}_i^r\}_{i \in \mathcal{A}_{\text{benign}}^r} \mid d_i \leq \epsilon\right\}. \tag{4.10}$$

4.2.4 Knowledge Distillation Process

This process refines the global model by distilling knowledge from the selected ensemble of benign models $\mathcal{E}^r$. The server uses an unlabeled validation dataset D_{val} (constituting 16% of the total training data) for knowledge distillation.

For each sample $x \in D_{\text{val}}$, the server computes the logits from each model in $\mathcal{E}^r$ and averages them to produce ensemble logits:

$$\textbf{Ensemble\_Logits}(x) = \frac{1}{|\mathcal{E}^r|} \sum_{\mathbf{w}_i \in \mathcal{E}^r} f_{\mathbf{w}_i}(x), \tag{4.11}$$

where $f_{\mathbf{w}_i}(x)$ denotes the raw output scores (logits) of model $\mathbf{w}_i$.

Pseudo-labels are derived by applying the softmax function with a temperature parameter T:

$$\tilde{\boldsymbol{y}}(x) = \text{softmax}\left(\frac{\textbf{Ensemble\_Logits}(x)}{T}\right). \tag{4.12}$$

This softening technique yields a smoother distribution over classes, making the target more informative.

The distilled model $\boldsymbol{M}_{\text{distill}}$ is trained to minimize the Kullback-Leibler (KL) divergence between its own soft predictions and the ensemble-derived pseudo-labels:

$$\mathcal{L}_{\text{KD}} = D_{\text{KL}}\left(\tilde{\boldsymbol{y}}(x) \;\middle\|\; \text{softmax}\left(\frac{f_{\boldsymbol{M}_{\text{distill}}}(x)}{T}\right)\right). \tag{4.13}$$

To improve training stability and generalization, we employ a *Stochastic Weight Averaging (SWA)* scheme that maintains a running average of the model weights. After E_{KD} epochs of distillation, the SWA model is adopted as the updated global model:

$$\boldsymbol{M}^{r+1}_{\text{global}} \leftarrow \boldsymbol{M}_{\text{SWA}}. \tag{4.14}$$

Algorithm 7 outlines the complete knowledge distillation process.

4.3 Experiments

In this section, we evaluate the effectiveness of the proposed RKD framework under backdoor attack scenarios in a federated learning environment. In our setup, multiple clients collaboratively train a global model under the coordination of a central server. This iterative training process continues until the model converges. The implementation of the RKD framework is available on GitHub[1].

4.3.1 Datasets and Models

We conduct experiments on three well-known datasets—CIFAR-10, EMNIST, and Fashion-MNIST—each offering distinct image types and complexity levels:

[1] https://github.com/EbtisaamCS/RKD

Algorithm 7 Knowledge Distillation Process

```
Require: $\mathcal{E}^r$: Ensemble of selected benign models, $D_{\text{val}}$: Unlabeled data for
    distillation, $T$: Temperature for softmax, $E_{\text{KD}}$: Number of epochs, $\eta$: Learning
    rate
Ensure: Updated global model $M_{\text{global}}^{r+1}$
Ensemble_Logits$(x) = \frac{1}{\|\mathcal{E}^r\|} \sum_{M_i \in \mathcal{E}^r} f_{M_i}(x)$    ▷ Compute ensemble logits for all
    $x \in D_{\text{val}}$
Generate pseudo-labels:
    $$\tilde{y}(x) = \text{softmax}\Big(\frac{\text{Ensemble\_Logits}(x)}{T}\Big)$$
Initialize $M_{\text{SWA}} \leftarrow M_{\text{distill}}$ and $n_{\text{SWA}} \leftarrow 1$
for epoch $e = 1$ to $E_{\text{KD}}$ do
    for each mini-batch $\{x_b\} \subset D_{\text{val}}$ do
        Distill_Logits$(x_b) = f_{M_{\text{distill}}}(x_b)$
        $\mathcal{L} = D_{\text{KL}}\big(\tilde{y}(x_b) \,\big\|\, \text{softmax}\big(\frac{\text{Distill\_Logits}(x_b)}{T}\big)\big)$
        Update $M_{\text{distill}}$ using SGD with learning rate $\eta$
    end for
    $M_{\text{SWA}} \leftarrow \dfrac{n_{\text{SWA}} \cdot M_{\text{SWA}} + M_{\text{distill}}}{n_{\text{SWA}} + 1}$
    $n_{\text{SWA}} \leftarrow n_{\text{SWA}} + 1$
end for
$M_{\text{global}}^{r+1} \leftarrow M_{\text{SWA}}$
return Updated global model $M_{\text{global}}^{r+1}$
```

CIFAR-10 [32]: CIFAR-10 consists of 60,000 color images of size 32×32 pixels, evenly distributed across 10 classes. It serves as a widely-used benchmark for low-resolution image classification tasks.

EMNIST [16]: EMNIST extends MNIST by including 814,255 grayscale handwritten character images spanning 62 classes (digits and letters), each with a resolution of 28×28 pixels.

Fashion-MNIST [67]: Fashion-MNIST contains 70,000 grayscale images of fashion items from 10 categories, also at a resolution of 28×28 pixels. It presents a more challenging alternative to the original MNIST dataset.

For each dataset, we employ model architectures tailored to their specific characteristics:

CIFAR-10: A ResNet-18 [24] architecture is used to capture hierarchical visual features in color images. Training is conducted with a batch size of 64 and an initial learning rate of 0.01.

EMNIST: A lightweight convolutional neural network (CNN) with two convolutional layers (each followed by max pooling and dropout) and a fully connected output layer. It is trained with a batch size of 64 and a learning rate of 0.001.

Fashion-MNIST: A CNN consisting of two convolutional layers with batch normalization and dropout, followed by a fully connected classifier. Training uses a batch size of 64 and a learning rate of 0.001.

We select CIFAR-10, EMNIST, and Fashion-MNIST for our experiments instead of larger datasets like CIFAR-100 or Tiny ImageNet for three key reasons. First, these three datasets are standard benchmarks in federated learning research, especially in the context of security and robustness, which ensures comparability with prior work. Second, their relatively small size and manageable complexity enable efficient experimentation under a wide range of attack scenarios without excessive computational cost—an important factor for extensive evaluation of defense mechanisms. Third, the combination of color (CIFAR-10) and grayscale (EMNIST and Fashion-MNIST) data, along with varied numbers of classes and image characteristics, allows us to test the adaptability and generalizability of our defense across different data modalities. Our focus is not on model performance under ideal conditions but rather on assessing robustness in challenging FL settings with limited resources, where practical defenses are most needed.

4.3.2 Attack Setup

To assess the robustness of RKD in the presence of backdoor attacks, we simulate an FL environment with 30 clients. We consider three scenarios in which 20%, 40%, or 60% of these clients are compromised by an adversary.

Each compromised client injects backdoor triggers into 50% of its local training data. The backdoor trigger is a specific pattern added to images, and the labels of these poisoned samples are overwritten with a target class defined by the adversary. This setup reflects a realistic attack in which malicious clients embed a backdoor while preserving normal performance on clean data.

All clients follow the FL protocol, submitting model updates to the central server. However, the compromised clients aim to bias the global model toward recognizing their backdoor trigger. Meanwhile, benign clients train on unmodified local data.

4.3.3 Threat Model

We consider a practical federated learning setup where a subset of clients may behave maliciously by launching backdoor attacks. The central server is honest but untrusted—it cannot identify malicious clients and relies solely on received updates and auxiliary unlabeled data for defense. Attackers act independently and lack access to other clients' updates, yet they craft updates that mimic benign behavior to evade detection.

To evaluate RKD under challenging and diverse adversarial conditions, we test it against four representative backdoor attack strategies:

- **A3FL** [76]: An adaptive backdoor method that refines triggers via Projected

Gradient Descent (PGD) to remain effective across evolving global model updates.

- **F3BA** [18]: Selectively flips model parameters with the highest sensitivity to the global loss, embedding stealthy triggers with minimal impact on model behavior.
- **DBA** [68]: Distributes the trigger across multiple malicious clients, maintaining stealth at the individual level while ensuring the full backdoor emerges upon aggregation.
- **ADBA** [21]: An anti-distillation backdoor attack adapted to FL, where malicious clients embed triggers that persist through the distillation process between global and client models.

These attacks represent both adaptive and distributed threat scenarios, allowing for a comprehensive evaluation of RKD's robustness.

4.3.4 Heterogeneous Setting

A common and challenging issue in FL arises from Non-IID data distributions across clients. To simulate realistic heterogeneity, we partition each dataset among clients using a Dirichlet distribution [75] with concentration parameter α. Each client i receives a vector of class proportions p_i, where:

$$p_i = [p_{i,1}, p_{i,2}, \ldots, p_{i,C}] \sim \text{Dirichlet}(\alpha), \tag{4.15}$$

with C being the number of classes.

A smaller value of α leads to more imbalanced class distributions across clients, thereby simulating stronger Non-IID conditions. Conversely, larger α values yield more uniform class distributions, approximating the IID setting.

We explore a range of α values to vary the degree of Non-IID data distributions among clients:

- **Extreme Heterogeneity:** $\alpha \in \{0.5, 0.3\}$. In these cases, clients predominantly receive data from only a narrow subset of classes.

- **Moderate Heterogeneity:** $\alpha \in \{0.9\}$. Class distributions are relatively more balanced among clients.

- **IID:** We also evaluate the IID scenario, where data is independently and identically distributed across all clients in the Appendix A.

By varying α across these ranges yields a continuum of data heterogeneity levels, facilitating a comprehensive assessment of how robustly the proposed RKD framework handles adversarial interference and adapts to the naturally occurring variability in client datasets.

4.3.5 Compared Defence Baselines.

In our experimental evaluation, we compare RKD against several state-of-the-art defenses. Clustering-based methods include *RFCL* (proposed in Chapter 3) and *FLAME* [50], which detect outlier updates via density-based clustering. Knowledge-distillation approaches include *FedDF* [37], *FedBE* [15], and *FedRAD* [60], all of which fuse client updates into a distilled global model. The Robust Learning Rate method (*RLR*) [52] adaptively scales each client's learning rate based on update alignment to curb adversarial influence. Finally, *FoolsGold* (FG) [20] assigns client weights according to gradient similarity to penalize coordinated adversaries.

4.3.6 Evaluation Metrics

We utilized two key evaluation metrics: *Main Task Accuracy (MTA)* and *Attack Success Rate (ASR).* These metrics provide a comprehensive understanding of the model's performance on legitimate tasks and its resistance to backdoor triggers.

Main Task Accuracy (MTA)

MTA measures the classification accuracy of the global model on a clean test dataset D_{test}, reflecting its ability to correctly predict true labels without backdoor interference. It is defined as:

$$\text{MTA} = \frac{\|\{x \in D_{\text{test}} \mid f(x) = y\}\|}{\|D_{\text{test}}\|}, \tag{4.16}$$

where x is an input sample from D_{test}, $f(x)$ is the prediction of the global model f, and y is the corresponding true label. Here, $\|\cdot\|$ denotes the cardinality of the set.

A higher MTA value indicates that the model performs well on the main classification task.

Attack Success Rate (ASR)

ASR evaluates the success of a backdoor attack by measuring the fraction of poisoned inputs misclassified into the attacker's target class. It is computed using a backdoor test dataset D_{poison}:

$$\text{ASR} = \frac{\|\{x \in D_{\text{poison}} \mid f(x) = y_{\text{target}}\}\|}{\|D_{\text{poison}}\|}, \tag{4.17}$$

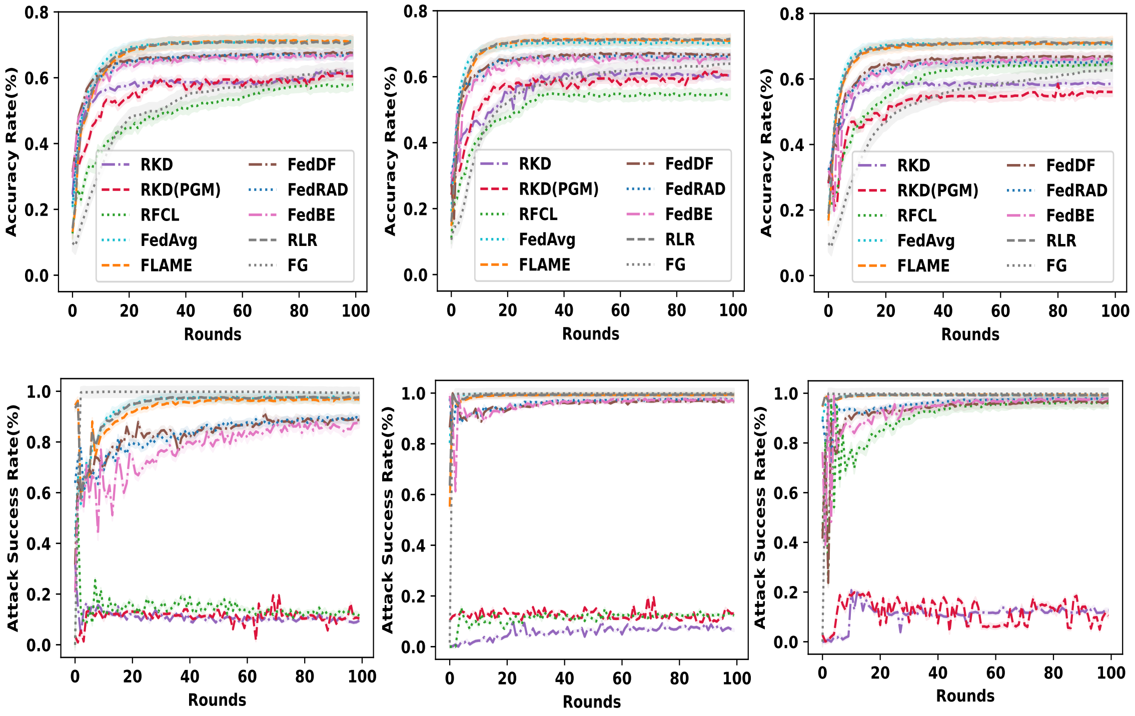
where $f(x)$ is the model's prediction for input x, and y_{target} is the attacker-specified target class.

A higher ASR indicates a more effective backdoor attack. A lower ASR indicates greater robustness against backdoor attacks, as it indicates that the model is less likely to misclassify backdoor inputs into the attacker's target class.

The goal of an effective Defence mechanism like the RKD framework is to maintain a high MTA while minimizing the ASR. This balance ensures that the model retains its performance on legitimate data while being resilient to manipulation attempts by adversaries. In our experiments, we focus on achieving this balance to demonstrate the RKD framework's capability to defend against sophisticated backdoor attacks without degrading the overall model performance.

4.3.7 Experimental Results

We evaluated the robustness of the RKD framework against advanced backdoor attacks in FL. The models were trained under Non-IID data distributions, measuring the MTA and ASR. To ensure reliability, all experiments were repeated five times with different data resampling, with confidence intervals reported at a significance level of $\rho = 0.01$. Rounds denote communication iterations in federated learning, where the global model is updated based on local client training (typically involving five local epochs) followed by server-side aggregation.



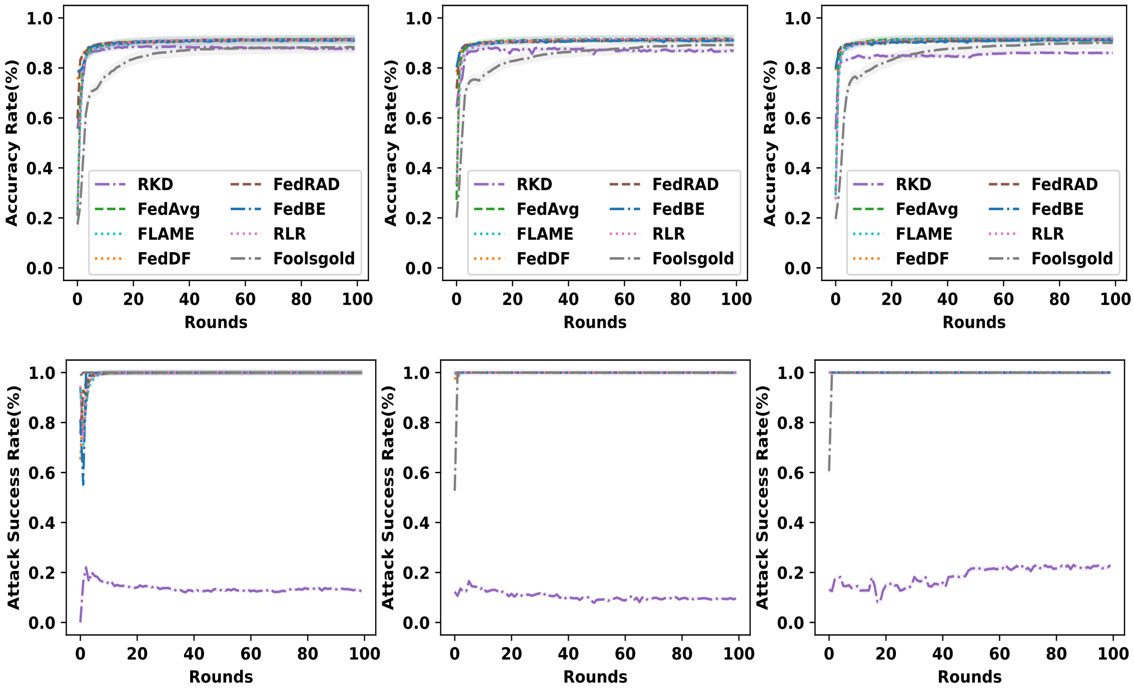
(a) 20% of A3FL (b) 40% of A3FL (c) 60% of A3FL

Figure 4.2: Performance of baselines and RKD on CIFAR-10 under Non-IID ($\alpha = 0.3$), evaluated against 20%, 40%, and 60% A3FL attacker clients.

Defence Against A3FL Attack.

Under highly heterogeneous Non-IID conditions ($\alpha = 0.3$), RKD demonstrated significant resilience against the A3FL attack on the CIFAR-10 and Fashion-MNIST datasets. As shown in Figures 4.2 and 4.3, RKD achieved a substantially lower attack success rate while maintaining high accuracy compared to baseline methods.

These results confirm that RKD effectively distinguishes malicious from benign client updates and aggregates only reliable models. By restricting the dissemination of the updated global model to clients identified as benign, RKD prevents adversaries from adapting their strategies based on the latest global updates. In our primary



(a) 20% of A3FL

(b) 40% of A3FL

(c) 60% of A3FL

Figure 4.3: Performance of baselines and RKD on Fashion-MNIST under Non-IID ($\alpha = 0.3$), evaluated against 20%, 40%, and 60% A3FL attacker clients.

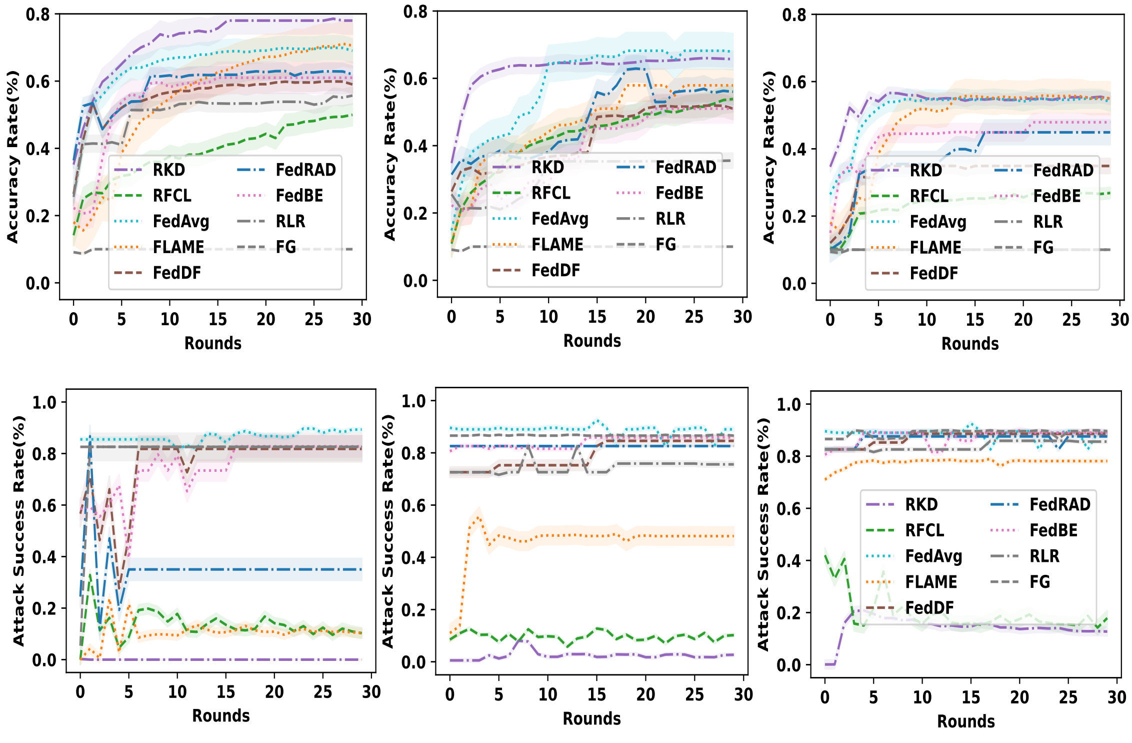
approach—the **Exclusion Strategy**—malicious clients continue training with their current local models. Alternatively, the **Perturbation Strategy** (denoted as RKD (PGM)) provides suspected malicious clients with a minimally perturbed global model: $M_{\text{pert}}^{r+1} = M_{\text{global}}^{r+1} + \eta$, where $\|\eta\| \approx 1 \times 10^{-4}$. This slight perturbation effectively obscures the precise state of the global model, thereby limiting the opportunity for adaptive adversaries to refine their attacks.

