People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
SAAD DAHLAB UNIVERSITY OF BLIDA 1
Faculty of Sciences
Department of Computer Science

Master's dissertation

Speciality : Computer science Security

Presented by :

Benkessiouer Hind

Hakem Yassine

THEME

Poisoning attacks detection in federated learning for healthcare applications

Publicly defended on **02/07/2025**, before a jury composed of :

Dr.	Boudraa Sawsen	USDB	President
Prof.	Berramdane	USDB	Thesis Examiner
Dr.	Madani Amina	USDB	Thesis Supervisor

Acknowledgements

We would like to express our sincerest gratitude to Dr. Madani Amina, our thesis supervisor, for her continuous support, encouragement, and motivation throughout this project. Her instructions and advice were most beneficial in conducting our research.

Also, we would like to extend our sincerest gratitude to Mme Ouaili, a committed researcher, for her technical support and constructive remarks, which played a crucial role in the elaboration of our suggested solution and the attainment of the results of this dissertation.

Additionally, we would like to thank the teaching and administrative personnel of the Computer Science Department of Saad Dahleb Blida University for providing the favorable learning environment and facilities required for the fulfillment of this work.

Without the care and guidance of our families, the completion of this thesis would have been impossible. We would like to thank our parents for their constant love and faith in our abilities that have given us the courage to overcome any obstacle.We are grateful for the unwavering encouragement, forbearance, and support that our siblings have accorded to us.

We would like to express our deepest gratitude to our fellow colleagues and the members of the 5Nome group for five years that were nothing less than extraordinary. During this time, we faced challenges, marked triumphs, pulled all-nighters, and pushed through times of doubt; however, our greatest achievement was that we became one. The commitment, friendship, and unshakeable determination displayed by these people not only characterized this learning experience but also left deep impressions on our lives. The learning experience would have been vastly different had it not been for their contributions. Moreover, we acknowledge with thanks the open-source and academic communities for their selfless contributions, which made the application of this research possible.

Federation toolkits with PyTorch and PathMNIST support easy experimentation and implementation. In short, it is important to acknowledge the broader research community that works to provide the privacy, security, and ethical conduct of machine learning. We build upon their efforts in establishing our thesis and seek to make our own contribution to the collective success in guaranteeing the trustworthy and viable application of federated learning in actual healthcare environments.

Abstract

Federated Learning (FL) in healthcare offers a clear approach to collaboratively train machine learning algorithms, all while ensuring patient confidentiality and adhering to regulatory standards. Nonetheless, this decentralized framework introduces fresh security challenges, especially in the form of poisoning attacks. In these scenarios, malicious clients intentionally modify data or alter model updates, aiming to undermine overall performance or introduce harmful activities.

This thesis explores the challenges posed by poisoning attacks in healthcare-related federated learning systems and introduces a new hybrid adaptive defense system that dynamically balances effectiveness and resilience in response to immediate threat assessments. Our approach combines various aggregation methods with anomaly detection to effectively address different attack scenarios while ensuring that system performance remains strong during regular operations.

We conducted thorough experiments on the PathMNIST dataset, simulating different adversarial poisoning attacks that included label flipping, backdoor methods, and composite strategies. The findings indicate that the proposed enhancements significantly boost defense efficiency, particularly regarding accuracy, F1-score, and false negative rate, while maintaining a reasonable level of computational overhead. The hybrid system is designed to handle challenges gracefully and adjusts well to unexpected changes, which makes it a good fit for important healthcare settings. This work highlights the importance of flexible security solutions in federated environments and lays a strong foundation for future advancements in trusted collaborative learning, especially in sensitive areas.

Keywords: Federated Learning; Healthcare; Machine Learning; Poisoning Attacks; Hybrid Adaptive Defense; Aggregation Methods; Anomaly Detection; PathMNIST.

Résumé

L'apprentissage fédéré (FL) dans le secteur de la santé offre une approche claire pour former collaborativement des algorithmes d'apprentissage automatique, tout en garantissant la confidentialité des patients et en respectant les normes réglementaires. Néanmoins, ce cadre décentralisé introduit de nouveaux défis de sécurité, notamment sous la forme d'attaques de poisoning. Dans ces scénarios, des clients malveillants modifient intentionnellement les données ou altèrent les mises à jour du modèle, dans le but de saper la performance globale ou d'introduire des activités nuisibles.

Cette thèse explore les défis posés par les attaques par empoisonnement dans les systèmes d'apprentissage fédéré liés à la santé et introduit un nouveau système de défense hybride adaptatif qui équilibre dynamiquement efficacité et résilience en réponse aux évaluations immédiates des menaces. Notre approche combine diverses méthodes d'agrégation avec la détection d'anomalies pour traiter efficacement différents scénarios d'attaque tout en garantissant que la performance du système reste solide lors des opérations régulières.

Nous avons mené des expériences approfondies sur le jeu de données PathMNIST, en simulant différentes attaques de poisoning adversarial qui comprenaient le retournement d'étiquettes, les méthodes de porte dérobée et des stratégies composites. Les résultats indiquent que les améliorations proposées augmentent considérablement l'efficacité de la défense, en particulier en ce qui concerne la précision, le score F1 et le taux de faux négatifs, tout en maintenant un niveau raisonnable de surcharge computationnelle. Le système hybride est conçu pour gérer les défis avec aisance et s'adapte bien aux changements inattendus, ce qui en fait un bon choix pour les environnements de soins de santé importants.

Ce travail souligne l'importance des solutions de sécurité flexibles dans les environnements fédérés et jette des bases solides pour les avancées futures dans l'apprentissage collaboratif de confiance, en particulier dans les domaines sensibles.

Mots-clés : apprentissage fédéré ; santé ; algorithmes d'apprentissage automatique ; attaques par empoisonnement ; système de défense hybride adaptatif ; méthodes d'agrégation ; détection d'anomalies ; PathMNIST.