Comparative experiments show that RKD (PGM) maintains an average accuracy nearly identical to that of RKD using the Exclusion Strategy, while still mitigating adaptive attack risks. Overall, the experimental findings illustrate that RKD robustly mitigates backdoor attacks under Non-IID conditions.

Defense Against F3BA Attack.

RKD effectively defends against F3BA on CIFAR-10 and EMNIST datasets under non-IID conditions ($\alpha = 0.5$), as substantiated by Figures 4.4 and 4.5. Using cosine similarity-based clustering, RKD filters out anomalies from compromised clients and integrates knowledge distillation to maintain low ASR and high accuracy.

RKD's iterative training enhances the global model's accuracy and resilience, demonstrating its superiority over methods like FedAvg. This is particularly evident in high attacker ratios of 40% and 60%, highlighting RKD's robust defence against sophisticated attacks like F3BA.

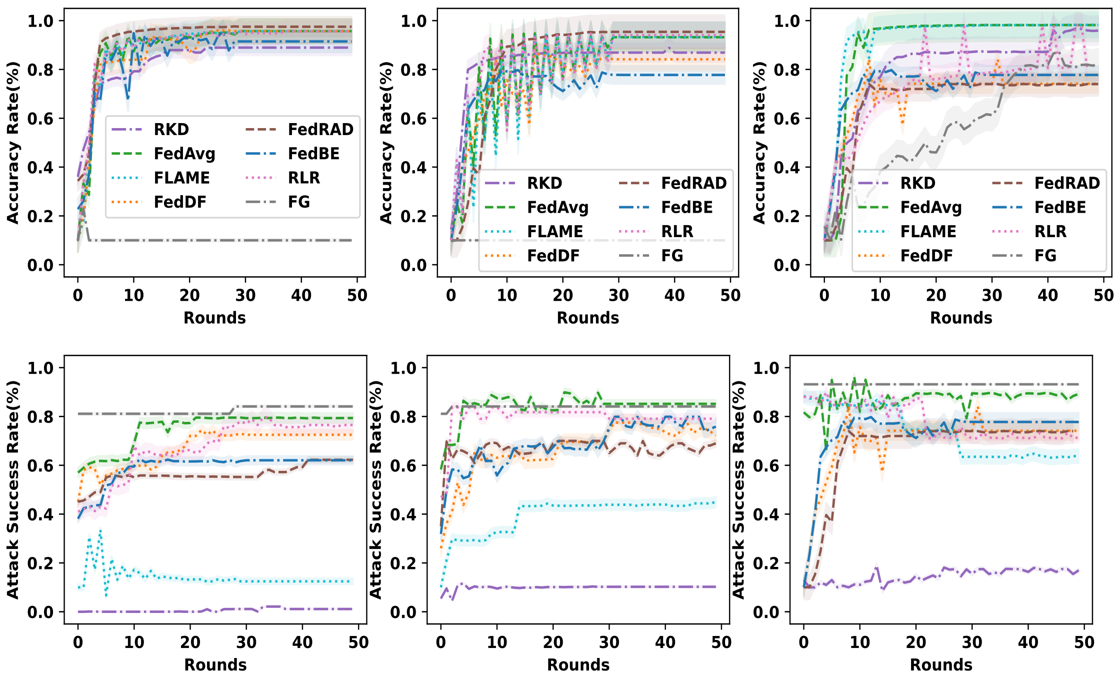


(a) 20% of F3BA (b) 40% of F3BA (c) 60% of F3BA

Figure 4.4: Performance of baselines and RKD on CIFAR-10 under Non-IID ($\alpha = 0.5$), evaluated against 20%, 40%, and 60% F3BA attacker clients.

Defense Against DBA Attack.

RKD effectively defends against the Distributed Backdoor Attack (DBA) on CIFAR-10 and EMNIST under non-IID settings ($\alpha = 0.9$), as shown in Figures 4.6 and 4.7. Using cosine similarity-based clustering, RKD detects and isolates malicious updates. Median model selection ensures that only benign models contribute to the global model, minimizing backdoor triggers.



(a) 20% of F3BA (b) 40% of F3BA (c) 60% of F3BA

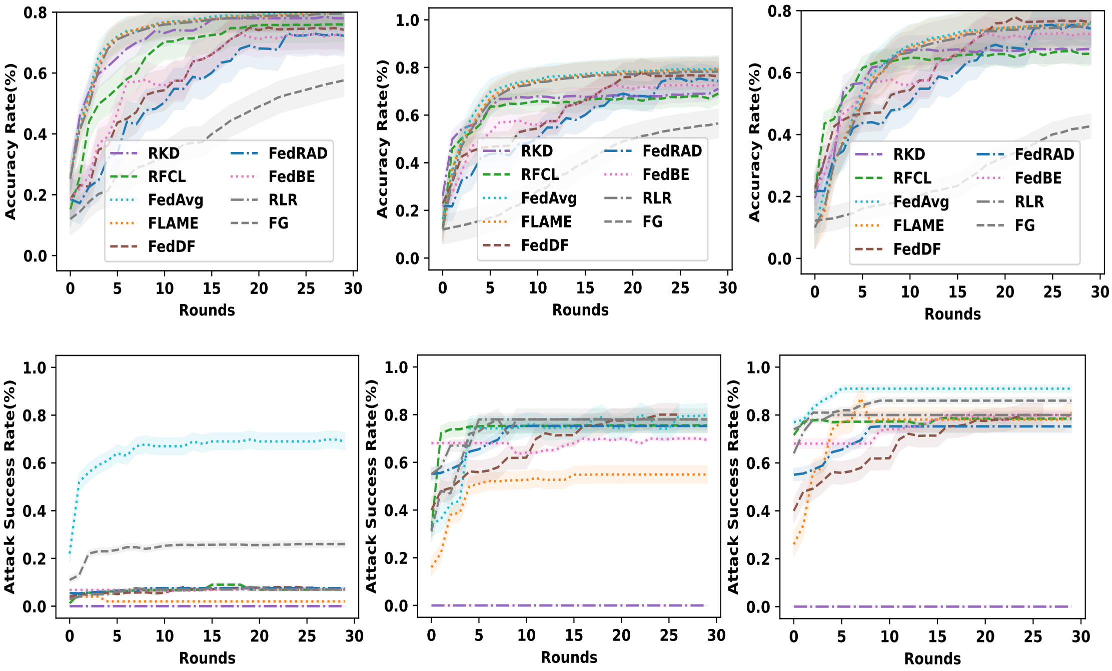
Figure 4.5: Performance of baselines and RKD on EMNIST under Non-IID ($\alpha = 0.5$), evaluated against 20%, 40%, and 60% F3BA attacker clients.

During knowledge distillation, RKD synthesizes insights from vetted models into a robust aggregated model, enhancing generalizability and security. Compared to methods like FedDF, FedRAD, and FedBE, RKD provides superior protection by meticulously analyzing and distilling knowledge from selected models. This enables RKD to maintain high accuracy while significantly reducing the ASR, demonstrating its effectiveness against sophisticated attacks like DBA.

Defense Against ADBA Attack.

The RKD framework robustly defends against Anti-Distillation Backdoor Attacks (ADBA) on CIFAR-10 under Non-IID conditions ($\alpha = 0.5$), as shown in Figure 4.8. Compared to FedAvg and other baseline methods, RKD effectively detects and mitigates ADBA backdoor attacks, demonstrating superior resilience and enhanced model integrity in challenging heterogeneous environments.

Defense Against TSBA Attack.



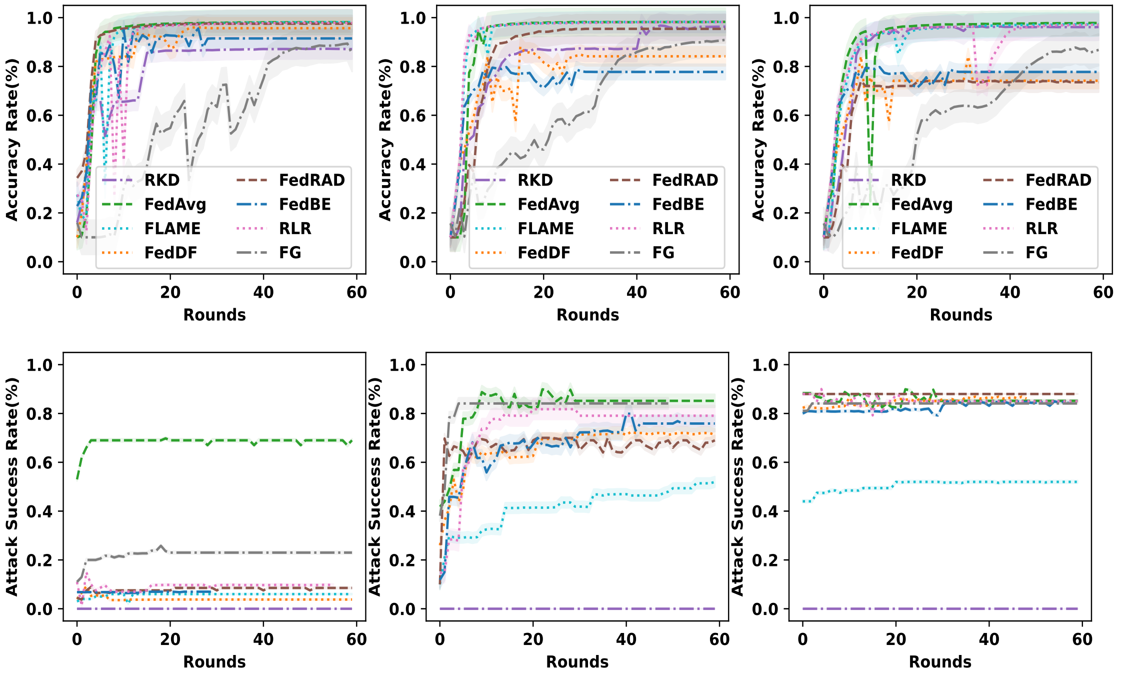
(a) 20% of DBA (b) 40% of DBA (c) 60% of DBA

Figure 4.6: Performance of baselines and RKD on CIFAR-10 under Non-IID ($\alpha = 0.9$), evaluated against 20%, 40%, and 60% DBA attacker clients.

The RKD framework robustly defends against TSBA on CIFAR-10 and EMNIST under Non-IID conditions ($\alpha = 0.5$), as shown in Figures 4.9 and 4.10. RKD detects and mitigates TSBA manipulations, maintaining high accuracy and low ASR even with increased attacker ratios. Unlike other methods that falter under poisoned conditions, RKD excels with clean and poisoned datasets.

The Impact of Heterogeneous Degree.

We evaluated baseline defence methods and the RKD framework under varying degrees of data heterogeneity, including moderate ($\alpha = 0.7$) and extreme ($\alpha = 0.3, 0.1$) Non-IID conditions (see Figure 4.11). Under extreme heterogeneity, many baseline methods achieve high accuracy on clean inputs but struggle to detect subtle backdoor triggers—resulting in elevated ASR. In contrast, RKD consistently maintains robust defence by effectively excluding malicious updates, which helps to suppress ASR while sustaining high MTA.



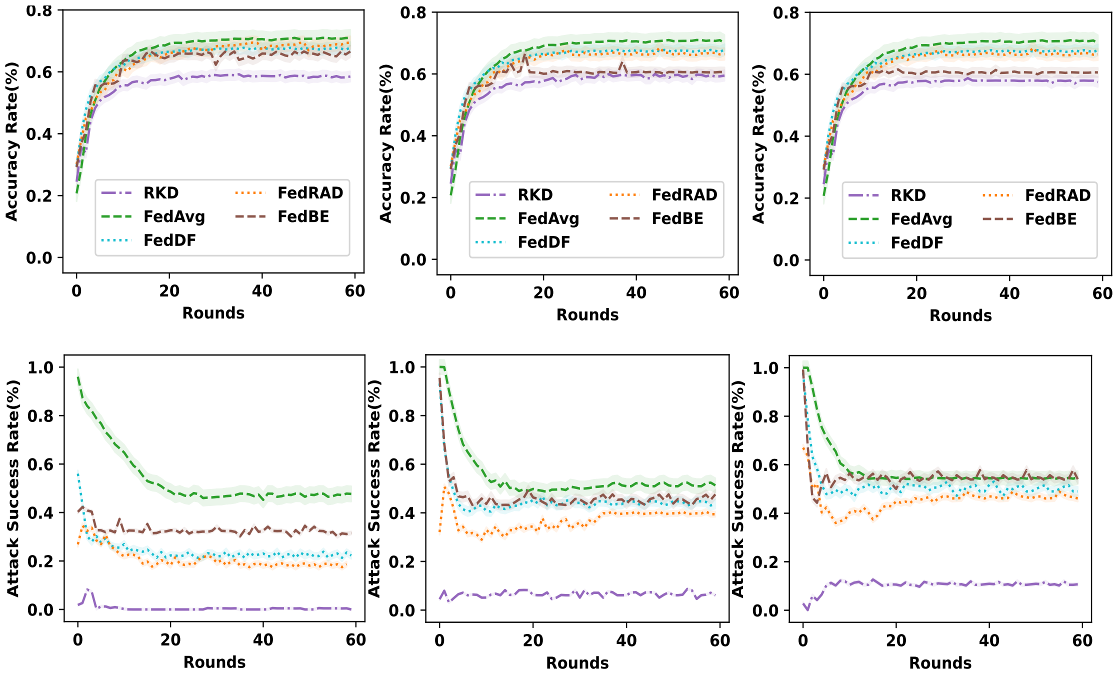
(a) 20% of DBA (b) 40% of DBA (c) 60% of DBA

Figure 4.7: Performance of baselines and RKD on EMNIST under Non-IID ($\alpha = 0.9$), evaluated against 20%, 40%, and 60% DBA attacker clients.

Notably, although extreme heterogeneity adversely impacts overall accuracy for all methods, RKD outperforms baseline defences by achieving a better balance between low ASR and high MTA. This indicates that a key contribution of our work is enhancing robustness in highly Non-IID scenarios. Moreover, under extremely Non-IID conditions ($\alpha = 0.1$), baseline methods often struggle to generalize, resulting in model collapse that leads to a low ASR—since their offline behavior prevents an accurate assessment of robustness. In contrast, RKD sustains stable learning and robust defence, achieving both high accuracy and a genuinely low ASR. Overall, these results highlight RKD's superior effectiveness in challenging heterogeneous data environments.

Empirical Analysis of Q Sensitivity.

We evaluated the impact of the minimum cluster size Q on Main Task Accuracy (MTA) and Attack Success Rate (ASR) using the CIFAR-10 dataset under a Non-



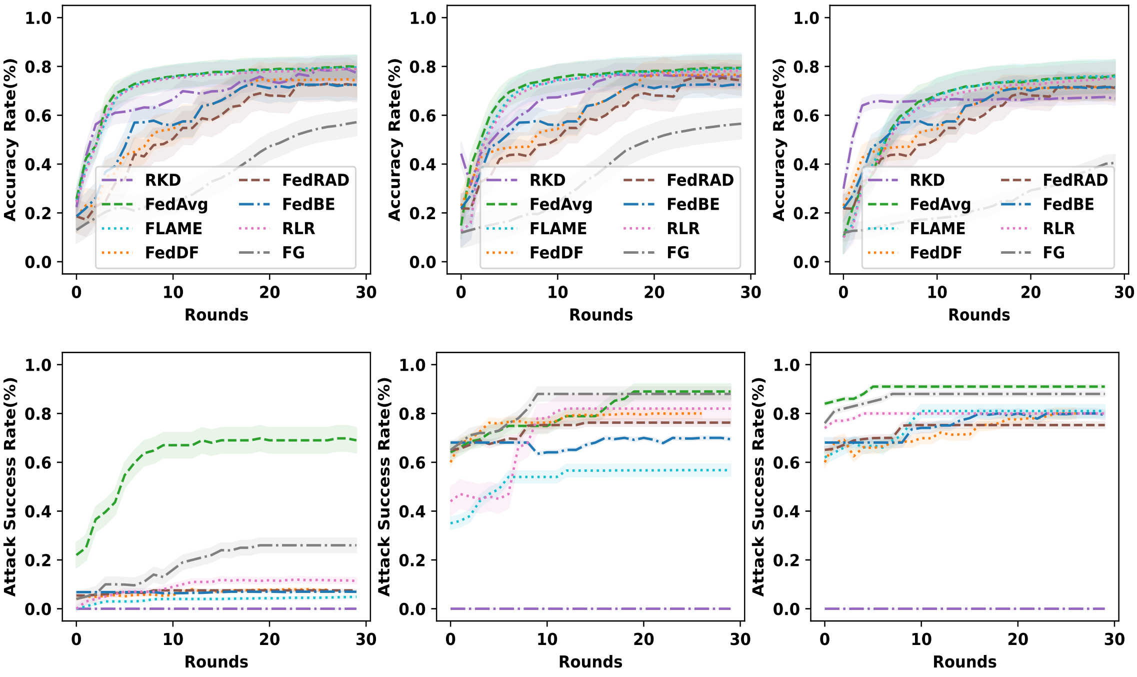
(a) 20% of ADBA (b) 40% of ADBA (c) 60% of ADBA

Figure 4.8: Performance of baselines and RKD on CIFAR-10 under Non-IID ($\alpha = 0.5$), evaluated against 20%, 40%, and 60% ADBA attacker clients.

IID setting with 30 clients, 40% of which were malicious and executing A3FL backdoor attacks. Figure 4.12 presents the results.

When Q is fixed at 2, the resulting small clusters allow malicious updates to dominate, yielding a high ASR despite a relatively high MTA. In contrast, fixing Q at 20 causes many malicious updates to be included in the benign cluster, leading to slightly lower accuracy and higher ASR.

A dynamic adjustment of Q mitigates these issues by excluding malicious updates while retaining the majority of benign ones, thus ensuring consistently high MTA and low ASR. These findings underscore the importance of dynamically tuning Q to reduce the influence of residual outliers and adversarial updates, thereby preserving the overall robustness and performance of the global model.



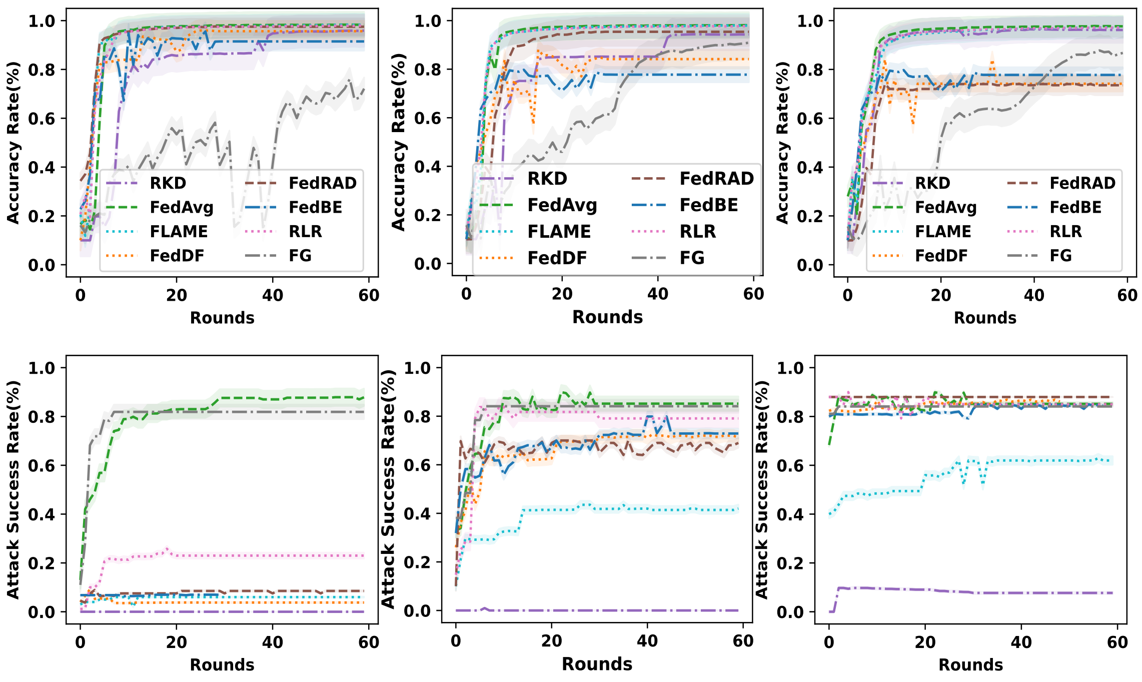
(a) 20% of TSBA (b) 40% of TSBA (c) 60% of TSBA

Figure 4.9: Performance of baselines and RKD on CIFAR-10 under Non-IID ($\alpha = 0.5$), evaluated against 20%, 40%, and 60% TSBA attacker clients.

4.3.8 Scalability Analysis

RKD enhances scalability by applying cosine similarity to model updates before clustering, transforming high-dimensional parameter vectors into scalar similarity scores. This dimensionality reduction significantly lowers computational complexity, making the clustering process more efficient. By avoiding clustering in the high-dimensional parameter space, RKD reduces both the time and resources required for defence operations.

As shown in Table 4.2, RKD's defence time is 42.029 seconds, substantially faster than FedDF and FedBE, which require 141.714 and 198.765 seconds, respectively. While RLR and RFCL exhibit the shortest defence time, it compromises on detection accuracy due to its simplistic approach. FLAME is slightly more efficient in defence time, but RKD achieves a better balance between performance and robustness. These results highlight RKD's overall efficiency and scalability, demonstrating that



(a) 20% of TSBA (b) 40% of TSBA (c) 60% of TSBA

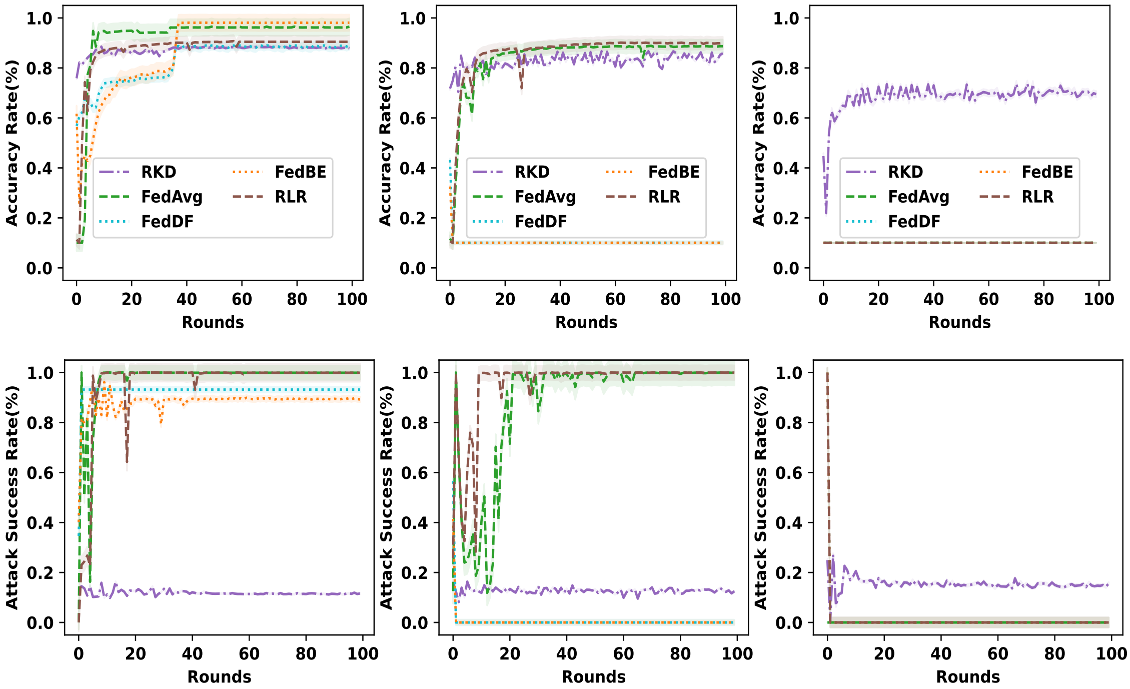
Figure 4.10: Performance of baselines and RKD on EMNIST under Non-IID ($\alpha = 0.5$), evaluated against 20%, 40%, and 60% TSBA attacker clients.

its methodological design—specifically the use of cosine similarity and efficient clustering—provides robust defence without incurring significant computational overhead.

4.3.9 Ablation Study

We conducted ablation studies to evaluate the effectiveness of each component within the RKD framework against sophisticated backdoor attacks. Specifically, we analyzed the impact of removing key components: Automated Clustering, Model Selection, and Knowledge Distillation, as shown in Figure 4.13.

RKD without Clustering and Model Selection (Median). Removing both the *Automated Clustering* and *Model Selection* components, and leaving only the knowledge distillation process, significantly weakens the framework's defenses. While it performs reasonably well under a 20% DBA attack, as the proportion of



(a) Non-IID=0.7 (b) Non-IID=0.3 (c) Non-IID=0.1

Figure 4.11: Performance impact of heterogeneous degrees on baselines and RKD on Fashion-MNIST, evaluated against 60% F3BA attacker clients.

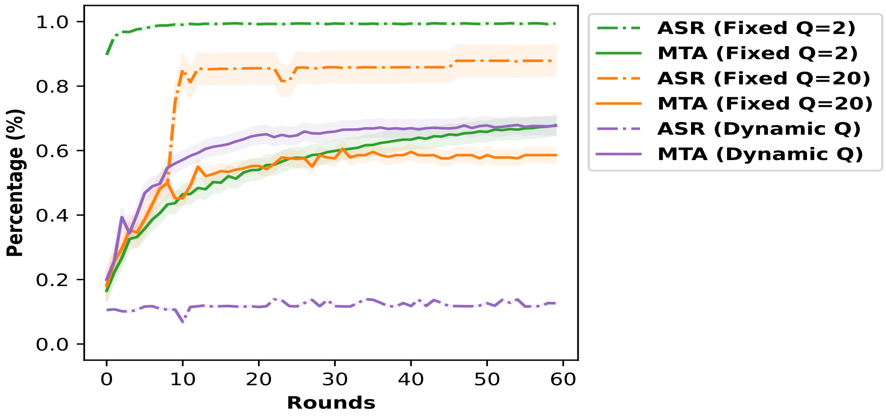


Figure 4.12: Impact of Q on MTA and ASR.

adversarial clients rises to 40%, its defensive capabilities drop sharply. At a 60% F3BA attacker ratio, the model suffers from severe performance degradation, with misclassifications aligning with attackers' objectives. This highlights that without clustering, RKD is unable to effectively identify and isolate malicious updates,

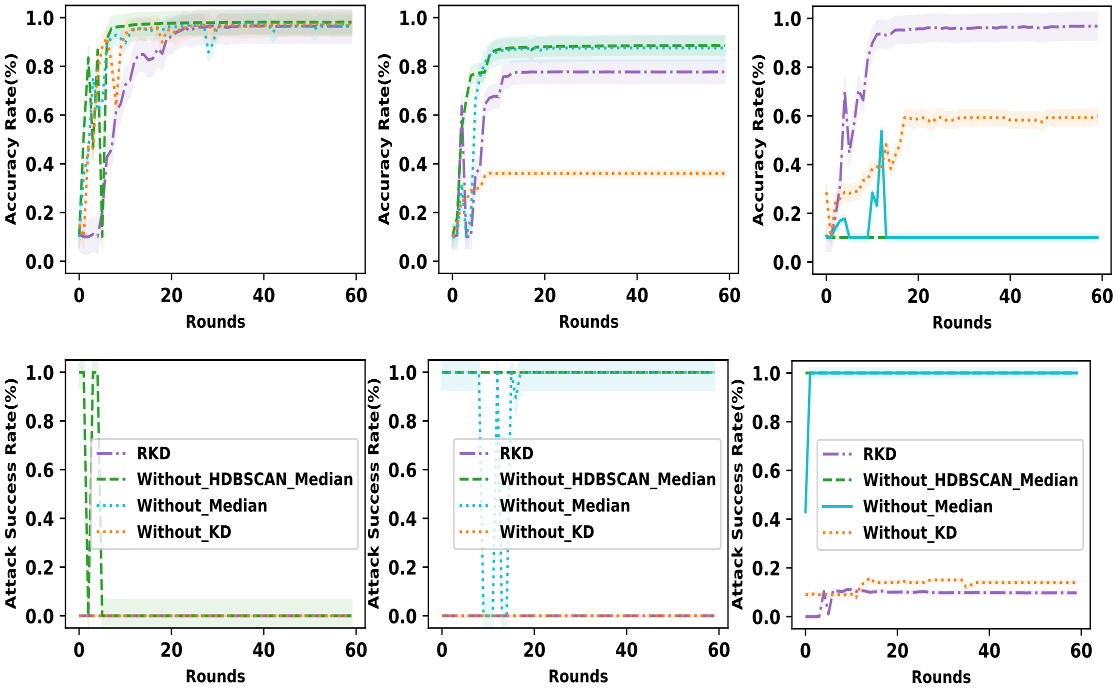
Table 4.2: Defense time comparison (in seconds) across methods, ordered fastest to slowest. Methods proposed in this thesis are marked as (ours).

Method	**Defense Time (s)**
RLR [52]	0.02
RFCL (ours, see Chapter 3)	2.75
FLAME [50]	37.06
RKD (ours, see Chapter 4)	42.03
FedDF [37]	141.71
FedBE [15]	198.77

allowing backdoor attacks to compromise the global model.

RKD without Model Selection (Median). Excluding the *model selection* component while retaining HDBSCAN clustering, this variant effectively manages a 20% DBA attack ratio. However, with a 40% attack ratio, performance is noticeably decreased. Under a 60% F3BA attacker ratio, the removal of model selection further impairs RKD's defense, resulting in increased ASR and reduced MTA. Even after the clustering phase effectively isolates most malicious updates, some malicious or anomalous updates may still be present. Without employing the median to mitigate the influence of residual outliers, these outliers can disproportionately affect the aggregated model, resulting in unstable ASR measurements. These findings underscore the critical role of model selection in refining the aggregation process by selecting models closest to the benign cluster centroid, thus enhancing robustness against higher proportions of attackers.

RKD without Knowledge Distillation. Retaining both clustering and model selection but omitting the *Knowledge Distillation* module, this variant successfully identifies and excludes malicious models, achieving a lower ASR. However, it suffers from a significant drop in accuracy, approximately 20% lower than the full RKD framework. This underscores the crucial role of KD in transferring knowledge from the selected ensemble of models to the global model, which enhances both accuracy and generalization, particularly across Non-IID settings. Without KD, the framework is unable to effectively integrate and refine the knowledge from ensemble models, leading to poor performance in Non-IID environments despite successful isolation of malicious updates. KD specifically addresses the challenge of generalization in Non-IID scenarios, ensuring the model remains robust and effective.



(a) 20% of DBA (b) 40% of DBA (c) 60% of F3BA

Figure 4.13: Ablation study of RKD method against attacks.

4.3.10 Discussion and Limitations

The empirical evaluations presented in this work demonstrate that the Robust Knowledge Distillation (RKD) framework can consistently suppress backdoor attacks while maintaining high main-task accuracy in FL settings. By integrating automated clustering (to isolate outliers), median-based model selection (to refine the aggregation process), and knowledge distillation (to fuse benign models into a resilient global model), RKD addresses the distinct challenges posed by non-IID data distributions and adaptive adversaries. Ablation studies confirm that each component contributes meaningfully to defence efficacy, underscoring the importance of a holistic design that balances detection, exclusion, and refined model.

The key strengths of RKD lie in its capacity to operate under a wide range of adversarial ratios and heterogeneous data conditions. The clustering phase leverages cosine similarity, which captures the directional alignment of updates rather than their magnitude—a property crucial for detecting stealthy attacks such as F3BA and TSBA. The median-based model selection then ensures that only models representative of benign updates proceed to knowledge distillation, thereby avoiding the risk of inheriting partial or subtle backdoor triggers.

Moreover, the dynamic adjustment of HDBSCAN's minimum cluster size Q allows sensitivity to outliers to be systematically tuned over the course of FL training, striking a balance between early-round diversity and late-round detection.

Despite its strong performance, RKD faces several limitations. The knowledge distillation process relies on an unlabeled dataset D_{val}. In real-world FL applications, assembling or curating such a dataset may be impractical—particularly when the target domain is ill-defined or highly diverse. Exploring synthetic datasets approaches could help alleviate this requirement.

Although cosine similarity–based clustering is more efficient than high-dimensional parameter clustering, RKD still entails additional computational steps (e.g., ensemble logits computation for knowledge distillation). While our experiments indicate that RKD remains competitive in throughput, investigating lightweight approximations to clustering or faster ensemble distillation strategies would further enhance scalability, especially in resource-constrained FL scenarios.

While results show that RKD remains robust under challenging levels of data heterogeneity ($\alpha \in \{0.1, 0.3\}$), more extreme skew can degrade global performance or heighten the false-positive rate during clustering. Further refinement of model selection and distillation processes is needed to adapt seamlessly to highly imbalanced or non-representative data distributions without compromising accuracy.

Both the dynamic formula for Q in HDBSCAN and the threshold parameter ϵ in the model selection of the median hinge on careful tuning. Although we demonstrate that these hyperparameters generalize well to various benchmarks, automating their selection or making them adapt over time remains an open question.

4.4 Conclusion

In this chapter, we introduced the Robust Knowledge Distillation (RKD) framework as a multifaceted defence against backdoor attacks in Federated Learning (FL). By integrating automated clustering, median-based model selection, and a knowledge distillation module, RKD systematically detects, isolates, and excludes malicious contributions under a wide range of adversarial scenarios and Non-IID data.

Our experimental results demonstrated that RKD consistently achieves both high main-task accuracy (MTA) and low attack success rates (ASR), even when faced with up to 60% malicious clients. The automated clustering phase, which dynamically adjusts the HDBSCAN minimum cluster size Q, proved effective in isolating subtle outlier updates; the median-based selection process further refines the set of benign models, and knowledge distillation ensures that the aggregated global model captures the collective strengths of these benign models. Extensive ablation studies confirmed that each component within RKD contributes significantly to its overall efficacy.

While RKD offers strong performance and scalability advantages over several existing robust FL approaches, key avenues for further research remain. These include reducing reliance on an auxiliary dataset, devising more efficient clustering schemes for extremely large-scale federations, and enhancing adaptivity to evolving or multi-stage adversarial attacks. Addressing these challenges will be crucial for deploying RKD in real-world FL environments, where both the data distributions and threat models evolve continuously.

Chapter 5

A Comprehensive Defending Framework

This chapter introduces SD-CSFL, a synthetic-data-driven conformity-scoring defense that blocks both gradient-manipulation and backdoor attacks in federated learning by combining a privacy-preserving synthetic calibration set, entropy-based non-conformity scores, adaptive thresholds, and stratified sampling to filter malicious updates even under extreme heterogeneity. Section 5.2 formalizes the threat model and details entropy scoring on synthetic data, adaptive client thresholds, and balanced stratified sampling. Section 5.3 benchmarks SD-CSFL against IPM, ALiE, A3FL, F3BA, and CerP across varying attacker ratios and heterogeneity levels. Section 5.3.9 analyzes score distributions, threshold dynamics, and validates the privacy of the calibration data. Section 5.3.10 isolates the contribution of stratified sampling.

5.1 Introduction

Building upon the specific threat models addressed in Chapter 3 (gradient manipulation) and Chapter 4 (backdoor attacks), this chapter introduces a unified and adaptable defense framework—Synthetic Data-Driven Conformity Scoring for

Federated Learning (SD-CSFL). SD-CSFL is designed to detect a wide spectrum of adversarial behaviors while preserving privacy and robustness under Non-IID data distributions.

Unlike prior defenses such as FLTrust [13] and SageFlow [53], which rely on trusted proxy datasets, SD-CSFL leverages independently generated synthetic data to evaluate the trustworthiness of client models via entropy-based nonconformity scoring. This approach avoids assumptions of external data availability and better supports privacy-preserving deployments.

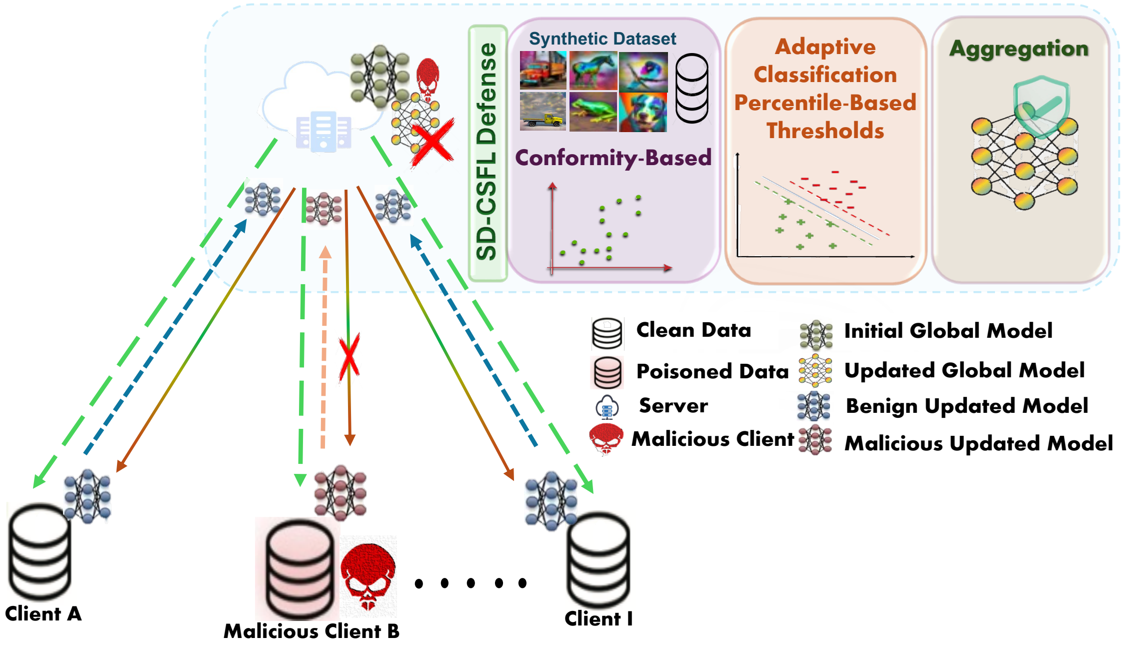


Figure 5.1: Overview of the Synthetic Data-Driven Conformity Scoring framework for Federated Learning (SD-CSFL).

As shown in Figure 5.1, SD-CSFL filters potentially malicious updates using adaptive percentile-based thresholds. It further incorporates stratified sampling to ensure calibration data diversity, which is essential in heterogeneous settings.

Experiments on CIFAR-10 and Birds datasets confirm SD-CSFL's effectiveness, showing up to a 35% gain in accuracy and an 80% reduction in backdoor success rates under strong Non-IID poisoning scenarios.

The rest of this chapter details the SD-CSFL framework, including its entropy-based scoring, synthetic data generation, and experimental validation against state-of-the-art defenses.

5.2 Methodology

In this section, we introduce the Synthetic Data-Driven Conformity Scoring for Federated Learning (SD-CSFL) framework, a robust defense against both gradient manipulation and backdoor attacks. SD-CSFL relies on an independently generated, privacy-preserving calibration dataset and an entropy-based nonconformity scoring mechanism to detect and exclude malicious client contributions. By employing adaptive thresholding and stratified sampling, SD-CSFL effectively limits adversarial influence even under Non-IID data distributions.

5.2.1 Overview of the SD-CSFL Framework

Figure 5.2 provides a visual overview of the SD-CSFL framework and highlights its key components. The process begins with each client sending its locally trained model to the central server. The server evaluates each model using a synthetic calibration dataset, computing entropy-based nonconformity scores to estimate the reliability of client updates. These scores are plotted and assessed using percentile-based adaptive thresholds, which define a classification band that separates potentially benign from potentially malicious updates.