Contents

List of Figures iv

List of Tables v

General Introduction 1

1 Overview of Federated Learning 5

1.1 Introduction . . . 5
1.2 Artificial intelligence . . . 6
1.3 Machine learning . . . 6
1.4 Federated learning . . . 7
1.5 Traditional machine learning Vs Federated learning . . . 8
1.6 Federated learning types . . . 10
1.6.1 Horizantal Federated Learning . . . 10
1.6.2 Vertical Federated Learning . . . 12
1.6.3 Federated Transfer Learning . . . 13
1.7 Federated Aggregation Algorithms . . . 15
1.7.1 Federated Averaging (FedAvg) . . . 15
1.7.2 Federated Proximal (FedProx) . . . 15
1.7.3 Federated Normalized Averaging (FedNova) . . . 16
1.7.4 Krum . . . 16
1.8 FL-based applications . . . 16
1.8.1 Healthcare . . . 17
1.8.2 Internet of Things (IoT) . . . 17
1.8.3 Internet of Underwater Things (IoUT) . . . 17
1.8.4 Industrial engineering . . . 17
1.9 Conclusion . . . 17

2 Poisoning Attack Detection in Federated Learning for Healthcare Applications : State of the Art **19**
2.1 Introduction . . . 19
2.2 Security threats in FL . . . 20
2.3 Poisoning attacks . . . 20
2.3.1 What is a poisoning attack? . . . 20
2.3.2 Types of poisoning attacks . . . 21
2.3.3 The impact of poisoning attacks in FL . . . 23
2.4 Techniques for detecting poisoning attacks . . . 24
2.4.1 Robust aggregation . . . 24
2.4.2 Anomaly detection . . . 25
2.4.3 Client behavior tracking and trust systems . . . 26
2.5 Challenges in detecting poisoning attacks . . . 27
2.5.1 Non-IID data distribution . . . 27
2.5.2 Limited observability at the server level . . . 27
2.5.3 Strategic/adaptive attackers . . . 27
2.5.4 Computational and communication resource constraints . . . 28
2.5.5 Trade-offs between robustness and model performance . . . 28
2.6 Attacks and security in FL applied in healthcare . . . 28
2.6.1 Why security is essential in healthcare FL . . . 28
2.7 Related work . . . 29
2.7.1 Comparative Analysis . . . 32
2.7.2 Discussion and insight . . . 35
2.8 Conclusion . . . 36

3 Our Proposed Approach **37**
3.1 Introduction . . . 37
3.2 Overall architecture . . . 38
3.3 Data preprocessing . . . 39
3.3.1 Healthcare-Specific Data Preprocessing Pipeline . . . 40
3.3.2 Data Cleaning and Validation . . . 41
3.3.3 Data Augmentation for Class Balance . . . 41
3.3.4 Dataset Splitting and Distribution . . . 41
3.4 Model Local training . . . 42
3.4.1 Proposed CNN Architecture . . . 42
3.5 Proposed Defense Architectures for Healthcare Federated Learning : . . . 44
3.5.1 Approach 1 Pure Robust Aggregation : . . . 44

3.5.2 Approach 2 Detection + Standard Aggregation: 51
3.5.3 Approach 3 Hybrid Adaptive Defense: 61
3.6 Conclusion 71

4 Implementation and experimentation 73
4.1 Introduction 73
4.2 Healthcare dataset 73
4.2.1 Dataset overview 73
4.2.2 Federated learning setup 74
4.2.3 Dataset configuration 74
4.3 Model Specifications 75
4.3.1 CNN architecture 76
4.3.2 Model parameters 76
4.3.3 Training configuration 76
4.3.4 Regularization Techniques 77
4.4 Hardware specifications 77
4.5 Software environment 77
4.5.1 Kaggle 78
4.5.2 Python libraries 78
4.6 Evaluation metrics 79
4.6.1 Classification performance metrics 79
4.6.2 Medical safety metrics 80
4.6.3 Defense effectiveness metrics 80
4.6.4 Robustness assessment 80
4.7 Experimental results 81
4.7.1 Defense performance against Byzantine attacks 81
4.7.2 Defense robustness analysis 85
4.7.3 Comparison of defense Trade-offs 85
4.7.4 Statistical significance analysis 86
4.7.5 Comparison with State-of-the-Art 87
4.7.6 Synthesis and key findings 88
4.8 Conclusion 89

General conclusion 90

Bibliography 92

List of Figures

1.1 Illustration of FL architecture and training process [1] 8
1.2 Federated Learning System (Horizontal) [2]. 11
1.3 Architecture for a vertical federated learning system [2]. 12
1.4 Architecture for a transfer federated learning system [2]. 14

2.1 Poisoning attack in a FL environment [3]. 21

3.1 Overall architecture of our solution. 39
3.2 Non-IID Data Distribution Across Federated Clients. 42
3.3 Our proposed CNN architecture for FL in healthcare environments 44
3.4 Pure Robust aggregation for healthcare. 45
3.5 Anomaly Detection+Simple Aggregation for healthcare. 52
3.6 Hybrid adaptive defense for healthcare. 62

4.1 PathMNIST sample images 75
4.2 Convergence comparison of defense approaches under various attacks . . . 82
4.3 Detection Performance Heatmap Comparison 83
4.4 Deffense Effectiveness VS Percentage of malicious clients 84
4.5 Security performance trade-off for defense Approaches 86
4.6 Comparison of Defense Methods on PathMNIST. 87

List of Tables

1.1 Key differences between traditional machine learning and federated learning. 9

2.1 Impact of poisoning attacks across FL stages 24
2.2 Comparative Table of Federated Learning Attacks and Defenses 34

3.1 Advantages and Limitations of Trimmed Mean Aggregation in Healthcare FL 51
3.2 Comparison of Detection-Based vs. Robust Aggregation Approaches 61

4.1 PathMNIST Dataset Split Distribution 74
4.2 Model configuration parameters 76
4.3 Training hyperparameters 76
4.4 Hardware Specifications of Experimental Systems 77
4.5 Overall Performance Comparison of Defense Approaches 81
4.6 F1-Score by pathology class under mixed attacks (Round 15) 82
4.7 Detection performance metrics (Average over 15 rounds) 83
4.8 Computational Overhead Comparison 84
4.9 Defense effectiveness scores 85
4.10 False Negative Rate Comparison (%) 85
4.11 Hybrid Adaptive Aggregation Selection Pattern 86
4.12 Statistical significance of performance differences (p-values) 87

General Introduction

Problem statement

Recent advancements in Artificial Intelligence (AI) and machine learning (ML) have significantly reshaped the healthcare sector, driving progress in clinical decision support, early disease detection, medical imaging analysis, and personalized treatment recommendations. These innovations depend heavily on access to large volumes of high-quality data, typically collected from diverse healthcare institutions, laboratories, and edge devices such as wearable monitors. However, the traditional centralized approach to training machine learning model where raw patient data must be aggregated into a central server raises critical concerns related to data privacy, security, and legal compliance with strict healthcare regulations, including the General Data Protection Regulation (GDPR) and the Health Insurance Portability and Accountability Act (HIPAA). Therefore, Federated Learning (FL) has begun to attract significant attention. Instead of sending data to a central server, FL lets each hospital or device keep its own data and only send model updates to a server. That way, the data remains local, but everyone contributes to improving the shared model. It represents a suitable middle path while preserving privacy and collaboration, that said, FL is not a flawless solution. Since it relies on updates from lots of different clients, it opens the door for attackers who aim to disrupt system integrity. If a client is compromised or intentionally malicious, It may transmit corrupted data or altered model updates, potentially poisoning the training pipeline. This is especially worrying in healthcare, where one wrong prediction could impact someones treatment or diagnosis. These attacks differ in nature and impact. Some are simple, like changing labels in the training data (called data poisoning). Others involve sending updates that are intentionally designed to trick the model (model poisoning), or inserting hidden behaviors that only activate in certain situations (backdoor attacks). And identifying these attacks is highly challenging. The data in healthcare is often unevenly distributed across sources, and different hospitals have different types of patients and records.This is known as Non-Independent and Identically Distributed (non-IID) data, and it makes it harder to know what normal even looks like. On top of that, attackers can act in very sneaky ways that appear identical to standard clients. Some of the current defenses either assume that clients behave similarly, or they need a lot of computing power, which many hospitals do not possess. This raises several important research questions:

- How can we effectively detect data poisoning and model poisoning attacks in a federated learning environment with non-IID data, such as that found in healthcare systems?
- How can we distinguish between malicious client behavior and naturally irregular behavior caused by the heterogeneous nature of medical data?
- And finally, is it possible to design a detection method that is lightweight, privacy-preserving, and suitable for the resource constraints of real-world hospital environments?

These are the core challenges our work aims to address.

Research methodology

So in this thesis, the goal is to establish a more efficient solution to detect poisoning attacks in Federated Learning, with a focus on healthcare systems. The idea is to develop a detection method that is lightweight, respects patient privacy, and remains effective in identifying malicious behavior. To achieve this, we propose three server-side defense strategies that operate during the model aggregation stage of the FL process.

- The initial one is pure robust aggregation, employing statistical techniques such as the trimmed mean to reduce the influence of outlier updates, assuming that most clients are honest.
- The second method combines anomaly detection with traditional aggregation; under this approach, the suspicious client updates are first detected based on behavior analysise.g., the evaluation of deviation from the mean and studying gradient patterns. Then the non-anomalous updates are only aggregated using a simple method like FedAvg.
- The third, more advanced approach forms a hybrid adaptive defense that dynamically switches between detection-based and robust responses, depending on the threat severity as determined during training. This adaptability attempts to find a trade-off between robustness and efficiency, which is based on the attack severity and type.

Our methods are designed to address both targeted and untargeted poisoning attacks, carried out via either data or model manipulation. All defenses are applied at the server side, during the aggregation step of the Federated Learning pipeline, without requiring access to clients raw data. This setup ensures scalability and privacy preservation, which are essential in healthcare environments where computation and communication resources may be limited. These approaches form the foundation of our proposed framework, which is later evaluated under realistic FL conditions with heterogeneous data and various poisoning scenarios.

We employ the PathMNIST dataset for experimental purposes, comprising histopathology images for the multi-class classification of tissue types. The experimental configuration emulates a genuine Federated Learning environment comprising both reliable and hostile clients. Each client trains on a segment of the dataset under non-IID conditions, mirroring the variability characteristic of real-world healthcare systems. A subset of clients is programmed to act deliberately, either by altering labels or manipulating the model, to assess the efficacy of our proposed detection mechanisms. The system utilizes Pytorch framework, and multiple attack scenarios are evaluated across diverse settings. The efficacy of our defensive strategies is assessed through various critical indicators, such as classification accuracy, detection rate of malicious clients, and false positive rate. The results indicate that our methodology effectively mitigates poisoning assaults while sustaining robust model performance and safeguarding data privacy.

Dissertation organization

This thesis is split into four chapters:

- **Chapter 1:** introduces federated learning, Describes how it differs from traditional machine learning, and why it matters for healthcare.
- **Chapter 2:**explains the basics of poisoning attacks.And examines prior work by researchers addressing these attacks different strategies, their limits, and how they perform in healthcare scenarios.
- **Chapter 3:**describes the method we developed, including the rationale behind our selection and its operational mechanism.
- **Chapter 4:** goes into the experiments, the setup, results, and the insights they provide about the performance of the approach.

By addressing a pressing security gap in federated healthcare systems, this research contributes to the development of trustworthy, privacy-preserving AI frameworks capable of safe deployment in real-world medical settings.

Chapter 1

Overview of Federated Learning

1.1 Introduction

Artificial intelligence (AI) has undergone significant advancements in recent decades and is now the primary driving force behind nearly all other technological advancements. Artificial intelligence is dedicated to the creation of systems that are capable of performing tasks that are typically associated with human intelligence, including decision-making, learning, and reasoning. Machine learning (ML) is of paramount significance among the numerous AI methodologies, as it enables computers to enhance their performance by learning from their experiences without being explicitly programmed to do so for a particular task.

Traditional machine learning methods, however, often rely on the concentration of large volumes of data in a single point, which triggers serious concerns about data ownership, security, and privacy. Considering growing sensitivities to these concerns, Federated Learning (FL) has been proposed as a new paradigm. FL allows many devices or organizations to jointly train a common model without requiring the sharing of raw data, thus tapping the power of machine learning while maintaining privacy.

This chapter discusses the fundamental ideas of artificial intelligence and machine learning, then compares traditional techniques to federated techniques. In addition, it introduces the federated learning framework, explaining its benefits, types, main aggregation algorithms, and real-world applications. Based on studying these fundamental areas, we provide a foundation for understanding how federated learning solves the privacy and security issues associated with traditional machine learning.

1.2 Artificial intelligence

The field of artificial intelligence is a subfield of computer science that allows us to build intelligent computers that can think and behave like people and make decisions on their own [4].

The two artificial words that make up AI define something that is manufactured by humans and intelligence that refers to the ability to think for itself, which makes artificial intelligence a "power of reflection created by humans" [4].

This field was founded on the notion that machines would eventually be able to think like humans thanks to their intelligence and conscience. In addition to being a pioneer in the field of research, AI has played a significant role in the development of industries and labor as we know them today [4].

AI systems go through several stages of planning, reasoning, data analysis, result prediction, and action in consequence in order to ultimately create a conscience [5]. Utilizing statistics, probabilities, and various other mathematical tools is another aspect of IA [4].

1.3 Machine learning

It is a branch of artificial intelligence connected with algorithms that learn from data to make predictions independently. These predictions can be made using unsupervised learning, finding general patterns in the data; or supervised learning, learning patterns from existing data. ML models can classify events as true or false, cluster data points based on commonalities in data, and predict numerical values based on historical data. Unlike modern advancements such as FL or Deep Learning, traditional ML usually requires central data collection and processing [6].

Different types of ML algorithms that are [7]:

1. **Supervised learning:** Data are labeled with classes and outcomes (task-driven approach).