Clients falling outside this threshold band are flagged as malicious, while those within it are deemed benign. Only the updates from benign clients are aggregated to produce the next version of the global model. Malicious clients are excluded from receiving the global model or may receive a perturbed version.

This Figure 5.2 underscores the contribution of SD-CSFL: it introduces a systematic, data-driven, and privacy-preserving approach to secure aggregation in FL by integrating synthetic calibration, entropy-based scoring, and adaptive client selection.

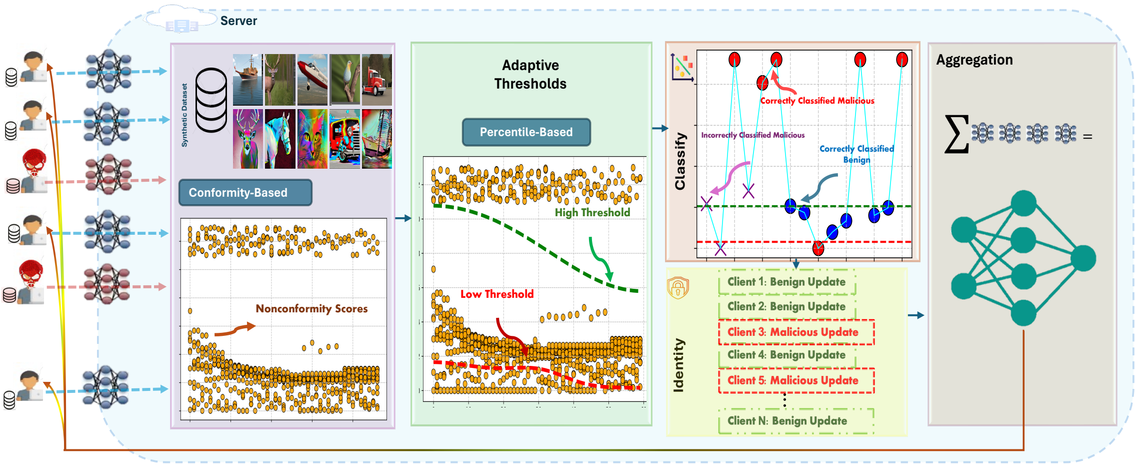


Figure 5.2: Illustration of the SD-CSFL workflow. Each client's model is evaluated on a clean synthetic calibration dataset to compute entropy-based nonconformity scores, which quantify prediction uncertainty and deviation from expected behavior. These scores are processed using adaptive percentile-based thresholds to classify clients as either benign or malicious. Only updates from clients deemed benign are aggregated into the global model, which is then redistributed exclusively to the benign clients, enhancing the robustness and security of the FL process.

Initially, in the first round ($r = 0$), the server distributes the global model $\mathbf{M}^0_{\text{global}}$ to all clients $\mathcal{C}$ for local training. Each client i trains on its local dataset and sends an updated model $\mathbf{M}^r_i$ back to the server. In subsequent rounds ($r > 0$), the server distributes the updated global model $\mathbf{M}^r_{\text{global}}$ only to clients classified as potentially benign in the previous round, denoted as $\mathcal{B}^{r-1}$.

To assess the reliability of received local updates, the server computes a nonconformity score for each client model $\mathbf{M}^r_i$ using an entropy-based function on a synthetic calibration dataset $\mathcal{D}_{\text{calibration}}$:

$$\mathbf{Score}^r_i = f_{\text{entropy}}\big(\mathbf{M}^r_i, \mathcal{D}_{\text{calibration}}\big). \tag{5.1}$$

An adaptive percentile-based thresholding mechanism then classifies clients as potentially benign $\mathcal{B}^r$ or potentially malicious $\mathcal{M}^r$:

$$\mathcal{B}^r, \mathcal{M}^r = \text{Classify}\Big(\{\mathbf{Score}^r_i\}_{i\in\mathcal{C}}\Big). \tag{5.2}$$

The server aggregates updates from the set $\mathcal{B}^r$ of potentially benign clients to form the next global model:

$$\mathbf{M}_{\text{global}}^{r+1} = \text{Aggregate}\Big(\{\mathbf{M}_i^r \mid i \in \mathcal{B}^r\}\Big). \tag{5.3}$$

Clients classified as potentially malicious $\mathcal{M}^r$ are excluded from receiving the updated global model in the subsequent rounds. Instead, they continue training their local models and remain engaged in the training process until they are reclassified as benign.

Alternative Approach: Perturbed Global Model (PGM). Rather than outright exclusion, SD-CSFL can send a perturbed version of the global model $\mathbf{M}_{\text{global}}^{r,\text{perturbed}}$ to clients in $\mathcal{M}^r$. This perturbed model incorporates minimal noise (e.g., scale $= 1 \times 10^{-4}$) to obscure precise model parameters, limiting exploitation by malicious clients while preventing them from discerning that they have been flagged. This strategy maintains the participation of clients in training and ensures that benign clients mistakenly classified as malicious can still benefit once reclassified.

This iterative process repeats for R rounds. Detailed operations of the framework are provided in Algorithm 8.

Computing Nonconformity Score. We compute nonconformity scores for client models using a synthetic dataset. First, the synthetic dataset $\mathcal{D}_{\text{calibration}}$ is prepared, providing a controlled environment for evaluating client models and ensuring independence from potentially compromised client data. A balanced calibration set $\mathcal{L}_{\text{balanced}}$ is created using the procedure discussed later. This step mitigates the effects of class imbalance within the synthetic dataset.

For each client model $\mathbf{M}_i^r \in \mathbf{M}_{\mathcal{C}}^r$, we compute a nonconformity score. For each batch $(\mathbf{X}, \mathbf{y})$ in the balanced calibration set $\mathcal{L}_{\text{balanced}}$, the model $\mathbf{M}_i^r$ produces output probabilities $\mathbf{P} \in \mathbb{R}^{n \times K}$. The entropy H_j for each sample j is then calculated as:

$$H_j = -\sum_{k=1}^{K} \mathbf{P}_{jk} \log(\mathbf{P}_{jk}), \tag{5.4}$$

where K is the number of classes and $\mathbf{P}_{jk}$ denotes the predicted probability for class k of sample j.

Algorithm 8 SD-CSFL Framework Methodology

```
Require: Set of clients $\mathcal{C}$, number of rounds $R$, synthetic calibration dataset $\mathcal{D}_{\text{calibration}}$
Ensure: Final global model $\mathbf{M}_{\text{global}}^{R}$
Initialize the global model $\mathbf{M}_{\text{global}}^{0}$
for $r = 0$ to $R - 1$ do
    if $r = 0$ then
        Send $\mathbf{M}_{\text{global}}^{0}$ to all clients $i \in \mathcal{C}$
    else
        Send $\mathbf{M}_{\text{global}}^{r}$ to clients classified as potentially benign $i \in \mathcal{B}^{r-1}$
    end if
    for each client $i \in \mathcal{C}$ do
        Collect $\mathbf{M}_{i}^{r}$
    end for
    for each client $i \in \mathcal{C}$ do
        Compute nonconformity score: $\text{Score}_{i}^{r} = f_{\text{entropy}}\left(\mathbf{M}_{i}^{r}, \mathcal{D}_{\text{calibration}}\right)$
    end for
    Classify clients: $\mathcal{B}^{r}, \mathcal{M}^{r} = \text{Classify}\left(\{\text{Score}_{i}^{r}\}_{i \in \mathcal{C}}\right)$
    Aggregate models: $\mathbf{M}_{\text{global}}^{r+1} = \text{Aggregate}\left(\{\mathbf{M}_{i}^{r} \mid i \in \mathcal{B}^{r}\}\right)$
end for
return Final global model $\mathbf{M}_{\text{global}}^{R}$
```

The mean entropy $\bar{H}_l$ for batch l is computed as:

$$\bar{H}_l = \frac{1}{n} \sum_{j=1}^{n} H_j, \tag{5.5}$$

where n is the number of samples in the batch.

After processing all m batches for a model, the nonconformity score s_i^r for model $\mathbf{M}_i^r$ is computed as the average of the batch mean entropies:

$$s_i^r = \frac{1}{m} \sum_{l=1}^{m} \bar{H}_l. \tag{5.6}$$

The computed nonconformity scores for all clients in round r are denoted as:

$$\mathbf{Score}^r_{\mathcal{C}} = \{s^r_i \mid i \in \mathcal{C}\}. \tag{5.7}$$

The full scoring procedure is detailed in Algorithm 9.

Algorithm 9 Computing Nonconformity Score

```
Require: Synthetic dataset $\mathcal{D}_{\text{calibration}}$, list of client models $\mathbf{M}^r_{\mathcal{C}}$, batch size $b$
Ensure: List of clients' nonconformity scores $\mathbf{Score}^r_{\mathcal{C}}$
Create balanced calibration set $\mathcal{L}_{\text{balanced}}$.
for each model $\mathbf{M}^r_i \in \mathbf{M}^r_{\mathcal{C}}$ do
    for each batch $(\mathbf{X}, \mathbf{y}) \in \mathcal{L}_{\text{balanced}}$ do
        Obtain model outputs and compute probabilities $\mathbf{P}$
        for each sample $j = 1$ to $n$ do
            Compute entropy: $H_j \leftarrow -\sum_{k=1}^{K} \mathbf{P}_{jk} \log(\mathbf{P}_{jk})$
        end for
        Compute mean entropy for the batch: $\bar{H}_l \leftarrow \frac{1}{n}\sum_{j=1}^{n} H_j$
    end for
    Compute nonconformity score: $\mathbf{Score}^r_i \leftarrow \frac{1}{m}\sum_{l=1}^{m} \bar{H}_l$
end for
return $\mathbf{Score}^r_{\mathcal{C}}$
```

Balanced Calibration Set Method. To address class imbalance, which can adversely affect the reliability of nonconformity scores, we ensure equal representation of each class in the calibration set using stratified sampling. We begin by analyzing the class distribution in the calibration dataset $\mathcal{D}_{\text{calibration}}$, determining the count c_k for each class $k \in \{1, \dots, K\}$. We then compute the class weight:

$$w_k = \frac{1}{c_k}, \tag{5.8}$$

giving higher weight to underrepresented classes.

Each sample $(\mathbf{x}_i, y_i) \in \mathcal{D}_{\text{calibration}}$ is then assigned a sampling score based on the weight of its class label:

$$s_i = w_{y_i}, \tag{5.9}$$

which reflects the relative importance of its class in the sampling process.

We apply weighted sampling using the scores $\{s_i\}$ to construct a balanced calibration subset:

$$\mathcal{L}_{\text{balanced}} \subset \mathcal{D}_{\text{calibration}}, \tag{5.10}$$

such that the resulting class distribution is approximately uniform. This balanced calibration set is crucial for ensuring that the entropy-based nonconformity scores (see Equation 5.6) are not biased due to over- or under-representation of particular classes.

Nonconformity Score-Based Classification. Let $\mathbf{Score}_{\mathcal{C}}^r = \{s_1^r, s_2^r, \ldots, s_{|\mathcal{C}|}^r\}$ represent the nonconformity scores of all clients at round r. The server uses a user-defined false-positive (FP) budget $\delta \in (0, 1)$ to set symmetric percentile thresholds:

$$p_{\text{low}} = \frac{\delta}{2}, \quad p_{\text{high}} = 1 - \frac{\delta}{2}, \tag{5.11}$$

and computes the corresponding threshold values:

$$\tau_{\text{low}}^r = Q_{p_{\text{low}}}(\mathbf{Score}_{\mathcal{C}}^r), \quad \tau_{\text{high}}^r = Q_{p_{\text{high}}}(\mathbf{Score}_{\mathcal{C}}^r), \tag{5.12}$$

where $Q_p(\cdot)$ denotes the empirical percentile function. Clients whose scores fall within the range

$$\tau_{\text{low}}^r \leq s_i^r \leq \tau_{\text{high}}^r \tag{5.13}$$

are classified as potentially benign ($\mathcal{B}^r$), while others are flagged as potentially malicious ($\mathcal{M}^r$). These thresholds are recalculated each round to maintain robustness under changing score distributions.

Synthetic Data. To construct our calibration dataset $\mathcal{D}_{\text{calibration}}$, we adopt a synthetic data generation approach inspired by prior work [30], leveraging Stable Diffusion-V2 [57] and ChatGPT-3.5 [51]. In our method, we specifically utilize

Algorithm 10 Client Classification Using Percentile-Based Nonconformity Scores

```
Require: Score vector $\mathbf{Score}_{\mathcal{C}}^{r}$, FP budget $\delta$
Ensure: Potentially benign set $\mathcal{B}^r$, potentially malicious set $\mathcal{M}^r$
Compute percentiles: $p_{\text{low}} \leftarrow \delta/2, \quad p_{\text{high}} \leftarrow 1 - \delta/2$
Compute adaptive thresholds: $\tau_{\text{low}} \leftarrow Q_{p_{\text{low}}}(\mathbf{Score}_{\mathcal{C}}^{r}), \quad \tau_{\text{high}} \leftarrow Q_{p_{\text{high}}}(\mathbf{Score}_{\mathcal{C}}^{r})$
Initialize: $\mathcal{B}^r \leftarrow \emptyset, \quad \mathcal{M}^r \leftarrow \emptyset$
for each client $i \in \mathcal{C}$ do
    if $\tau_{\text{low}} \leq s_i^r \leq \tau_{\text{high}}$ then
        $\mathcal{B}^r \leftarrow \mathcal{B}^r \cup \{i\}$
    else
        $\mathcal{M}^r \leftarrow \mathcal{M}^r \cup \{i\}$
    end if
end for
return $\mathcal{B}^r$, $\mathcal{M}^r$
```

artistic synthetic data, generated using prompts enriched with keywords such as *non-photorealistic*, *exaggerated artistic effects*, and *bold brushstrokes.* These prompts are designed to create visually distinct and highly stylized representations of classes.

The use of such synthetic data introduces controlled abstraction and variability, making it ideal for computing nonconformity scores. This approach ensures a robust and domain-relevant evaluation process while preserving the privacy of client data distributions and avoiding reliance on potentially compromised client data.

5.3 Experiments

In this section, we simulate a federated learning environment in which multiple clients collaboratively train a global model under the coordination of a central aggregator until convergence. Our goal is to demonstrate the effectiveness of **SD-CSFL** against both gradient manipulation and backdoor attacks, even under Non-IID conditions and various adversarial participation rates.

5.3.1 Datasets and Models

We conduct experiments on *CIFAR-10* and *Birds*, along with their synthetic counterparts, *CIFAR-10-Synth* and *Birds-Synth* [30].

CIFAR-10. We use a CNN with three convolutional layers having 32, 64, and 128 filters, respectively. Each convolutional layer is followed by batch normalization, ReLU activation, and MaxPooling. A fully connected (FC) layer with 256 units, ReLU activation, and 0.25 dropout precedes the final output layer, which has 10 classes. Training uses a batch size of 64, a learning rate of 0.01, and 50,000 training samples [14]. Server-side calibration is performed on 14,523 samples from CIFAR-10-Synth.

Birds. We adopt a *ResNet50* model, pre-trained and fine-tuned on `layer4`, with a 1024-unit FC classifier followed by batch normalization, ReLU activation, 0.5 dropout, and an FC layer for 525 classes. Training uses a batch size of 16, a learning rate of 0.001, and 84,635 training samples [58]. Server-side calibration is performed on 20,475 samples from Birds-Synth.

We select CIFAR-10 and Birds—along with their synthetic counterparts—for three key reasons. First, this combination provides a broad spectrum of visual complexity: CIFAR-10 offers low-resolution, general-purpose images across 10 classes, while Birds presents a high-resolution, fine-grained classification challenge spanning 525 classes. This contrast allows us to evaluate the scalability and generalizability of SD-CSFL under both standard and real-world conditions. Second, the Birds dataset introduces naturally imbalanced and heterogeneous class distributions, better simulating the data disparities encountered in practical FL deployments. Third, both datasets are paired with synthetic counterparts (CIFAR-10-Synth and Birds-Synth [30]), which enable server-side calibration without accessing private client data—a core requirement of our privacy-preserving defense. These datasets thus provide a suitable and realistic testbed for evaluating robustness under diverse attack strategies and data modalities.

We use CNNs tailored to each dataset's complexity. CIFAR-10 employs a lightweight

3-layer CNN architecture to allow controlled experimentation and comparability with related work, while the Birds dataset uses a fine-tuned ResNet50 to handle its high intra-class variance and deeper feature hierarchies. This setup ensures the defense is stress-tested across both constrained and resource-rich federated environments.

5.3.2 Attack Setup

We simulate 30 clients in total, with 20%, 40%, and 60% of them being adversarial. In backdoor scenarios, each malicious client poisons 50% of its local training data by embedding backdoor triggers, then trains its model following its designated backdoor strategy. In gradient manipulation scenarios, attackers alter gradients' directions or magnitudes to degrade or steer the global model.

5.3.3 Threat Model

We consider a realistic federated learning environment in which a subset of clients is adversarial. These malicious participants aim to compromise the global model either by manipulating gradients (model poisoning) or by injecting malicious patterns into their local data (backdoor attacks). The server is honest-but-curious: it follows the protocol but has no prior knowledge of which clients are compromised and no access to raw client data. Its only tools for defense are the received model updates and a synthetic calibration dataset.

The attackers operate independently and without coordination. They do not know the updates of benign clients but can carefully craft their submissions to evade standard detection—particularly challenging under Non-IID data distributions, where natural variability between client updates masks adversarial anomalies.

To rigorously evaluate the robustness of **SD-CSFL**, we consider five sophisticated and representative attacks:

- **IPM** [69]: A gradient manipulation attack where malicious clients align their updates in a specific direction to disrupt convergence while remaining statistically benign.

- **ALiE** [7]: Adversaries craft updates constrained within the statistical range (e.g., mean and standard deviation) of benign gradients, making detection based on deviation or norm ineffective.
- **A3FL** [76]: An adaptive backdoor attack that leverages iterative optimization (e.g., PGD) to design highly effective and stealthy trigger patterns.
- **F3BA** [18]: A focused-flip attack that manipulates selected model weights to inject backdoors while preserving overall update structure.
- **CerP** [43]: A parameter perturbation attack that subtly alters weights to embed backdoors, aiming for minimal detectable deviation from benign models.

These diverse attack strategies ensure that SD-CSFL is evaluated under both stealthy and aggressive adversarial behaviors, across both model- and data-level threats.

5.3.4 Heterogeneous Setting

We evaluate the robustness of **SD-CSFL** under Non-IID data distributions by sampling client datasets via a Dirichlet distribution [27]. By varying the Dirichlet parameter α, we control the degree of heterogeneity: $\alpha = 0.9$ for moderate skew and $\alpha = 0.5$ for highly imbalanced distributions.

5.3.5 Baselines

To contextualize SD-CSFL's performance, we compare it against a range of existing FL defenses:

RFCL (proposed in Chapter 3) and **FLAME** [50]: Clustering-based defenses relying on outlier detection in high-dimensional parameter spaces.

RKD (discussed in Chapter 4), along with **FedDF** [37], **FedBE** [15], and **FedRAD** [60]: Knowledge-distillation methods offering varying degrees of resistance to malicious updates.

RLR [52]: A robust learning-rate adjustment method.

Median, **Mkrum** [11], and **CC** [28]: Classic robust aggregation mechanisms that filter or reweight suspicious updates.

5.3.6 Evaluation Metrics

We evaluate defense performance using two key metrics: *Main Task Accuracy (MTA)* and *Attack Success Rate (ASR)*.

MTA measures the model's accuracy on clean test data $\mathcal{D}_m$: $MTA = \frac{\left|\{x \in \mathcal{D}_m \mid f(x)=y\}\right|}{\left|\mathcal{D}_m\right|}$, where $f(x)$ is the model prediction and y is the true label.

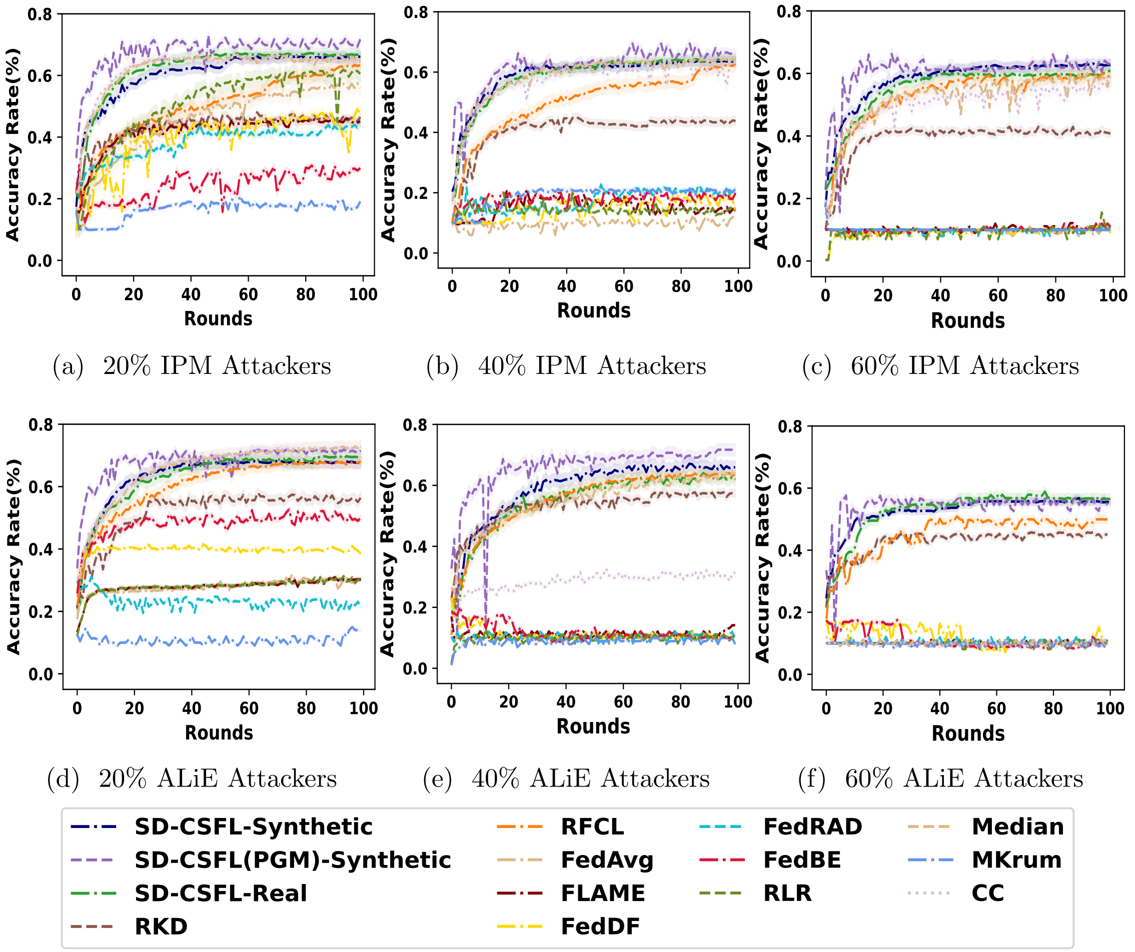
ASR quantifies the success of backdoor triggers using poisoned test data $\mathcal{D}_b$: $ASR = \frac{\left|\{x \in \mathcal{D}_b \mid f(x)=y_{\text{target}}\}\right|}{\left|\mathcal{D}_b\right|}$, with y_{target} denoting the attacker-specified label.

5.3.7 Experimental Results

We present empirical results of our SD-CSFL framework tested against gradient manipulation and backdoor attacks in FL on the CIFAR-10 dataset under Non-IID conditions ($\alpha = 0.9$) and ($\alpha = 0.5$). Experiments were repeated five times, and results demonstrate statistical significance.

Effective Defense Against IPM and ALiE Attacks. Figures 5.3 and 5.4 illustrate SD-CSFL's performance against IPM and ALiE attacks on CIFAR-10 and Birds datasets. SD-CSFL outperforms baselines like FedAvg, FLAME, RFCL, RKD and Median, maintaining higher accuracy as the proportion of compromised clients increases.

For CIFAR-10, both synthetic and real calibration datasets show slightly smaller performance compared to other scenarios. The synthetic calibration dataset, enriched with carefully designed attributes and variability, enhance the detection of malicious behaviors, they tend to underperform slightly relative to the real



(a) 20% IPM Attackers

(b) 40% IPM Attackers

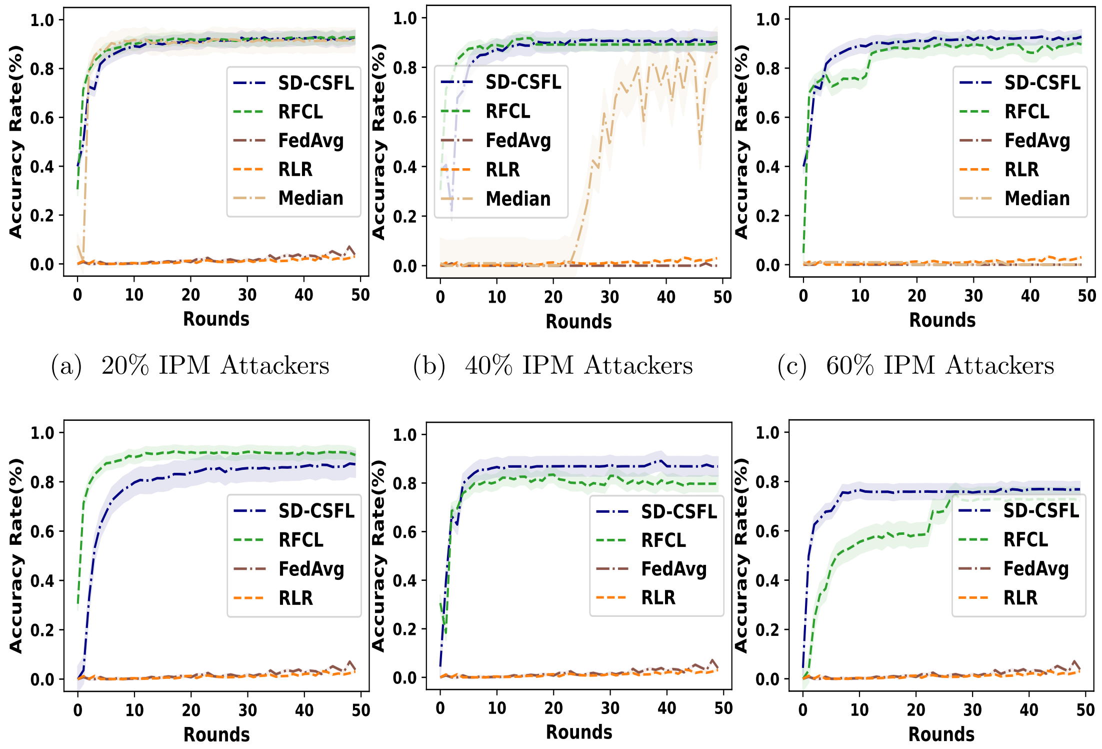
(c) 60% IPM Attackers

(d) 20% ALiE Attackers

(e) 40% ALiE Attackers

(f) 60% ALiE Attackers

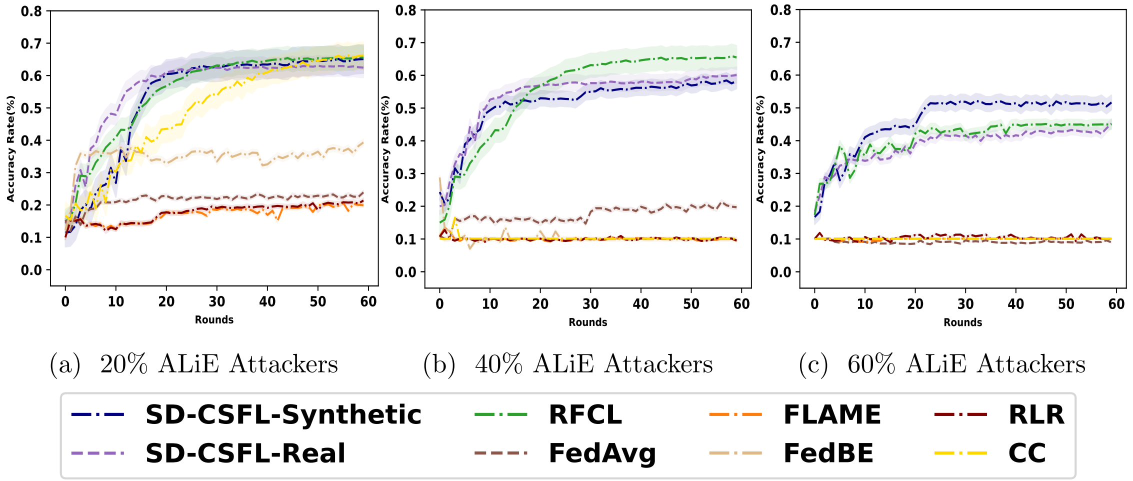
Figure 5.3: Performance of baselines and SD-CSFL on CIFAR-10 against IPM and ALiE attack under Non-IID ($\alpha = 0.9$).



(a) 20% IPM Attackers (b) 40% IPM Attackers (c) 60% IPM Attackers

(d) 20% ALiE Attackers (e) 40% ALiE Attackers (f) 60% ALiE Attackers

Figure 5.4: Performance of baselines and SD-CSFL on Birds against IPM and ALiE attacks under Non-IID ($\alpha = 0.9$).



(a) 20% ALiE Attackers (b) 40% ALiE Attackers (c) 60% ALiE Attackers

Figure 5.5: Performance of baselines and SD-CSFL on CIFAR-10 against ALiE attack under Non-IID ($\alpha = 0.5$).

calibration dataset. The latter achieves higher accuracy due to its closer alignment with the target domain.

The perturbed global model (PGM) in SD-CSFL introduces minor performance fluctuations due to the added stochastic noise. However, this trade-off significantly enhances robustness by limiting exploitation from malicious clients while maintaining their participation in training.

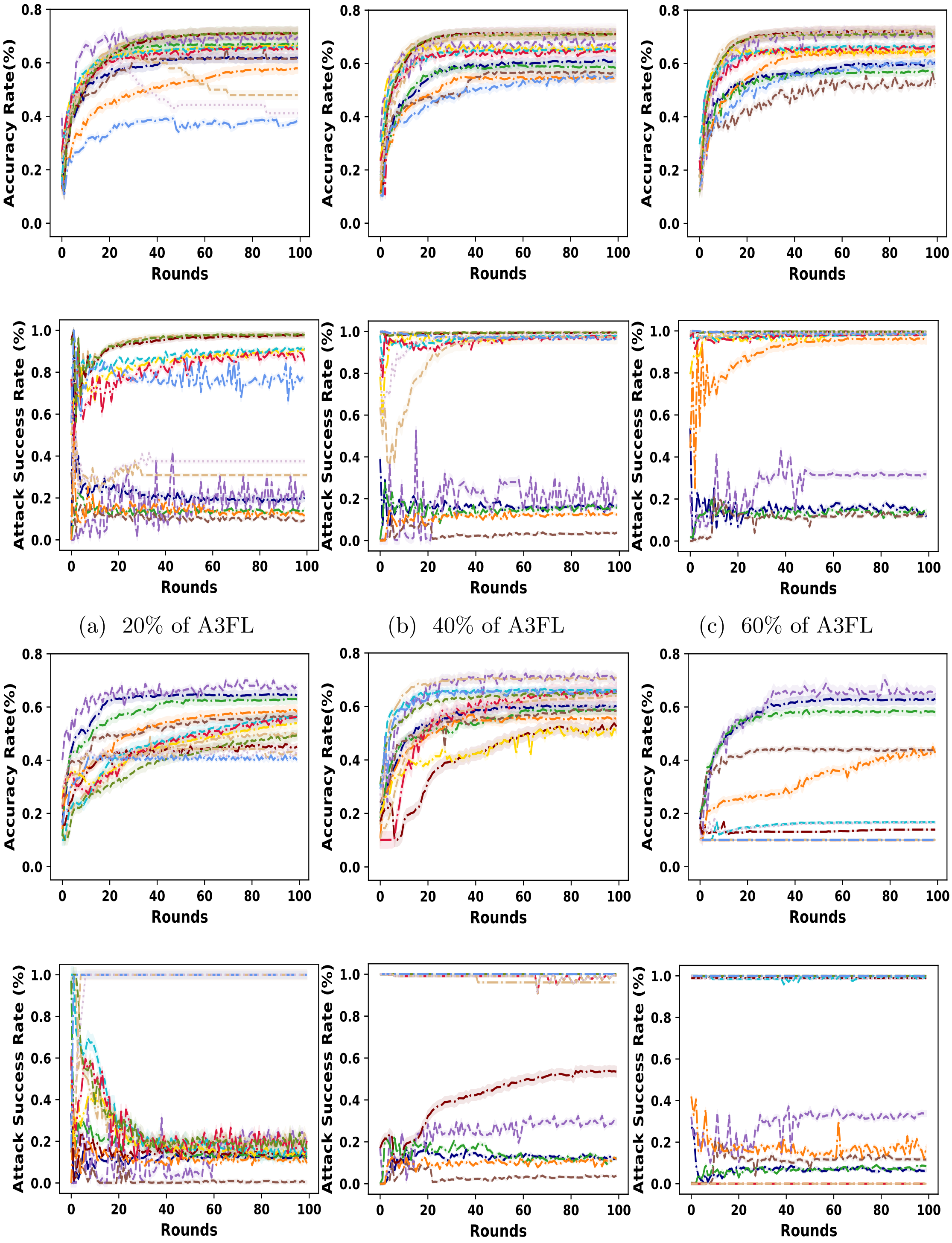
To further examine ALiE attacks under more pronounced Non-IID settings ($\alpha = 0.5$), Figure 5.5 shows that SD-CSFL preserves strong resilience despite the increased intensity of adversarial noise. Adaptive thresholding and entropy-based scoring work jointly to identify and neutralize even subtle gradient manipulations.

Effective Defense Against A3FL, F3BA, and CerP Attacks.

Figures 5.6 and 5.7 highlight SD-CSFL's defense against A3FL, F3BA, and CerP backdoor attacks on CIFAR-10 and Birds datasets. SD-CSFL consistently outperforms baselines like FedAvg, FLAME, RFCL, and Median, maintaining higher accuracy and significantly lower attack success rates, even as the proportion of compromised clients increases.

In A3FL attacks, where adaptive strategies modify triggers to blend into the global model, SD-CSFL demonstrates robust defenses by leveraging entropy-based nonconformity scores to identify and exclude suspicious updates from aggregation. For F3BA and CerP attacks, SD-CSFL achieves low attack success rates across all levels of compromised clients, as shown in Figures 5.6 and 5.7. These results emphasize SD-CSFL's effectiveness in detecting and isolating malicious updates, preventing them from influencing the global model. The framework maintains consistent performance trends across varying attack strategies, with synthetic and real calibration datasets both showing strong results. Real datasets, however, exhibit slightly better accuracy due to their domain relevance.

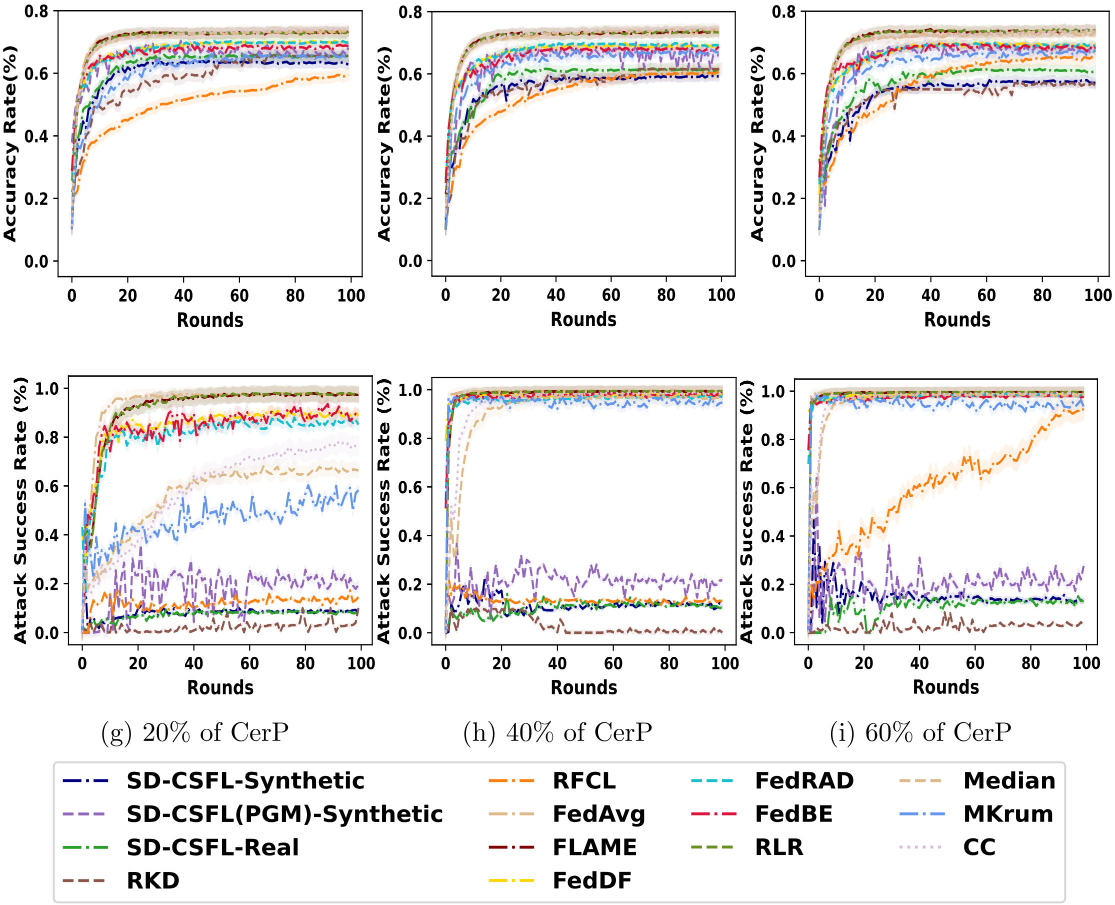
For CIFAR-10, incorporating the perturbed global model (PGM) into SD-CSFL introduces slight performance fluctuations. This affects the model's performance by marginally increasing the attack success rate in certain cases. Despite this impact,



(a) 20% of A3FL (b) 40% of A3FL (c) 60% of A3FL

(d) 20% of F3BA (e) 40% of F3BA (f) 60% of F3BA

Figure 5.6: Performance of baselines and SD-CSFL on CIFAR-10 against A3FL and F3BA attacks under Non-IID ($\alpha = 0.9$), Part 1.



(g) 20% of CerP

(h) 40% of CerP

(i) 60% of CerP

Figure 5.6: (Continued) Performance of baselines and SD-CSFL on CIFAR-10 against CerP attack under Non-IID ($\alpha = 0.9$).

SD-CSFL maintains competitive model performance and continues to outperform baseline methods in mitigating advanced backdoor attacks.

Figure 5.8 illustrates SD-CSFL's robustness under A3FL [76] attacks on the CIFAR-10 dataset in challenging Non-IID settings ($\alpha = 0.5$). SD-CSFL consistently outperforms baseline defenses by maintaining stable accuracy and suppressing attack success rates across varying proportions of compromised clients.