2. **Unsupervised learning:** Data are not labeled. The algorithm separates the samples into different classes based only on feature patterns in the training data (data-driven approach).

3. **Semi-Supervised learning:** : It is executed on both the labeled and unlabeled data. It utilizes the labeled portion to guide learning on the unlabeled portion.

4. **Reinforcement learning:** The algorithm operates sequentially, evaluating actions in an environment to maximize long-term reward (environment-driven approach).

1.4 Federated learning

FL was envisioned to support machine learning over decentralized data sources with consideration for minimizing communication costs and protecting data privacy. McMahan et al. [8] in 2017 envisioned mobile devices as the most relevant use case, also pointing to issues with constrained computational capacity, unstable network availability, and user privacy maintenance requirements. In contrast to conventional distributed optimization, which presupposes uniform data distribution across servers, FL functions under the premise that data is disseminated in a non-IID (non-independent and identically distributed) fashion, characterized by uneven distribution among clients.

The framework facilitates data retention at its source while permitting localized training on each device or dataset. A coordinator server oversees the process by sending the model to the nodes participating in the training, who then relay updates to the server for integration. Decentralized architecture reduces the need to communicate raw data, thus maintaining privacy without hindering the training of machine learning models. The initial attributes of Federated Learning (FL) were largely responsible for the development of algorithms with robustness to heterogeneous data distributions, thus having minimal client input at every training round. This paradigm has taken on a central position in advancing communication-effective methods, which ultimately led to the development of the Federated Averaging (FedAvg) algorithm. Being the first method of its type in this field, FedAvg allows the efficient training of models in decentralized settings and solves problems related to heterogeneity of the data distribution along with limited client resources.

The FL architecture is composed of [5]:

- **Clients:**Local devices that hold and process data.
- **Server:** Central entity that coordinates the training process and aggregates model updates.

As presented in Figure 1.1, we can summarize the implementation of FL in the following four steps [5] :

1. **Initialization:**The server initializes a global model and sends it to the clients.
2. **Local Training:** Each client trains the model on its local data and computes updates.
3. **Aggregation:** Clients send the updates back to the server, which aggregates them to update the global model.
4. **Iteration:** Steps 2 and 3 are repeated for several rounds until the model converges.

As shown in Figure 1.1, the FL architecture operates through a sequential process in which a central server distributes a global model, clients perform local training on private data, and the server aggregates updates to refine the model without accessing raw data.

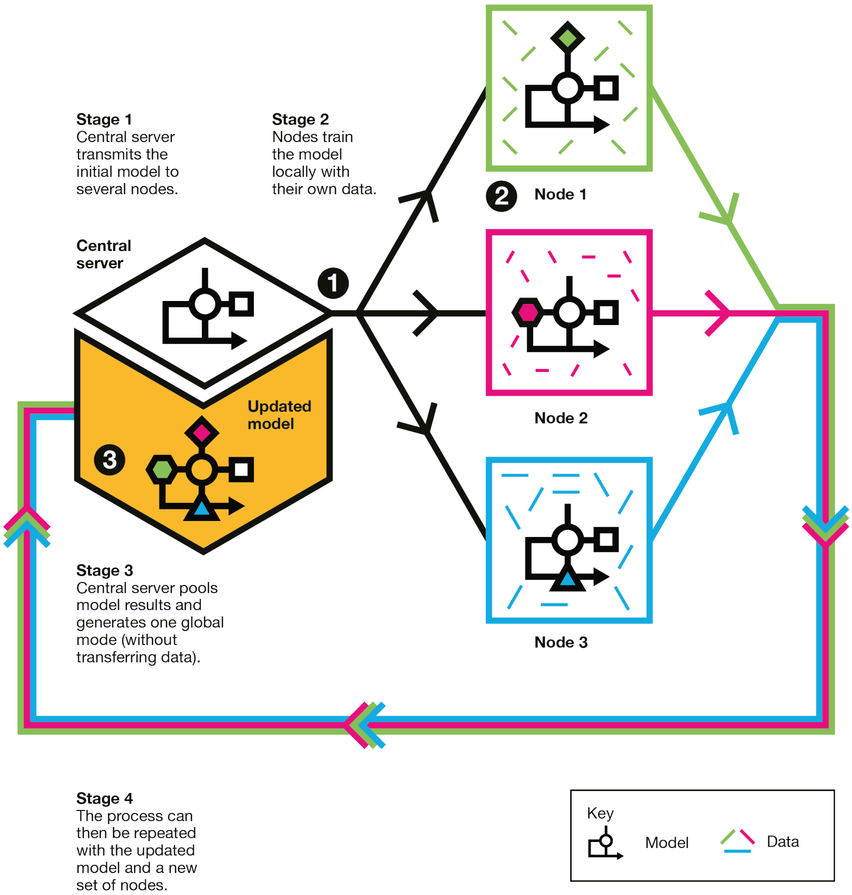

Figure 1.1: Illustration of FL architecture and training process [1]

1.5 Traditional machine learning Vs Federated learning

It is important to understand first the basic principles of FL and classic ML before discussing the difference between the two. Classic ML, which often involves centralized data gathering, computes all training data at a single point. While this method allows for fast computation, it also provides direct access to full datasets. Although this method allows for speedy computation and ready access to entire datasets, it has its drawbacks. It also creates significant privacy, security, and scalability issues especially with sensitive or distributed data. On the other hand, FL provides a distributed method allowing model training straight on user devices or institutional servers without moving raw data. This design promotes cross-institutional collaboration, regulatory compliance, and privacy preservation.

Table 1.1 emphasizes the main distinctions between these two paradigms in terms of data location, privacy, data distribution, communication, computation load, scalability and regulatory compliance. FL provides some unique advantages that address modern

Tableau 1.1: Key differences between traditional machine learning and federated learning.

Aspect	**Traditional Machine Learning**	**Federated Learning**
Data location	Data is stored and processed in a central server, where all data is collected for training.	Only model updates are sent to the central server for aggregation; all other data stays on local devices.
Privacy	Privacy risks are generated by the centralization of sensitive data, which may be disclosed during training.	By keeping data on local devices and only transmitting model updates, privacy is improved and exposure is decreased.
Data distribution	Data is assumed to be uniformly distributed across clients, and can be freely accessed.	Data is frequently non-IID, meaning each client has its own varied and unequally distributed local data.
Communication	Communication occurs infrequently and primarily entails transmitting substantial datasets to the central server.	Communication is more frequent, but limited to exchanging model updates.
Computation Load	Most of the computation is handled by the central server, which usually has ample resources.	Computation is distributed across clients, which may have limited computational power.
Scalability	Scalability is limited by the central servers resources, reducing efficiency with many clients.	FL is naturally scalable due to distributed client processing, allowing it to manage large numbers of participants.
Regulatory Compliance	Central data storage can violate data privacy regulations (e.g., GDPR).	FL aligns with data protection laws since raw data remains with the client.

challenges in ML, particularly those dealing with decentralized and sensitive data. Its merits span a variety of sectors, providing practical solutions for privacy, efficiency, and personalization. Here are a few of the key benefits [9] :

- **Collaborative Learning:** Has the potential to let multiple entities or devices work together in the training of a model by holding the data locally. That means, there is no need to transfer or centrally gather data for handling large volumes in model training.