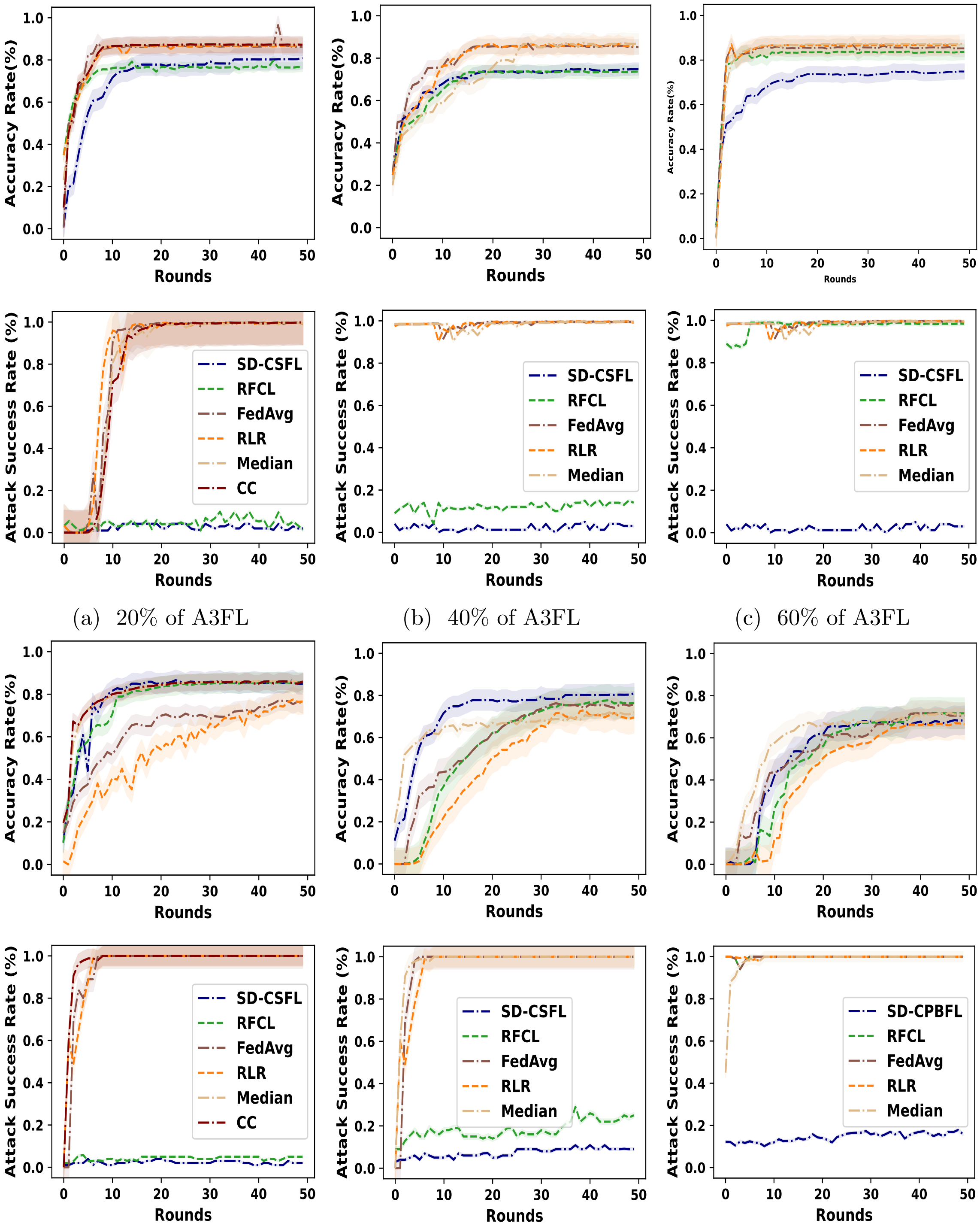
As the proportion of adversarial clients increases from 20% to 60%, SD-CSFL maintains competitive accuracy: around 0.6 for 20%, above 0.5 for 40%, and above 0.4 even under 60% attack intensity. These results reflect SD-CSFL's ability to preserve model performance despite increasing adversarial pressure.

Attack success rates remain well-controlled. Under a 20% A3FL attack, the success rate stays below 0.2, indicating effective suppression of backdoor activation. Even as attack intensity rises to 40% and 60%, the success rate remains bounded—rising to approximately 0.4 and peaking around 0.6—yet the global model is never fully compromised. This demonstrates SD-CSFL's capacity to withstand even aggressive adaptive backdoor strategies.

These findings validate the role of entropy-based nonconformity scoring and adaptive percentile thresholding in isolating suspicious updates. Together with a balanced synthetic calibration set, these mechanisms enable SD-CSFL to effectively defend the global model in Non-IID federated settings.

5.3.8 Scalability Analysis

As shown in Table 5.1, SD-CSFL's defense time is 19.45 seconds—substantially faster than FLAME (37.06s), FedDF (141.71s), and comparable to RKD (42.03s), while offering strong defense capabilities. Although RLR and RFCL achieve shorter runtimes, they do so at the cost of reduced robustness: RLR relies on simplistic learning rate adjustments, and RFCL depends on clustering in high-dimensional parameter space. RKD improves scalability by converting model updates into scalar similarity scores before clustering. Overall, SD-CSFL maintains a favorable balance



(a) 20% of A3FL

(b) 40% of A3FL

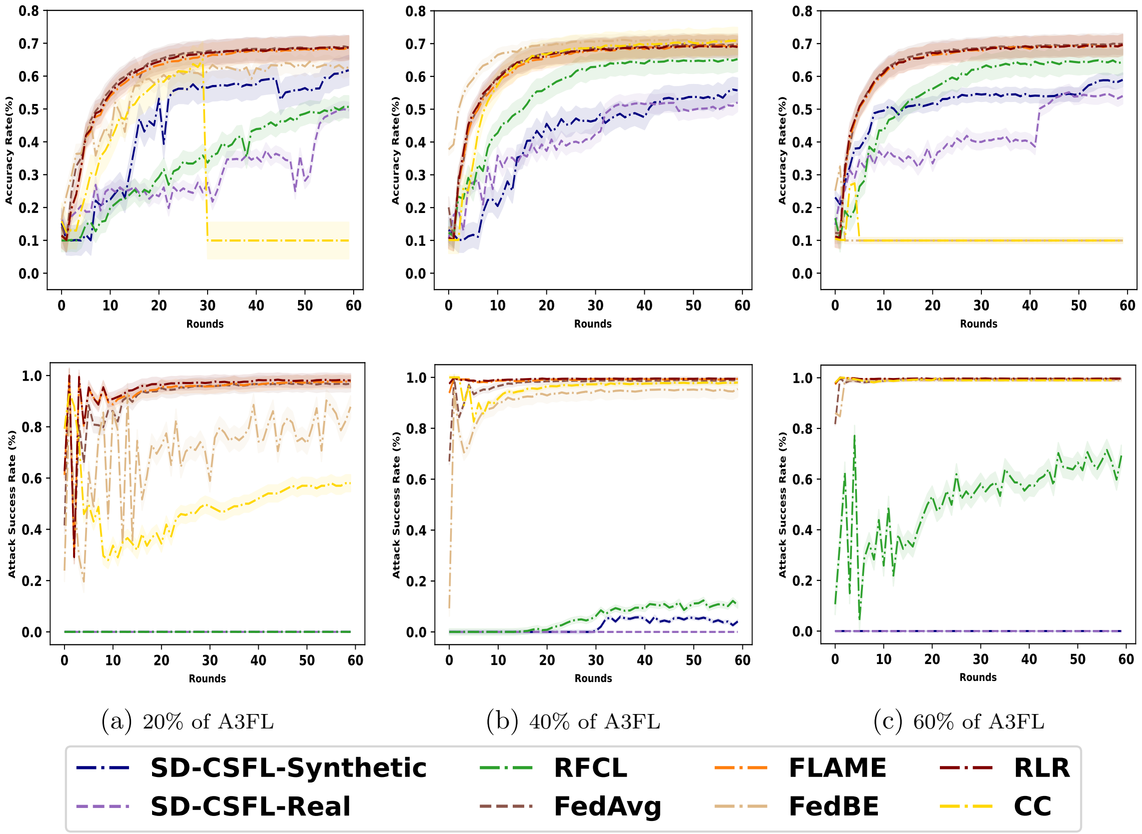
(c) 60% of A3FL

(d) 20% of F3BA

(e) 40% of F3BA

(f) 60% of F3BA

Figure 5.7: Performance of baselines and SD-CSFL on Birds against A3FL and F3A attacks under Non-IID ($\alpha = 0.9$).



(a) 20% of A3FL

(b) 40% of A3FL

(c) 60% of A3FL

Figure 5.8: Performance of baselines and SD-CSFL on CIFAR-10 against A3FL attack under Non-IID ($\alpha = 0.5$).

between computational efficiency and adversarial resilience.

The SD-CSFL measured runtime includes only the online defense components: entropy-based scoring, percentile thresholding, and client update filtering. Synthetic data generation is performed offline and excluded from the reported timing.

Table 5.1: Defense time comparison (in seconds) across methods. Methods proposed in this thesis are marked as (ours).

Method	**Defense Time (s)**
RLR [52]	0.02
RFCL (ours, see Chapter 3)	2.75
SD-CSFL (ours, see Chapter 5)	19.45
FLAME [50]	37.06
RKD (ours, see Chapter 4)	42.03
FedDF [37]	141.71

5.3.9 Experimental Analysis of Attack Detection and Validation

We analyzed SD-CSFL's effectiveness in detecting gradient manipulation and backdoor attacks.

5.3.9.1 Detection of Gradient Manipulation Attacks

Gradient manipulation attacks, such as IPM and ALiE, impact nonconformity scores differently. ALiE attacks inject high-variance noise into gradients, significantly increasing prediction uncertainty and elevating entropy beyond the upper threshold τ_{high}. IPM attacks manipulate the inner product between gradients and true updates. Although these subtle perturbations are designed to mimic benign updates, they introduce detectable deviations in nonconformity scores that may breach τ_{low} or τ_{high}, depending on the intensity of the manipulation.

ALiE Attacks: ALiE attacks introduce high-variance noise, disrupting predictions and increasing entropy. For a sample x, the entropy $H(x, \mathbf{M}_i^r)$ is defined as $H(x, \mathbf{M}_i^r) = -\sum_{k=1}^{K} P(y = k \mid x, \mathbf{M}_i^r) \log P(y = k \mid x, \mathbf{M}_i^r)$, where K is the number of classes. When predictions become uniformly distributed ($P(y = k) \approx \frac{1}{K}$), $H(x, \mathbf{M}_i^r)$ approaches $\log K$, representing maximum uncertainty. The mean entropy across the calibration dataset $\mathcal{D}_{\text{calibration}}$, $\mathbf{Score}_i^r = \frac{1}{|\mathcal{D}_{\text{calibration}}|} \sum_{x \in \mathcal{D}_{\text{calibration}}} H(x, \mathbf{M}_i^r)$, also converges to $\log K$ under high-variance noise. Since SD-CSFL dynamically adjusts percentile-based thresholds τ_{low} and τ_{high} with $\tau_{\text{high}} \leq \log K$, ALiE attacks that elevate $\mathbf{Score}_i^r$ toward $\log K$ inevitably exceed τ_{high} and are flagged as malicious. Figure 5.9(a) illustrates how ALiE increases entropy, enabling SD-CSFL to detect and isolate malicious updates.

IPM Attacks: IPM attacks subtly manipulate the inner product between gradients and true updates, introducing directional biases while preserving statistical similarity in other dimensions. These manipulations have minimal impact on predicted probabilities but cause subtle shifts in nonconformity scores. Specifically, scores often fall within the range: $\tau_{\text{low}} \leq \mathbf{Score}_i^r \leq \tau_{\text{high}}$. However, gradient biasing increases the likelihood of deviations that breach these thresholds. SD-CSFL detects such deviations dynamically by adapting τ_{low} and τ_{high} based on the distribution of nonconformity scores. Figure 5.9(b) illustrates how IPM attacks cluster scores near the thresholds, where careful evaluation enables detection.

5.3.9.2 Detection of Backdoor Attacks

Backdoor updates face conflicting optimization objectives as they aim to optimize for clean data while inducing targeted misclassifications. The total loss function for backdoor updates is: $L_{\text{total}}(\theta) = \lambda_{\text{clean}} L_{\text{clean}}(\mathcal{L}_{\text{clean}}, \theta) + \lambda_{\text{poisoned}} L_{\text{poisoned}}(\mathcal{L}_{\text{poisoned}}, \theta)$, where θ represents the model parameters, $\mathcal{L}_{\text{clean}}$ and $\mathcal{L}_{\text{poisoned}}$ are the clean and poisoned datasets, respectively, and L_{clean} and L_{poisoned} are the corresponding cross-entropy loss functions. The weights λ_{clean} and $\lambda_{\text{poisoned}}$ balance the optimization between clean and poisoned data.

This conflicting optimization introduces gradient misalignment, increasing uncertainty in predictions for clean data. The entropy $H(x, \mathbf{M}_i^r)$ for clean samples reflects this uncertainty. For backdoor updates, this misalignment increases entropy, affecting the decision boundary. Empirically, backdoor updates exhibit higher scores than benign ones, often exceeding the threshold τ_{high} in SD-CSFL. Figure 5.9(c), (d), and (e) confirms this behavior, showing higher nonconformity scores for backdoor updates compared to benign ones.

Percentile–Threshold Dynamics and Robustness. To determine whether a client is potentially benign or malicious in each round r, the server computes the nonconformity score s_i^r for each client $i \in \mathcal{C}$, forming the set $\mathbf{Score}_{\mathcal{C}}^r = \{s_1^r, s_2^r, \ldots, s_{|\mathcal{C}|}^r\}$. Clients whose scores fall within a central range are considered potentially benign, while those with unusually low or high scores are flagged as potentially malicious. This range is defined by two adaptive thresholds, τ_{low}^r and τ_{high}^r, computed as empirical quantiles over $\mathbf{Score}_{\mathcal{C}}^r$.

These percentiles are derived from a user-defined false-positive (FP) budget $\delta \in (0, 1)$, which specifies the maximum acceptable probability of incorrectly rejecting a benign client per round. The thresholds are set symmetrically as $p_{\text{low}} = \delta/2$ and $p_{\text{high}} = 1-\delta/2$. This approach is based on conformal prediction [4], which guarantees that, under the assumption of approximate exchangeability among benign clients, the per-round false-positive rate does not exceed δ, regardless of the underlying score distribution.

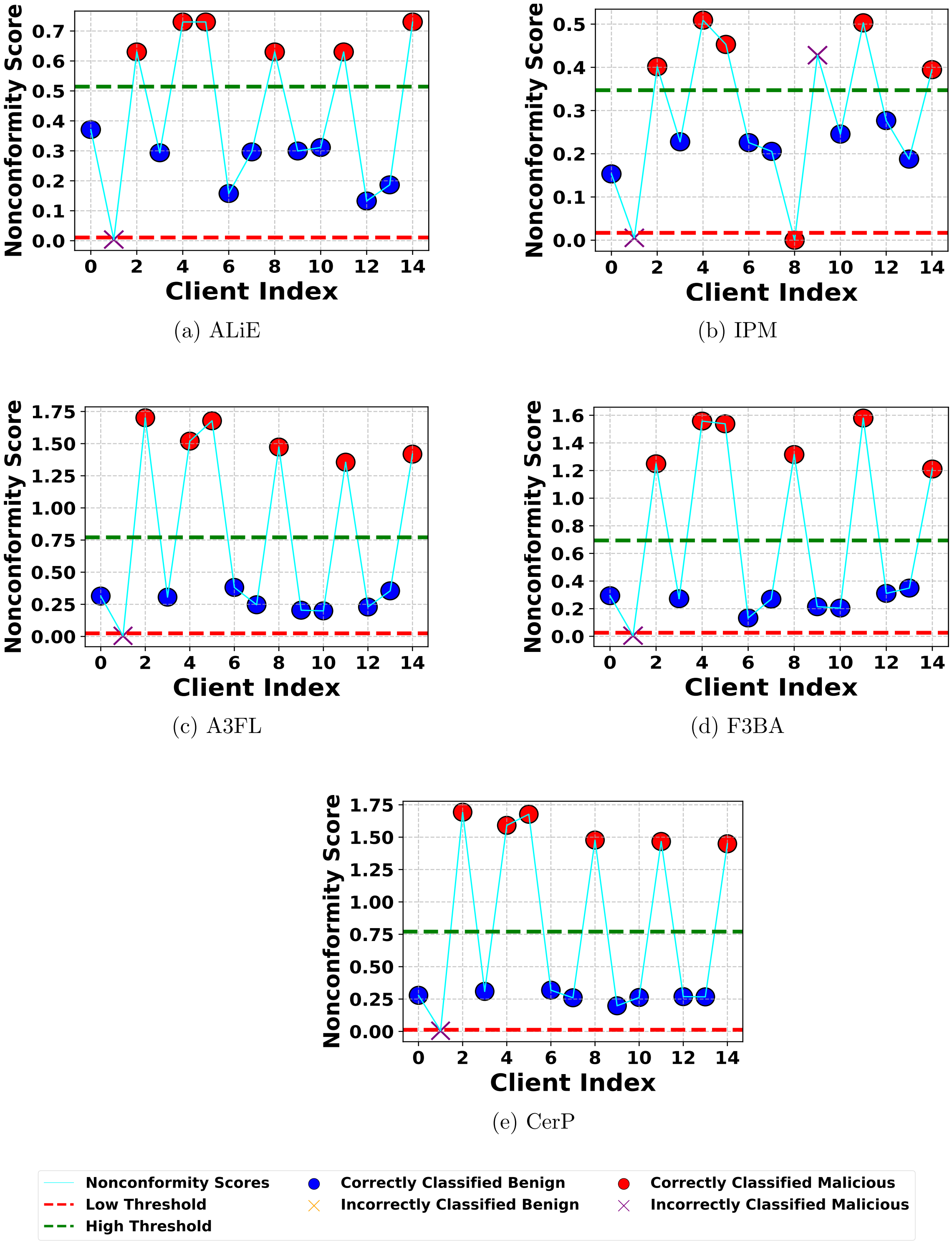
Since the thresholds are recomputed in every round, the acceptance band dynamically adapts to the evolving score distribution, enabling resilience to client drift, Non-IID heterogeneity, and adaptive attack patterns. We define true positives (TP) as malicious clients correctly flagged and excluded, and false positives (FP) as benign clients incorrectly rejected. Table 5.2 compares three settings of FP budget δ, highlighting the trade-off between detection and benign client retention. As δ increases, the acceptance band becomes narrower, improving TP rates (up to 100%) but also increasing FP. The setting $\delta = 0.60$ (band: 30–70%) offers a favorable balance and is used in all experiments.

Table 5.2: Impact of false-positive budget δ on classification performance under 40% compromised clients (CIFAR-10).

δ	Band (%)	Non-IID ($\alpha = 0.9$)			Non-IID ($\alpha = 0.5$)		
		FP (%)↓	TP (%)↑	F_1	FP (%)	TP (%)	F_1
0.25	13–88	8	75	0.80	11	56	0.65
0.60	30–70	**17**	**100**	**0.89**	**20**	**100**	**0.87**
0.75	38–63	27	100	0.83	31	100	0.81

5.3.9.3 Privacy Validation

We performed a privacy leak analysis to ensure that synthetic calibration samples do not expose sensitive information from real datasets. As shown in Figure 5.10(a), synthetic-to-real comparisons exhibit low cosine similarity, near-zero mutual information, and high KL divergence, confirming minimal alignment and distinct probability distributions. Real-to-real comparisons were performed between the full CIFAR-10 dataset and a subset comprising 5000 samples, evenly distributed across 10 classes (500 samples per class). These comparisons reveal high cosine similarity, low KL divergence, and high mutual information, demonstrating strong alignment within real data. In Figure 5.10(b), t-SNE plots display well-separated clusters for real and synthetic embeddings, reflecting their distinct distributions. Conversely, Figure 5.10(c) shows overlapping clusters in the real-to-subset comparison, indicating expected alignment within the same dataset. The representative samples in Figure 5.11 further highlight the unique visual properties of real and synthetic data, underscoring the fidelity and privacy-preserving attributes of the synthetic dataset.



(a) ALiE

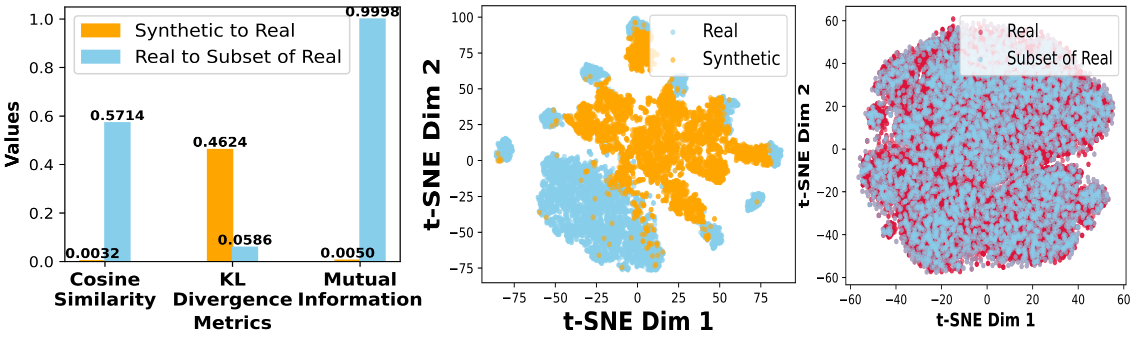
(b) IPM

(c) A3FL

(d) F3BA

(e) CerP

Figure 5.9: Distribution of scores for benign and malicious clients across various attack scenarios.



(a) Privacy metrics (b) t-SNE: Real to Synthetic (c) t-SNE: Real to Real

Figure 5.10: Privacy Metrics and Embedding Features Visualization for Real and Synthetic Data.