- **Data Security and Privacy:** Model training can be done without compromising data security and privacy. This is highly important when sensitive data is spread over many devices.
- **Edge Computing:** Because edge computing will enable devices to train models locally, it reduces the requirement for massive volumes of data to traverse over the network. That said, it also allows training models that then can be utilized and deployed on devices with limited resources-for example, mobile phones or Internet of Things sensors.
- **Handling non-IID data:** It is particularly well-suited for training models on non-IID data that is commonly found in real-world scenarios.
- **Scalability:**: It is highly scalable and can handle a large number of devices or entities.

1.6 Federated learning types

FL can be categorized into three main types based on how data is distributed across participating entities in terms of features and user samples. Each type addresses different collaboration scenarios depending on the overlap in data structure and user base. Here are the primary types of FL:

1.6.1 Horizantal Federated Learning

Horizontal Federated Learning (HFL) is used when multiple parties hold datasets with common feature space (i.e., same type of features) but with different user samples. This is often the case in industries such as healthcare or banking, where organizations gather the same kinds of data from different groups. HFL allows organizations to jointly train a common machine learning model while keeping their raw data within their organizational domains to preserve privacy and honor data protection laws. Google's HFL framework for Android devices allows local updates on the client side, which are then securely aggregated in the cloud to produce a global model, thus improving efficiency as well as privacy. Moreover, secure aggregation and compression methods further improve its security and scalability [10]. Figure 1.2, illustrates an HFL system where clients that own heterogeneous datasets collaborate to train a common global model via a central server.

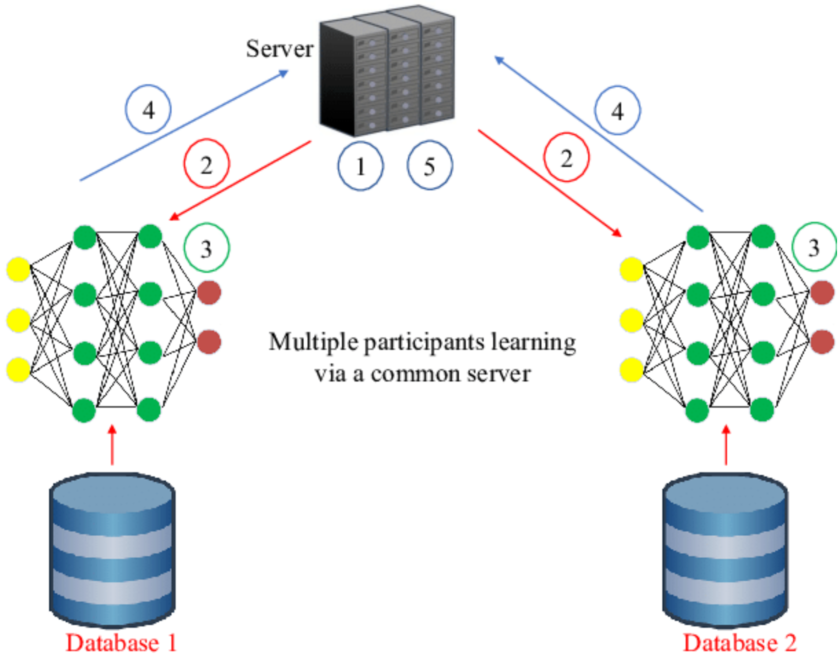

Figure 1.2: Federated Learning System (Horizontal) [2].

Algorithm 1 presents the FedAvg process used in HFL, where clients iteratively update a global model through local training and weighted aggregation.

Algorithm 1 Horizontal Federated Learning with FedAvg [8]

Input: Number of rounds T, number of clients K, local epochs E
Initialize global model weights w_0
for each round $t = 1$ to T **do**
    Server selects a random subset of clients $\mathcal{C}_t$
    for each client $k \in \mathcal{C}_t$ **in parallel do**
        Client receives global model w_t
        Client trains locally for E epochs on local data $\mathcal{D}_k$
        Client returns updated weights w_k
    end for
    Server aggregates updates: $w_{t+1} = \sum_{k \in \mathcal{C}_t} \frac{n_k}{n} w_k$
    Server updates and broadcasts w_{t+1} to all clients
end for

HFL assumes honest participants and a curious server that may try to infer private data. While security measures protect them from these risks, malicious users pose additional privacy challenges. Global model sharing and the corresponding parameters with all participants at the end of training raise exposure concerns [10].

1.6.2 Vertical Federated Learning

Vertical Federated Learning (VFL) is utilized in scenarios where two organizations have information concerning the same individuals but gather varying types of data. An example is when a bank manages financial transactions, while an online shopping website can record consumer buying habits. VFL enables them to collaborate to train a shared model by the secure assembling of their data, with sensitive information kept private. Techniques such as homomorphic encryption secure the process so that neither party sees raw data from the other. This allows both sides to benefit from a more complete prediction model, without sacrificing data confidentiality [10].

As presented in Figure 1.3, the architecture of a VFL system enables parties with different feature spaces to collaboratively train a model through encrypted entity alignment and secure gradient exchange.

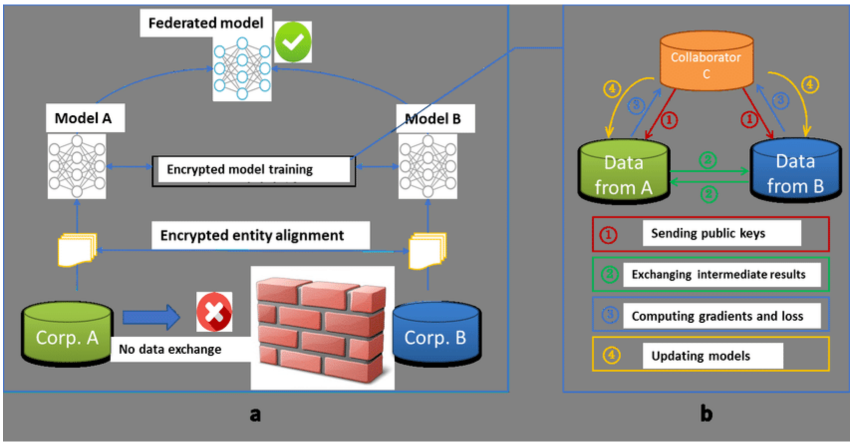

Figure 1.3: Architecture for a vertical federated learning system [2].

Algorithm 2 outlines the secure training process in VFL, where multiple parties with different feature sets collaboratively update a global model through encrypted computations and privacy-preserving alignment.

Algorithm 2 Secure Training in Vertical Federated Learning (VFL) [2]

Multiple parties with aligned users but different feature sets Trained global model with privacy preserved

Step 1: Perform entity alignment using privacy-preserving matching (e.g., PSI)

for each training round $t = 1$ to T **do**

  for each shared user **do**

- Each party computes intermediate outputs from local features
- Parties engage in a secure protocol (e.g., MPC or encryption) to:
  - Compute joint forward pass
  - Calculate joint loss
  - Derive gradients securely

Each party updates its local model component using encrypted gradients Repeat until model convergence.

A VFL system assumes honest-but-curious participants, with no collusion and at most one compromised party. Secure computations are often facilitated by a Semi-honest Third Party (STP), which must remain neutral. After training, each participant retains parameters related to its data, requiring collaboration during inference to produce results. Secure Multi-party Computation (SMPC) protocols ensure privacy throughout the process [10].

1.6.3 Federated Transfer Learning

Federated Transfer Learning (FTL) addresses scenarios where organizations have little or no overlap in users and also collect different features. This occurs when two institutions operate in completely different domains or regions for example, a hospital in Europe and a hospital in Africa.

In FTL, since clients do not share the same user base or feature space, they cannot jointly train a model from scratch. Instead, transfer learning is used: each party trains a model on its own task, and shared knowledge is transferred via shared layers or domain adaptation techniques.

Figure 1.4 illustrates the FTL architecture, where two parties with limited sample and feature overlap collaboratively train a model by transferring knowledge across domains.

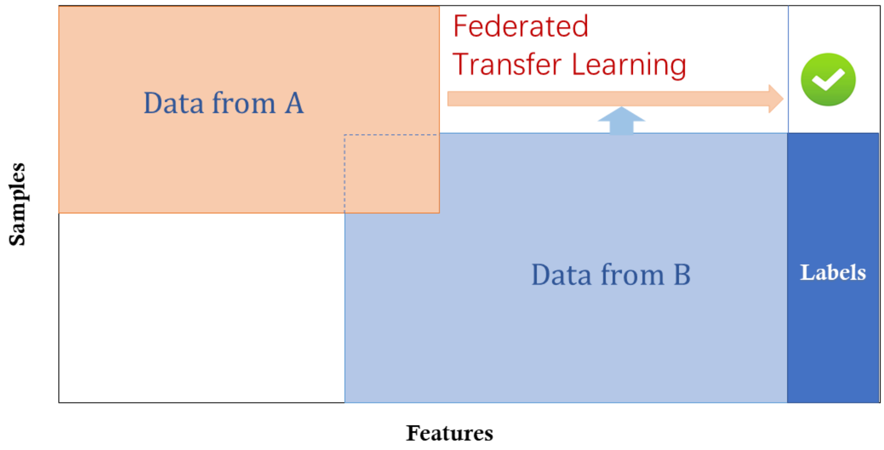

Figure 1.4: Architecture for a transfer federated learning system [2].

Algorithm 3 details the FTL process using shared representations, where clients with different features and users collaboratively align and adapt knowledge across domains.

Algorithm 3 Federated Transfer Learning (FTL) with Shared Representations

---

1: **Input:** Clients with different users and different feature spaces
2: Each client trains a local model on its own data
3: Define a shared representation layer (e.g., encoder or embedding)
4: **for** each training round $t = 1$ to T **do**
5:     Clients update local model parameters
6:     Clients extract and align knowledge using:

- Shared representations
- Knowledge distillation or domain adaptation

7:     Optionally, update and aggregate the shared representation layer
8: **end for**=0

The security definition of FTL builds upon that of VFL. In such a two-party setting, protocols guarantee the privacy of data in that neither party has direct access to the raw data of the other. The systems are dependent on secure computation and privacy-preserving techniques, as utilized in VFL, to guarantee protection on shared information with confidentiality of training and inference [10].

1.7 Federated Aggregation Algorithms

After local training on individual devices or entities, the core challenge in FL is how to aggregate these local model updates into a single global model. The choice of aggregation algorithm significantly affects convergence speed, fairness, robustness, and performance in non-IID (non-identically distributed) settings.

Below are the most common aggregation algorithms and their typical use cases:

1.7.1 Federated Averaging (FedAvg)

FedAvg [8] is the most prevalent aggregation mechanism in FL. It aggregates each client's local model weights using averaging, where the weight of each client is the number of data samples the client has. Aggregation is carried out by the formula

$$w_{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} w_k \ [8]$$

where w_i represents the local model from client i, n_i is the number of samples on that client, and n is the total number of samples across all participating clients. This method is particularly suited for standard FL setups, especially in HFL scenarios, where clients have similar data structures and enough local data and computing power. FedAvg is favored because it is simple, communication-efficient, and performs well when the data is either IID or only mildly non-IID.

1.7.2 Federated Proximal (FedProx)

FedProx [11] is an extension of the FedAvg algorithm, designed to address challenges that arise in FL environments with highly heterogeneous (non-IID) data. It updates the local training goal to include a proximal term that incurs penalties for large deviations of the local model from the global model. Each client optimizes the following loss function during training:

$$\mathcal{L}_i(w) = f_i(w) + \frac{\mu}{2} \|w - w_t\|^2 \ [11]$$

Here, $f_i(w)$ is the local loss function on client i, w_t is the global model at round t, and μ is a tunable parameter that controls the strength of the regularization. The use of this term improves the consistency of both the local and global models, thus ensuring more stable training and combating divergence caused by data heterogeneity. FedProx is particularly useful where client devices are heterogeneous with respect to their data distribution, computation power, or availability.

1.7.3 Federated Normalized Averaging (FedNova)

FedNova [12] strives to provide equitable and balanced aggregation when the clients perform different amounts of local computation, e.g., different numbers of local epochs or different batch sizes. It normalizes the local updates based on the amount of work done by each client and then aggregates them proportionally. Every client computes its normalized update as follows :

$$\Delta w_i = \frac{w_i - w_0}{\eta_i \cdot E_i} \text{ [12]}$$

where η_i is the learning rate, E_i is the number of local epochs, and w_i and w_0 are the local and global models respectively. The global model is then updated as:

$$w_{t+1} = w_0 + \sum_i \alpha_i \Delta w_i \text{ [12]}$$

FedNova is especially relevant to real-world uses of federated learning, where devices have different capabilities and cannot contribute equally to every round. FedNova helps to maintain convergence and model quality under these conditions.