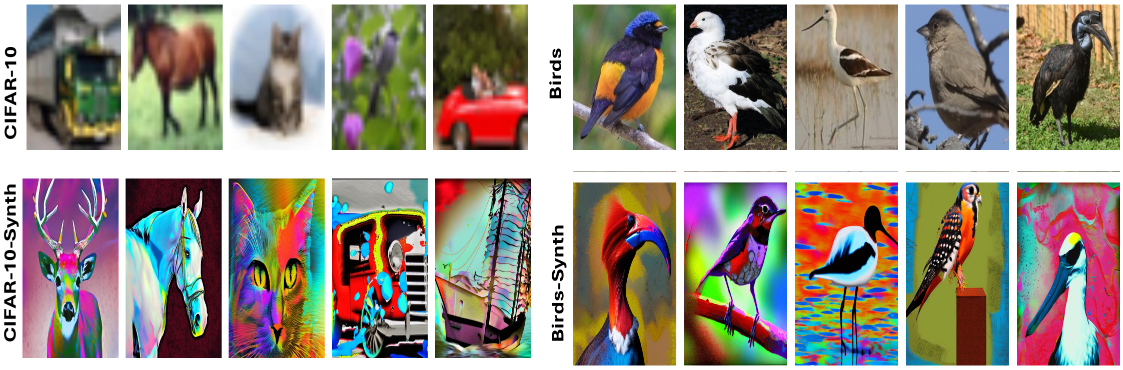


Figure 5.11: The first row displays images from the real CIFAR-10 and Birds datasets. The second row present synthetic samples from generated two datasets: CIFAR-10-Synth and Birds-Synth datasets.

5.3.10 Ablation Study

We conduct two ablation studies to evaluate design choices in SD-CSFL under 40% malicious clients in CIFAR-10.

Balanced vs. Non-Balanced Calibration Set. Figure 5.12 compares SD-CSFL's performance when using a balanced versus a non-balanced synthetic calibration set against A3FL and F3BA attackers. The balanced calibration set reflects the true class distribution, enabling SD-CSFL to more effectively distinguish benign from malicious updates. This results in higher accuracy and lower attack success rates. In contrast, the non-balanced set leads to less stable performance, with reduced accuracy and higher attack success, particularly under Non-IID conditions where class imbalance amplifies model drift. These results underscore the importance of designing a representative calibration set when applying nonconformity scoring.

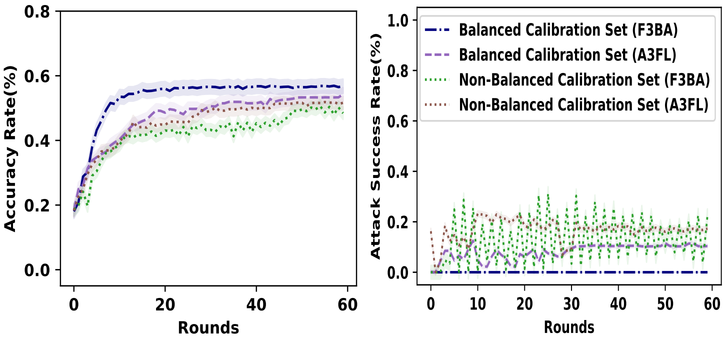


Figure 5.12: Impact of balanced vs. non-balanced calibration set under Non-IID ($\alpha = 0.9$).

Percentile vs. Median-Based Thresholding. To better understand the effectiveness of the percentile-based thresholding used in SD-CSFL, we compare it against a statistical approach based on the median and standard deviation. Specifically, the method defines an acceptance interval as median $\pm k\sigma$, where σ is the sample standard deviation of the nonconformity scores and k is a tunable parameter.

Table 5.3 presents a comparison between the median-based rule and the percentile-based acceptance band employed in SD-CSFL, under both moderate ($\alpha = 0.9$) and extreme ($\alpha = 0.5$) Non-IID conditions. We evaluate two representative values of k: 1.5 and 0.5. These are contrasted with the central 30–70% percentile band, which corresponds to a false-positive budget of $\delta = 0.60$.

The results show that the percentile-based method consistently achieves higher F_1 scores in both settings. Its thresholding relies entirely on rank statistics and is adaptively recalculated in each round, making it robust to asymmetric or heavy-tailed score distributions. In contrast, the performance of the median-based method deteriorates, particularly under extreme heterogeneity, indicating a sensitivity to score variability and outliers.

This difference in performance can be attributed to the underlying assumptions of each method. The median-based approach assumes a symmetric and well-behaved score distribution, relying on measures of central tendency and dispersion to define the threshold. Smaller values of k yield narrower intervals, increasing false positives, while larger values result in broader intervals that reduce sensitivity to anomalous behavior. In contrast, the percentile-based method is nonparametric and distribution-free, operating solely on rank statistics without assuming any specific distributional form. This makes it more suitable for the irregular and skewed score patterns commonly observed in adversarial federated learning scenarios.

Table 5.3: Comparison of percentile ($\delta = 0.60$) vs. median $\pm k\sigma$ thresholding under 40% compromised clients (CIFAR-10).

	Non-IID ($\alpha = 0.9$)			Non-IID ($\alpha = 0.5$)		
Threshold Rule	FP (%)↓	TP (%)↑	F_1	FP (%)↓	TP (%)↑	F_1
Median $\pm 1.5\sigma$	2	50	0.64	9	45	0.54
Median $\pm 0.5\sigma$	6	93	0.85	18	67	0.61
Percentile (30–70%)	**17**	**100**	**0.89**	**20**	**100**	**0.87**

5.3.11 Discussion and Limitations

Our experimental analysis demonstrates that the SD-CSFL framework robustly counters both gradient manipulation and backdoor attacks in federated learning. By leveraging a synthetically generated, privacy-preserving calibration dataset alongside an entropy-based nonconformity scoring mechanism, SD-CSFL effectively distinguishes between benign and malicious client updates even in challenging Non-IID environments. The adaptive thresholding and stratified sampling further enhance the framework's ability to maintain the integrity of the global model, ensuring a balance between security and performance.

Nonetheless, several limitations warrant discussion. First, the effectiveness of SD-CSFL is closely tied to the quality and representativeness of the synthetic calibration dataset. Generating high-fidelity synthetic data that accurately mirrors the diversity of real-world client data is a non-trivial task, and any shortcomings in the synthetic data could adversely affect the detection accuracy. Second, the adaptive thresholding mechanism, while dynamically adjusting to changes in the nonconformity score distribution, requires careful tuning of the percentile parameters. This tuning process may be sensitive to the specific data characteristics and attack strategies, potentially necessitating further automation for deployment in varied scenarios.

Finally, the computational overhead associated with calculating nonconformity scores across large calibration datasets might pose scalability challenges, particularly in environments with a vast number of clients.

Future work will aim to refine the adaptive thresholding for enhanced robustness, and explore computational optimizations to scale SD-CSFL for real-world federated learning applications.

5.4 Conclusion

In this chapter, we presented the Synthetic Data-Driven Conformity Scoring for Federated Learning (SD-CSFL) framework, a novel approach designed to secure federated learning systems against gradient manipulation and backdoor attacks. By leveraging an independently generated, privacy-preserving synthetic calibration dataset and employing an entropy-based nonconformity scoring mechanism, SD-CSFL effectively differentiates between benign and malicious client updates—even in challenging Non-IID environments.

Our experimental results, evaluated on CIFAR-10 and Birds datasets, demonstrate that SD-CSFL consistently outperforms existing defense mechanisms, maintaining higher model accuracy and significantly reducing attack success rates. The adaptive thresholding and stratified sampling strategies have proven essential for capturing subtle adversarial deviations, while the optional perturbed global model provides an additional layer of security without completely excluding flagged clients.

Overall, SD-CSFL represents a significant step towards robust and secure federated learning, offering a unified defense that effectively mitigates a wide range of adversarial threats while preserving data privacy.

Chapter 6

Conclusions and Future Work

6.1 Summary of Contributions

This thesis makes a significant contribution to enhancing the security and robustness of federated learning, particularly in the face of adversarial threats such as data poisoning, model poisoning, and backdoor attacks. The decentralized nature of federated learning introduces unique vulnerabilities, especially in safety-critical applications.

To address these challenges, we proposed novel defense mechanisms that integrate clustering-based aggregation, knowledge distillation, and synthetic data-driven anomaly detection. These approaches strengthen the resilience and trustworthiness of FL systems under diverse and adaptive adversarial conditions.

6.1.1 (Chapter 3) Defending Against Data and Model Poisoning Attacks

In this chapter, we presented **Robust Federated Clustering (RFCL)** [2], a novel aggregation framework designed to enhance the resilience of federated learning (FL) against data poisoning and model poisoning attacks. Traditional aggregation methods, such as FedAvg, Median, and Krum, assume that most client updates are benign, making them vulnerable to adversarial manipulations. RFCL addresses this limitation by dynamically detecting and filtering out malicious client updates through a combination of clustering and similarity-based analysis.

The proposed RFCL framework introduced a multi-center clustering strategy that partitions client updates into groups based on their similarity, enabling the system to identify and isolate adversarial contributions. This technique was further enhanced by the integration of HDBSCAN-based anomaly detection, which effectively identified outlier updates that deviated significantly from the benign client distribution. Additionally, RFCL employed cosine similarity filtering to refine aggregation, ensuring that only updates with high similarity to benign clusters were incorporated into the global model.

Experimental evaluations demonstrated that RFCL significantly improved FL security by mitigating the impact of poisoning attacks. The results indicated a 40% reduction in adversarial influence while maintaining high model performance in Non-IID settings. These findings highlight the effectiveness of RFCL in preventing malicious gradient manipulations and ensuring the integrity of the global model.

6.1.2 (Chapter 4) Counteracting Backdoor Attacks

This chapter addressed the challenge of backdoor attacks in FL, where adversarial clients inject malicious triggers into the global model while preserving accuracy on clean data. To counteract this threat, we introduced **Robust Knowledge Distillation (RKD)** [3], a novel framework that systematically detects and filters out backdoor-infected models before aggregation.

RKD employed an automated clustering-based filtering approach to separate backdoor-infected updates from benign contributions. By leveraging gradient similarity analysis, RKD efficiently isolated malicious models without requiring direct access to client data. Furthermore, RKD incorporated knowledge distillation techniques to ensure that only benign knowledge is retained in the global model, effectively neutralizing backdoor triggers without significantly affecting model performance.

To enhance the adaptability of the defense mechanism, RKD introduced an adaptive thresholding mechanism that dynamically adjusted its sensitivity based on observed adversarial behavior. This ensures robustness against evolving backdoor attack strategies, making RKD suitable for real-world FL applications. Experimental evaluations demonstrated that RKD achieved an 85% reduction in backdoor attack success rates, outperforming existing defenses while maintaining high classification accuracy for benign clients.

6.1.3 (Chapter 5) A Comprehensive Defending Framework

In this chapter, we presented **Synthetic Data-Driven Conformity Scoring for FL (SD-CSFL)**, a novel privacy-preserving anomaly detection framework that evaluates model integrity without requiring access to real client data. Unlike traditional anomaly detection methods that rely on inspecting raw updates, SD-CSFL introduced an innovative entropy-based nonconformity scoring method that detects adversarial deviations based on model behavior.

The SD-CSFL framework leveraged synthetic calibration datasets to assess the integrity of incoming model updates, thereby eliminating privacy concerns associated with direct data access. By computing entropy-based nonconformity scores, SD-CSFL effectively identified adversarial manipulations and prevented compromised updates from influencing the global model. Additionally, the framework incorporated a threshold dynamics mechanism that adaptively refined anomaly detection sensitivity across training rounds, ensuring sustained robustness over time.

Experimental results validated the effectiveness of SD-CSFL, demonstrating its ability to achieve an 80% detection accuracy while preserving the overall performance of the global model. These findings confirm that SD-CSFL provides a scalable and privacy-preserving approach to securing FL against adversarial attacks, making it a viable solution for applications in sensitive domains such as healthcare and finance.

6.2 Open Challenges and Future Research Directions

Despite the significant progress achieved through RFCL, RKD, and SD-CSFL, several open challenges remain, offering promising avenues for future exploration. While these defense mechanisms provide effective security against diverse adversarial threats in FL, further enhancements are necessary to improve computational efficiency and facilitate real-world deployment. This section outlines key research directions aimed at strengthening FL security.

6.2.1 Enhancing the Computational Efficiency of FL Defenses

Although the proposed defenses demonstrate strong robustness, their computational overhead remains a challenge, particularly for large-scale FL deployments. Clustering-based filtering, knowledge distillation, and entropy-based anomaly detection introduce additional processing requirements, which may hinder their practical deployment in resource-constrained environments such as edge computing and IoT networks. Future research should explore optimization strategies, including lightweight clustering techniques, hardware-accelerated security mechanisms leveraging GPUs, and federated pruning techniques to maintain efficiency while preserving model integrity. Addressing these computational constraints will be crucial for making FL security mechanisms scalable and practical.

6.2.2 Addressing Privacy and Security Trade-offs in FL

Ensuring strong security while preserving user privacy is a fundamental challenge in FL. While SD-CSFL successfully introduces synthetic data-driven anomaly detection, further research is needed to explore more advanced privacy-preserving security techniques. Future work should investigate integrating differentially private anomaly detection, secure multi-party computation (MPC)-based security mechanisms, and homomorphic encryption to ensure robust adversarial defense without compromising data confidentiality. Developing efficient privacy-preserving security mechanisms will be essential for compliance with regulations while maintaining the robustness of FL models.

6.2.3 Security Challenges in Cross-Silo Federated Learning

Cross-silo FL presents unique challenges compared to cross-device FL, particularly in scenarios where multiple organizations or institutions collaborate while maintaining strict data sovereignty requirements. Unlike cross-device FL, where clients are often mobile or edge devices with intermittent availability, cross-silo FL involves a smaller number of participants, such as hospitals, banks, or research institutions, each with significantly larger datasets and more powerful computational resources.

Despite these advantages, cross-silo FL faces security challenges related to model poisoning and adversarial collusion. Unlike cross-device FL, where malicious clients can be statistically filtered due to their abundance, adversarial clients in cross-silo FL can have a much larger impact due to the limited number of participants. Additionally, organizations may have varying degrees of trust, making it difficult to assume full cooperation. Future research should focus on developing robust trust mechanisms for cross-silo FL, such as blockchain-based trust verification, hierarchical anomaly detection models, and institution-specific security constraints to ensure adversarial robustness while maintaining institutional autonomy.

6.2.4 Overcoming Real-World Deployment Challenges

Although the proposed defense mechanisms have been extensively validated in controlled experimental settings, real-world FL deployments introduce additional complexities. In practical scenarios, FL systems must handle dynamic client participation, where clients may frequently join or leave the training process. Ensuring model robustness under such non-static conditions remains an open challenge.

Additionally, FL architectures deployed in highly decentralized environments, such as edge computing and cross-device FL, require further exploration of security mechanisms that can adapt to variable network conditions and heterogeneous computational resources. Future research should focus on designing FL security frameworks that maintain effectiveness despite client participation variability and unreliable communication infrastructures.

6.3 Final Remarks

This thesis introduces novel defense mechanisms that significantly enhance the security and robustness of Federated Learning. Through the development of RFCL, RKD, and SD-CSFL, we establish a comprehensive framework that mitigates data poisoning, model poisoning, and backdoor threats while preserving model performance and privacy.

While these contributions represent a major step forward in FL security, further research is needed to improve computational efficiency, develop adaptive defense mechanisms, and facilitate large-scale deployment in real-world scenarios. The methodologies proposed in this thesis lay the groundwork for the continued advancement of secure and privacy-preserving federated learning systems.

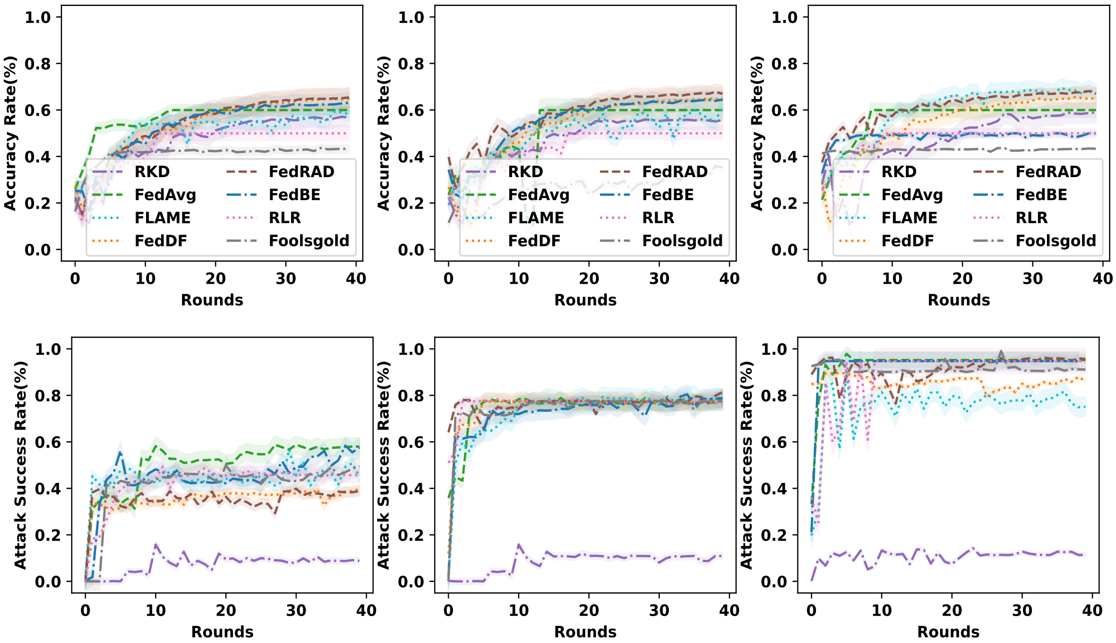
Appendix A

Experiments of RKD Under IID Conditions

This appendix presents supplementary experimental results evaluating the proposed RKD framework and baseline methods under IID data distributions on the CIFAR-10 dataset. We also compare performance under both IID and Non-IID settings when no attacks are present.

Defense Against A3FL Attack Under IID Conditions. Figure A.1 illustrates the performance of RKD and baseline models against the A3FL attack, considering different attacker ratios (20%, 40%, and 60%) in an IID setting. Despite the higher data homogeneity, the results show that RKD maintains a comparatively lower Attack Success Rate (ASR) and a higher Main Task Accuracy (MTA) than the baselines, reinforcing RKD's capability to defend FL in diverse attack scenarios. The baselines demonstrate modest robustness at smaller adversarial ratios, possibly due to the uniform data distribution enabling more generalizable model learning.

Defense Against F3BA Attack Under IID Conditions. Figure A.2 shows the results of RKD's defense against the F3BA attack under an IID data distribution. Across varying attacker ratios (20%, 40%, and 60%), RKD consistently achieves higher accuracy and lower ASR relative to competing defences. The baseline methods also retain some robustness—particularly at lower adversarial



(a) 20% of A3FL

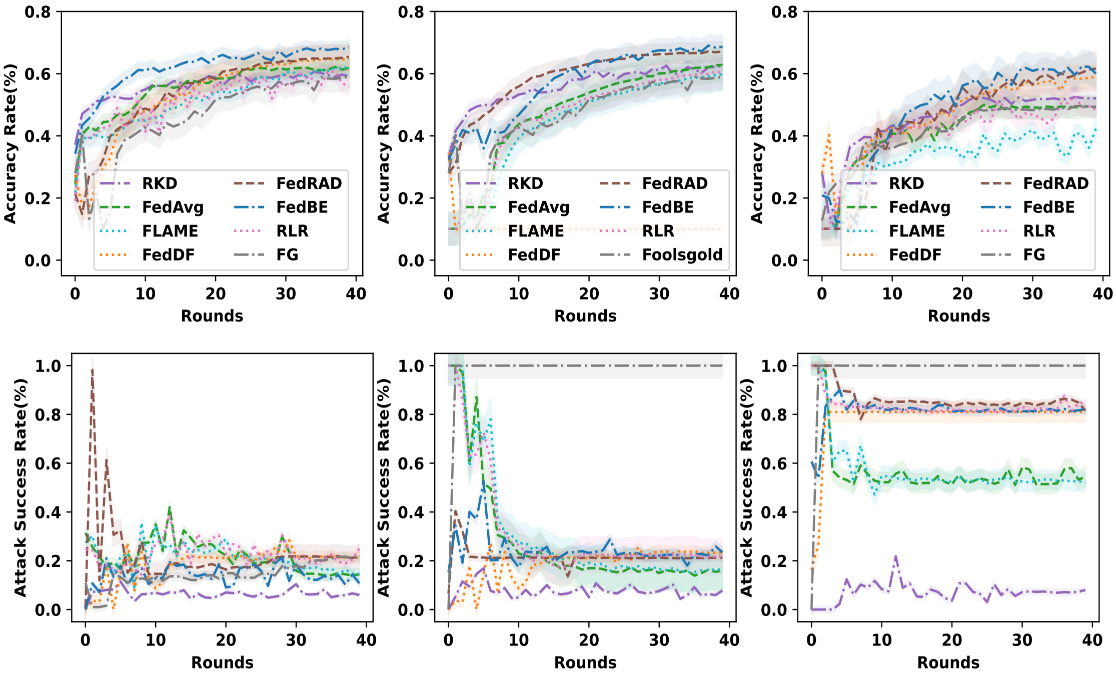
(b) 40% of A3FL

(c) 60% of A3FL

Figure A.1: Performance of baselines and RKD on CIFAR-10 under IID settings against A3FL attackers.

ratios—potentially due to the uniform class distribution promoting more stable feature extraction and model convergence.