1.7.4 Krum

Krum [13] is a robust aggregation algorithm specifically designed to protect FL from Byzantine (malicious or faulty) clients. Unlike averaging methods, Krum selects a single model update for each round. For each client update, it computes the sum of squared distances to the $n - f - 2$closest other update, where f is the maximum number of faulty clients assumed. The update with the lowest total distance score is chosen as the global model:

$$\text{Score}(i) = \sum_{j \in \text{Closest}(i, n-f-2)} \|w_i - w_j\|^2 \text{ [13]}$$

This method reduces the impact of outliers and tainted updates, making it especially suitable for adversarial environments in which the clients' reliability is uncertain. However, it demands more computational resources and is less effective in benign environments.

1.8 FL-based applications

FL can be used in different fields, and any field that deals with a person's private information can benefit from FL. Below are some of the applications that can use and benefit from FL models.

1.8.1 Healthcare

FL has been increasingly utilized in the healthcare domain for model training while protecting patient data privacy. Some key applications are as follows [14] :Electronic Medical Records, Biomedical Image Analysis, EEG Classification and Clinical Prediction

1.8.2 Internet of Things (IoT)

FL is a potent approach to privacy concerns in the Internet of Things industry, offering secured, decentralized data processing. FL is applied in : smart transportation, smart city and smart grid.

1.8.3 Internet of Underwater Things (IoUT)

FL is used by the IoUT to overcome certain difficulties and extend its applications to several fields, including [14]: environmental monitoring,Underwater Exploration,Disaster Prevention,Defense & Military.

1.8.4 Industrial engineering

Application of FL can be witnessed in a large number of fields: 6G Industry, image detection, image representation, unmanned aerial vehicles, electric vehicles and text mining.

1.9 Conclusion

This chapter explores federated learning by examining the general concepts of artificial intelligence and machine learning. We observed that standard machine learning is helpful but requires all of the data to exist at the same location, creating huge issues with privacy, scale, and obeying the law. Federated learning mitigates these concerns by allowing several users to collaboratively train models without requiring access to their private data.

We examined the FL structure and its distinctions from conventional ML. We examined the principal categories of federated learning, namely horizontal, vertical, and transfer learning. We outlined substantial methods for aggregating outputs and observed that federated learning is becoming progressively beneficial in practical applications, especially in contexts where privacy is a critical issue.

This introduction to FL provides essential information to understand the challenges associated with its implementation.

The subsequent chapter will address the essential problem of security. We examine poisoning assaults and their effects on the dependability and safety of federated learning systems, especially in vital sectors like healthcare.

Chapter 2

Poisoning Attack Detection in Federated Learning for Healthcare Applications : State of the Art

2.1 Introduction

Fl has emerged as a powerful approach preserving data privacy by training ML models across remote clients. By permitting local training on edge devices and sharing only model updates with a central aggregator, FL removes the need to transport sensitive raw data. For industries sensitive to privacy such as healthcare, finance, and autonomous sys tems, this is quite enticing. Conversely, FLs distributed architecture generates significant and novel security problems different from those in traditional centralized learning set tings. Since training occurs across a wide range of consumers with little central authority, malicious actors can change data or model updates to subvert the worldwide model.

Major threats that have been identified include poisoning attacks, where the attackers insert manipulated data or gradients to try and control or bias the model, and inference attacks, where the attackers try to determine sensitive training data by analyzing shared updates [15].

The risks that come with malicious models are particularly troubling in high-stakes domains like healthcare, where such vulnerabilities can lead to misdiagnosis, skewed treatment recommendations, or compromises in patient privacy. As FL adoption particularly in cooperative medical AI applications increases, so does the need to grasp and manage these security concerns. This chapter looks at the security weaknesses of FL concentrating on poisoning attacks, detection methods, challenges, and a review of recent studies relevant to healthcare and other privacy-sensitive settings.

2.2 Security threats in FL

Although FL is meant to improve privacy by storing data on local devices, it is not free of security flaws. Its distributed architecture actually creates new kinds of risks quite different from those in centralized ML systems. Each participants behavior is largely uncontrolled since FLs training process depends on the cooperation of many independent clients. This paves the way for hostile activities that could steal sensitive information from shared model updates or undermine the training process.

One of the most important threats to federated learning (FL) is that the global model can be compromised by malicious clients, and they can use various tactics to sabotage the global model's integrity. This threat is specifically exacerbated by various forms of manipulations, including poisoning of local training data, delivering malicious model updates, or even disturbing the aggregation process. Unless carefully monitored, these can lead to catastrophic model performance degradation or even trigger latent behaviors on validation. Additionally, the exchange of model parameters between the central server and the clients opens the possibility of incidental leakage of confidential information. In some cases, attackers can reconstruct individual features of the local training data from shared gradients or updates. Such a problem is especially pressing in sensitive areas like healthcare or finance, where data confidentiality is highly important, as such inference attacks give rise to serious privacy issues [15].

The natural heterogeneity in client capabilities and information makes mitigation of these attacks much more challenging. It is hard to distinguish between benign shifts in behavior and truly malicious behavior, given that every client has different kinds of information and is run under different circumstances. This uncertainty makes it challenging to design secure and resilient federated learning systems [16].

FL is increasingly gaining traction, particularly in applications where privacy issues need to be carefully addressed. It is therefore critical to identify such security threats and develop effective countermeasures. In the following, we will emphasize poisoning attacks some of the most powerful and stealthy threats to FLand discuss their mechanisms, impacts, and potential defense strategies.

2.3 Poisoning attacks

2.3.1 What is a poisoning attack?

A poisoning attack is a form of adversarial machine learning attack when an assailant deliberately alters the training data or the training process of a learning algorithm to compromise the resultant model. These assaults seek to undermine the model's integrity by diminishing its performance or inducing erroneous behavior under specified conditions.

Poisoning attacks are typically performed during training, where data points explicitly created with ill intent are either added to the training set or modified from original data. The goal could be to make the model incorrectly classify a certain kind of input (targeted poisoning) or reduce the model's overall accuracy (untargeted poisoning) [17]. These attacks are particularly dangerous in automated decision systems, where poisoned models may influence medical diagnoses, financial decisions, or security mechanisms.

As presented in Figure 2.1, the poisoning attack process involves a malicious client submitting tampered model updates during training, resulting in a corrupted global model after aggregation.