Performance Under Non-IID and IID Conditions Without Attacks. Finally, in the absence of any attack, Figure A.3 compares the performance of RKD and baseline models on CIFAR-10 under both Non-IID and IID data distributions. As expected, all methods achieve near-zero ASR, given there is no adversarial interference. Models trained on IID data generally attain higher accuracy due to the uniform class distribution. Meanwhile, Non-IID settings introduce additional complexity that marginally lowers accuracy, indicating that attackers may exploit heterogeneous data distributions more easily than IID ones.

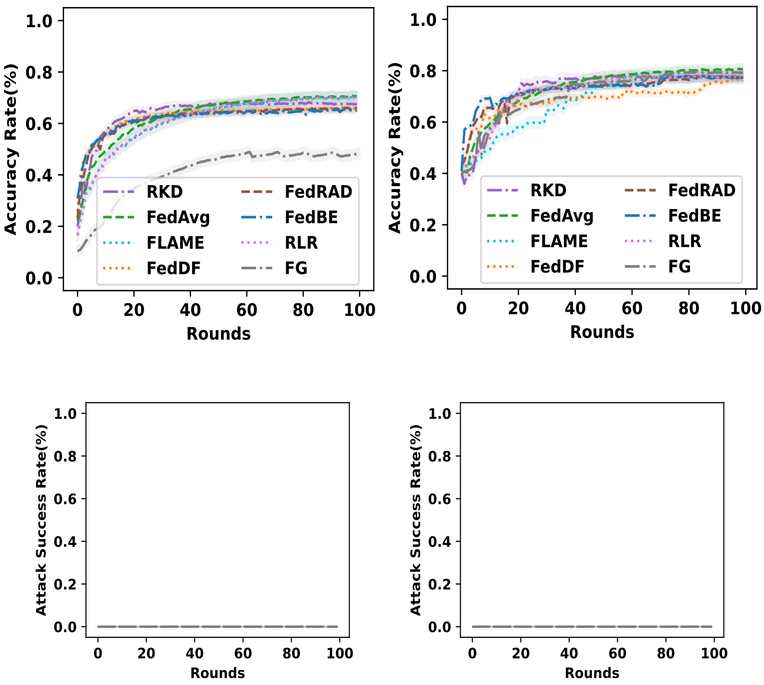


(a) 20% of F3BA

(b) 40% of F3BA

(c) 60% of F3BA

Figure A.2: Performance of baselines and RKD on CIFAR-10 under IID settings against F3BA attackers.



(a) Non-IID

(b) IID

Figure A.3: Performance of baselines and RKD on CIFAR-10 with no attack, comparing Non-IID and IID data distributions.

References

[1] Alharbi, Ebtisaam and Kerim, Abdulrahman and Marcolino, Leandro Soriano and Ni, Qiang. “Synthetic Data-Driven Federated Learning Defense Against Gradient Manipulation and Backdoor Attacks”. In: *ICLR 2026*. Under-review. 2026.

[2] Alharbi, Ebtisaam and Marcolino, Leandro Soriano and Gouglidis, Antonios and Ni, Qiang. “Robust Federated Learning Method Against Data and Model Poisoning Attacks with Heterogeneous Data Distribution”. In: *ECAI 2023*. IOS Press, 2023, pp. 85–92.

[3] Alharbi, Ebtisaam and Marcolino, Leandro Soriano and Gouglidis, Antonios and Ni, Qiang. “Robust Knowledge Distillation in Federated Learning: Counteracting Backdoor Attacks”. In: *SaTML 2025*. IEEE, 2025.

[4] Anastasios N Angelopoulos and Stephen Bates. “A gentle introduction to conformal prediction and distribution-free uncertainty quantification”. In: *arXiv preprint arXiv:2107.07511* (2021).

[5] Bagdasaryan, Eugene and Veit, Andreas and Hua, Yiqing and Estrin, Deborah and Shmatikov, Vitaly. “How to backdoor federated learning”. In: *International conference on artificial intelligence and statistics*. PMLR. 2020, pp. 2938–2948.

[6] Bao, Wenxuan and Wu, Jun and He, Jingrui. “BOBA: Byzantine-Robust Federated Learning with Label Skewness”. In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2024, pp. 892–900.

[7] Baruch, Gilad and Baruch, Moran and Goldberg, Yoav. “A little is enough: Circumventing defenses for distributed learning”. In: *Advances in Neural Information Processing Systems* 32 (2019).

[8] Beltrán, Enrique Tomás Martínez and Pérez, Mario Quiles and Sánchez, Pedro Miguel Sánchez and Bernal, Sergio López and Bovet, Gérôme and Pérez, Manuel Gil and Pérez, Gregorio Martínez and Celdrán, Alberto Huertas. “Decentralized federated learning: Fundamentals, state of the art, frameworks, trends, and challenges”. In: *IEEE Communications Surveys & Tutorials* (2023).

[9] Bhagoji, Arjun Nitin and Chakraborty, Supriyo and Mittal, Prateek and Calo, Seraphin. “Analyzing federated learning through an adversarial lens”. In: *International conference on machine learning*. PMLR. 2019, pp. 634–643.

[10] Bishop, Christopher M and Nasrabadi, Nasser M. *Pattern recognition and machine learning*. Vol. 4. 4. Springer, 2006.

[11] Blanchard, Peva and El Mhamdi, El Mahdi and Guerraoui, Rachid and Stainer, Julien. “Machine learning with adversaries: Byzantine tolerant gradient descent”. In: *Advances in neural information processing systems* 30 (2017).

[12] Campello, Ricardo JGB and Moulavi, Davoud and Zimek, Arthur and Sander, Jörg. “Hierarchical density estimates for data clustering, visualization, and outlier detection”. In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 10.1 (2015), pp. 1–51.

[13] Cao, Xiaoyu and Fang, Minghong and Liu, Jia and Gong, Neil Zhenqiang. “Fltrust: Byzantine-robust federated learning via trust bootstrapping”. In: *arXiv preprint arXiv:2012.13995* (2020).

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

[15] Chen, Hong-You and Chao, Wei-Lun. “Fedbe: Making bayesian model ensemble applicable to federated learning”. In: *arXiv preprint arXiv:2009.01974* (2020).

[16] Cohen, Gregory and Afshar, Saeed and Tapson, Jonathan and Van Schaik, Andre. “EMNIST: Extending MNIST to handwritten letters”. In: *2017 international joint conference on neural networks (IJCNN)*. IEEE. 2017, pp. 2921–2926.

[17] Di, Yicheng and Shi, Hongjian and Ma, Ruhui and Gao, Honghao and Liu, Yuan and Wang, Weiyu. “FedRL: a reinforcement learning federated recommender system for efficient communication using reinforcement selector and hypernet generator”. In: *ACM Transactions on Recommender Systems* (2024).

[18] Fang, Pei and Chen, Jinghui. “On the vulnerability of backdoor defenses for federated learning”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 37. 10. 2023, pp. 11800–11808.

[19] Fung, Clement and Yoon, Chris JM and Beschastnikh, Ivan. “Mitigating sybils in federated learning poisoning”. In: *arXiv preprint arXiv:1808.04866* (2018).

[20] Fung, Clement and Yoon, Chris JM and Beschastnikh, Ivan. “The limitations of federated learning in sybil settings”. In: *23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020)*. 2020, pp. 301–316.

[21] booktitle=Proceedings of the 29th ACM International Conference on Multimedia Ge, Yunjie and Wang, Qian and Zheng, Baolin and Zhuang, Xinlu and Li, Qi and Shen, Chao and Wang, Cong. “Anti-distillation backdoor attacks: Backdoors can really survive in knowledge distillation”. In: 2021, pp. 826–834.

[22] Guerraoui, Rachid and Rouault, Sébastien and others. “The hidden vulnerability of distributed learning in byzantium”. In: *International Conference on Machine Learning*. PMLR. 2018, pp. 3521–3530.

[23] Guo, Yifan and Wang, Qianlong and Ji, Tianxi and Wang, Xufei and Li, Pan. "Resisting distributed backdoor attacks in federated learning: A dynamic norm clipping approach". In: *2021 IEEE International Conference on Big Data (Big Data)*. IEEE. 2021, pp. 1172–1182.

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

[25] Hu, Zeou and Shaloudegi, Kiarash and Zhang, Guojun and Yu, Yaoliang. "Federated learning meets multi-objective optimization". In: *IEEE Transactions on Network Science and Engineering* 9.4 (2022), pp. 2039–2051.

[26] Ian T Jolliffe and Jorge Cadima. "Principal component analysis: a review and recent developments". In: *Philosophical transactions of the royal society A: Mathematical, Physical and Engineering Sciences* 374.2065 (2016), p. 20150202.

[27] Kairouz, Peter and McMahan, H Brendan and Avent, Brendan and Bellet, Aurélien and Bennis, Mehdi and Bhagoji, Arjun Nitin and Bonawitz, Kallista and Charles, Zachary and Cormode, Graham and Cummings, Rachel and others. "Advances and open problems in federated learning". In: *Foundations and trends® in machine learning* 14.1–2 (2021), pp. 1–210.

[28] Karimireddy, Sai Praneeth and He, Lie and Jaggi, Martin. "Learning from history for byzantine robust optimization". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 5311–5319.

[29] Karimireddy, Sai Praneeth and Kale, Satyen and Mohri, Mehryar and Reddi, Sashank and Stich, Sebastian and Suresh, Ananda Theertha. "Scaffold: Stochastic controlled averaging for federated learning". In: *International conference on machine learning*. PMLR. 2020, pp. 5132–5143.

[30] Kerim, Abdulrahman and Marcolino, Leandro Soriano and Nascimento, Erickson R and Jiang, Richard. "Multi-Armed Bandit Approach for Optimizing Training on Synthetic Data". In: *arXiv preprint arXiv:2412.05466* (2024).

[31] Yash Kotha. *Birds-525 Species Image Classification.* https://huggingface.co/datasets/yashikota/birds-525-species-image-classification. Accessed: 2025-07-15. 2023.

[32] Krizhevsky, Alex and Nair, Vinod and Hinton, Geoffrey and others. "The CIFAR-10 dataset". In: *online: http://www. cs. toronto. edu/kriz/cifar. html* 55.5 (2014), p. 2.

[33] LeCun, Yann and Bottou, Léon and Bengio, Yoshua and Haffner, Patrick. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[34] Li, Tian and Sahu, Anit Kumar and Talwalkar, Ameet and Smith, Virginia. "Federated learning: Challenges, methods, and future directions". In: *IEEE signal processing magazine* 37.3 (2020), pp. 50–60.

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

[36] Lim, Wei Yang Bryan and Luong, Nguyen Cong and Hoang, Dinh Thai and Jiao, Yutao and Liang, Ying-Chang and Yang, Qiang and Niyato, Dusit and Miao, Chunyan. "Federated learning in mobile edge networks: A comprehensive survey". In: *IEEE communications surveys & tutorials* 22.3 (2020), pp. 2031–2063.

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

[38] Liu, Yang and Kang, Yan and Xing, Chaoping and Chen, Tianjian and Yang, Qiang. “A secure federated transfer learning framework”. In: *IEEE Intelligent Systems* 35.4 (2020), pp. 70–82.

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

[40] Long, Guodong and Tan, Yue and Jiang, Jing and Zhang, Chengqi. “Federated learning for open banking”. In: *Federated learning: privacy and incentive.* Springer, 2020, pp. 240–254.

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

[42] Lyu, Lingjuan and Yu, Han and Ma, Xingjun and Chen, Chen and Sun, Lichao and Zhao, Jun and Yang, Qiang and Philip, S Yu. “Privacy and robustness in federated learning: Attacks and defenses”. In: *IEEE transactions on neural networks and learning systems* (2022).

[43] Lyu, Xiaoting and Han, Yufei and Wang, Wei and Liu, Jingkai and Wang, Bin and Liu, Jiqiang and Zhang, Xiangliang. “Poisoning with cerberus: Stealthy and colluded backdoor attack against federated learning”. In: *Proceedings of the AAAI Conference on Artificial Intelligence.* Vol. 37. 7. 2023, pp. 9020–9028.

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

[45] McInnes, Leland and Healy, John and Astels, Steve and others. “hdbscan: Hierarchical density based clustering.” In: *J. Open Source Softw.* 2.11 (2017), p. 205.

[46] McMahan, Brendan and Ramage, Daniel. “Federated learning: Collaborative machine learning without centralized training data”. In: *Google Research Blog* 3 (2017).

[47] Mothukuri, Viraaji and Parizi, Reza M and Pouriyeh, Seyedamin and Huang, Yan and Dehghantanha, Ali and Srivastava, Gautam. “A survey on security and privacy of federated learning”. In: *Future Generation Computer Systems* 115 (2021), pp. 619–640.

[48] Muñoz-González, Luis and Co, Kenneth T and Lupu, Emil C. “Byzantine-robust federated machine learning through adaptive model averaging”. In: *arXiv preprint arXiv:1909.05125* (2019).

[49] Neto, Helio N Cunha and Hribar, Jernej and Dusparic, Ivana and Mattos, Diogo Menezes Ferrazani and Fernandes, Natalia C. “A survey on securing federated learning: Analysis of applications, attacks, challenges, and trends”. In: *IEEE Access* 11 (2023), pp. 41928–41953.

[50] Nguyen, Thien Duc and Rieger, Phillip and De Viti, Roberta and Chen, Huili and Brandenburg, Björn B and Yalame, Hossein and Möllering, Helen and Fereidooni, Hossein and Marchal, Samuel and Miettinen, Markus and others. “{FLAME}: Taming backdoors in federated learning”. In: *31st USENIX Security Symposium (USENIX Security 22)*. 2022, pp. 1415–1432.

[51] OpenAI. *ChatGPT*. https://chat.openai.com/. Online; accessed: 2024-07-20. 2024.

[52] Ozdayi, Mustafa Safa and Kantarcioglu, Murat and Gel, Yulia R. “Defending against backdoors in federated learning with robust learning rate”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 10. 2021, pp. 9268–9276.

[53] Park, Jungwuk and Han, Dong-Jun and Choi, Minseok and Moon, Jaekyun. “Sageflow: Robust federated learning against both stragglers and adversaries”. In: *Advances in neural information processing systems* 34 (2021), pp. 840–851.

[54] Pillutla, Krishna and Kakade, Sham M and Harchaoui, Zaid. "Robust aggregation for federated learning". In: *IEEE Transactions on Signal Processing* 70 (2022), pp. 1142–1154.

[55] Quinn, Joanne and McEachen, Joanne and Fullan, Michael and Gardner, Mag and Drummy, Max. *Dive into deep learning: Tools for engagement.* Corwin Press, 2019.

[56] Rodríguez-Barroso, Nuria and Jiménez-López, Daniel and Luzón, M Victoria and Herrera, Francisco and Martínez-Cámara, Eugenio. "Survey on federated learning threats: Concepts, taxonomy on attacks and defences, experimental study and challenges". In: *Information Fusion* 90 (2023), pp. 148–173.

[57] Rombach, Robin and Blattmann, Andreas and Lorenz, Dominik and Esser, Patrick and Ommer, Björn. "High-resolution image synthesis with latent diffusion models". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.* 2022, pp. 10684–10695.

[58] Sankupellay, Mangalam and Konovalov, Dmitry. "Bird call recognition using deep convolutional neural network, ResNet-50". In: *Proc. Acoustics.* Vol. 7. 2018. 2018, pp. 1–8.

[59] Shejwalkar, Virat and Houmansadr, Amir. "Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning". In: *NDSS.* 2021.

[60] Sturluson, Stefán Páll and Trew, Samuel and Muñoz-González, Luis and Grama, Matei and Passerat-Palmbach, Jonathan and Rueckert, Daniel and Alansary, Amir. "Fedrad: Federated robust adaptive distillation". In: *arXiv preprint arXiv:2112.01405* (2021).

[61] Tan, Alysa Ziying and Yu, Han and Cui, Lizhen and Yang, Qiang. "Towards personalized federated learning". In: *IEEE transactions on neural networks and learning systems* 34.12 (2022), pp. 9587–9603.

[62] Tolpegin, Vale and Truex, Stacey and Gursoy, Mehmet Emre and Liu, Ling. "Data poisoning attacks against federated learning systems". In: *European Symposium on Research in Computer Security*. Springer. 2020, pp. 480–501.

[63] Wang, Hongyi and Sreenivasan, Kartik and Rajput, Shashank and Vishwakarma, Harit and Agarwal, Saurabh and Sohn, Jy-yong and Lee, Kangwook and Papailiopoulos, Dimitris. "Attack of the tails: Yes, you really can backdoor federated learning". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 16070–16084.

[64] Wang, Yongkang and Zhai, Di-Hua and Han, Dongyu and Guan, Yuyin and Xia, Yuanqing. "MITDBA: Mitigating Dynamic Backdoor Attacks in Federated Learning for IoT Applications". In: *IEEE Internet of Things Journal* (2023).

[65] Wu, Chen and Yang, Xian and Zhu, Sencun and Mitra, Prasenjit. "Toward cleansing backdoored neural networks in federated learning". In: *2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS)*. IEEE. 2022, pp. 820–830.

[66] Xia, Qi and Tao, Zeyi and Hao, Zijiang and Li, Qun. "FABA: an algorithm for fast aggregation against byzantine attacks in distributed neural networks". In: *IJCAI*. 2019.

[67] Xiao, H. "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms". In: *arXiv preprint arXiv:1708.07747* (2017).

[68] Xie, Chulin and Huang, Keli and Chen, Pin Yu and Li, Bo. "Dba: Distributed backdoor attacks against federated learning". In: *8th International Conference on Learning Representations, ICLR 2020*. 2020.

[69] Xie, Cong and Koyejo, Oluwasanmi and Gupta, Indranil. "Fall of empires: Breaking byzantine-tolerant sgd by inner product manipulation". In: *Uncertainty in Artificial Intelligence*. PMLR. 2020, pp. 261–270.

[70] Jie Xu et al. “Federated learning for healthcare informatics”. In: *Journal of healthcare informatics research* 5 (2021), pp. 1–19.

[71] Yang, Lei and Huang, Jiaming and Lin, Wanyu and Cao, Jiannong. “Personalized federated learning on non-IID data via group-based meta-learning”. In: *ACM Transactions on Knowledge Discovery from Data* 17.4 (2023), pp. 1–20.

[72] Yang, Liu and Tan, Ben and Zheng, Vincent W and Chen, Kai and Yang, Qiang. “Federated recommendation systems”. In: *Federated Learning: Privacy and Incentive* (2020), pp. 225–239.

[73] Yang, Qiang and Liu, Yang and Chen, Tianjian and Tong, Yongxin. “Federated machine learning: Concept and applications”. In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 10.2 (2019), pp. 1–19.

[74] Yin, Dong and Chen, Yudong and Kannan, Ramchandran and Bartlett, Peter. “Byzantine-robust distributed learning: Towards optimal statistical rates”. In: *International Conference on Machine Learning*. PMLR. 2018, pp. 5650–5659.

[75] Yurochkin, Mikhail and Agarwal, Mayank and Ghosh, Soumya and Greenewald, Kristjan and Hoang, Nghia and Khazaeni, Yasaman. “Bayesian nonparametric federated learning of neural networks”. In: *International conference on machine learning*. PMLR. 2019, pp. 7252–7261.

[76] Zhang, Hangfan and Jia, Jinyuan and Chen, Jinghui and Lin, Lu and Wu, Dinghao. “A3fl: Adversarially adaptive backdoor attacks to federated learning”. In: *Advances in Neural Information Processing Systems* 36 (2024).

[77] Zhang, Zhiyuan and Su, Qi and Sun, Xu. “Dim-Krum: Backdoor-Resistant Federated Learning for NLP with Dimension-wise Krum-Based Aggregation”. In: *Findings of the Association for Computational Linguistics: EMNLP 2022*. 2022, pp. 339–354.