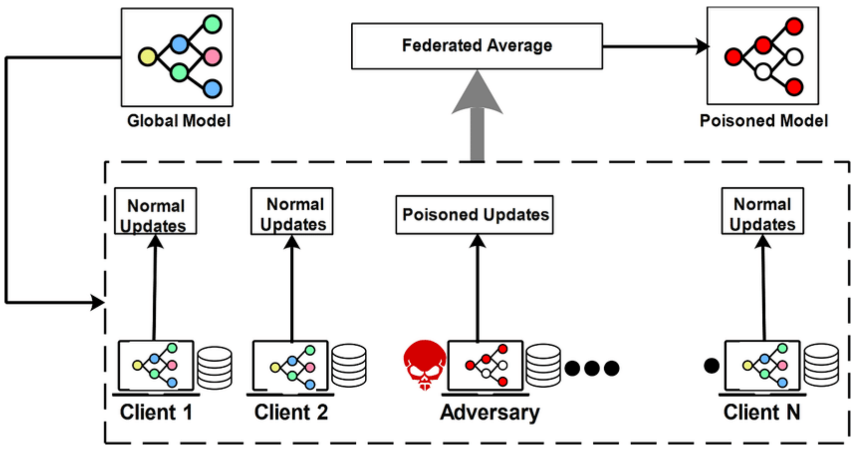

Figure 2.1: Poisoning attack in a FL environment [3].

In general, poisoning attacks are classified as integrity attacks, as opposed to confidentiality or availability threats, because they interfere with the accuracy of the model output. Their stealth and effectiveness make them one of the most critical concerns in ML security.

2.3.2 Types of poisoning attacks

Based on how the attacker disrupts the learning process, poisoning attacks can be categorized into two types: model poisoning and data poisoning. They both attempt to weaken the final model's integrity, but they vary in terms of execution, approach, and access.Furthermore, poisoning attacks can be either targeted or untargeted, depending on the attackers objective.

2.3.2.1 Data poisoning

In data poisoning, the attacker compromises the training set prior to or even in the course of training the model. This typically entails adding adversarial samples or altering existing samples to change the statistical properties of the training set. Since a number of ML models have strong assumptions about data quality and distribution, a very limited amount of carefully designed poisoned data can significantly affect the learned model [17]. There are two common strategies in data poisoning:

- **Label flipping:**Switching the labels of training instances (e.g., marking spam messages as non-spam). This distorts the decision boundary and results in misclassifications. Label flipping can be random (untargeted) or class-selective (targeted).
- **Outlier injection:** Injecting outlier or adversarially crafted points that distort decision boundaries

Data poisoning is an actual threat in the systems that rely upon crowdsourced data or lack rigorous verification protocols. It can either subtly change the model's behavior or directly impair the performance of models, depending upon the intended objective being targeted.

2.3.2.2 Model poisoning

Model poisoning involves directly manipulating the model parameters or training updates during training. This is typically done by attackers who have control over the models training process, or who can inject malicious updates in distributed or FL [18].

A common and powerful model poisoning technique is gradient manipulation. In this strategy, rather than calculating gradients from data exactly, the attacker constructs adversarial gradient updates that are intended to push the global model in a particular direction. These poisoned gradients can be engineered to have an unusually large effect or subtly designed to be statistically benign while being malicious in nature. Some attackers even perform reverse engineering of the loss surface in order to maximize interference while remaining stealthy on aggregation [19].

A very subtle type of model poisoning is called the backdoor attack. In this case, the attacker trains the model to behave normally with normal inputs but emit some outputs chosen by the attacker when a particular trigger is present. For example, an image classifier that classifies images may be taught to always classify any image that has a particular symbol into a particular class [20]. These triggers are usually subtle text excerpts or visual patterns and are not detectable using normal validation. Backdoor attacks are particularly risky since they have high accuracy under normal circumstances and hence may not be detected until deployment.

2.3.2.3 Targeted VS Untargeted VS Semi-Targeted

Poisoning attackseither data or model manipulation can also be categorized by their intent:

- **Targeted poisoning:**It is a method in machine learning where the adversary tries to mislabel certain inputs or data points. A model might be trained, for instance, to consistently mislabel an image of a particular individual or a piece of text with a certain sentence.Backdoor attacks fall in this class because they alter the model in a highly designed and targeted way. Backdoor attacks also fall under the category because they manipulate the model in a very controlled and specific way.

- **Untargeted poisoning:**The attack aims to deteriorate the model's performance in any possible way. This may be accomplished by adding noisy or conflicting data (in data poisoning) or gradient manipulation (in model poisoning) to worsen the model's accuracy over a broad spectrum of tasks [21].

- **Semi-targeted:** The attacker is assigned a class (source class), and he aims to poison the global model so that samples with the source class are identified as a class other than the source class. Unlike a targeted attack, the attacker is free to choose the target class to maximize the attack performance in a semi-target attack.

Targeted attacks tend to be more subtle and difficult to detect, whereas untargeted attacks tend to have a wider and more pervasive impact.

2.3.3 The impact of poisoning attacks in FL

Poisoning attacks in FL can have extensive effects, ranging from a reduction in global model accuracy to the injection of intended misbehaviors that are concealed at the validation stage. Untargeted attacks reduce global performance by introducing noise or adversarial gradients, whereas targeted attacks, i.e., backdoors, mislead the model only under particular circumstances, usually without affecting global accuracy. These threats represent a significant risk in non-IID settings, where inherent variability of updates causes detection of malicious behavior to be difficult. Apart from technical impact, poisoning erodes confidence in FL systemsespecially in safety-critical applications like healthcare or autonomous systemsand can cause irreversible corruption of the global model even after the attacker has left the training process.

Table 2.1 summarizes the effect of poisoning attacks on every step of the FL pipeline, highlighting the risk of model corruption not just at iteration time, but also at deployment time:

Tableau 2.1: Impact of poisoning attacks across FL stages

FL Stage	**Impact of Poisoning Attack**
Local Data Collection	Poisoned or mislabeled data (e.g., label flipping, outliers) leads to inaccurate local training.
Local Training	Adversaries craft malicious models or gradients, potentially embedding backdoors or disrupting convergence.
Model Update Sharing	Attackers send manipulated updates (e.g., scaled gradients or crafted weights) that bias the global model.
Server Aggregation	Poisoned updates skew the global model, especially when aggregation is sensitive or when the attacker holds more data or influence.
Global Model Update	Once updated, the poisoned global model affects all future training rounds, possibly propagating backdoor behavior system-wide.
Model Deployment	The backdoored or degraded model produces incorrect predictions, potentially leading to misdiagnosis, fraud, or security breaches.

2.4 Techniques for detecting poisoning attacks

Ensuring the integrity of its training process has become a major security issue as FL is used in actual applications. Poisoning attacks, in which hostile users intentionally alter data or model updates to compromise the global model, are among the most important threats to FL. A great variety of detection and defense techniques have been suggested in the literature to help to offset these hazards. Usually, these methods fit into three groups: client behavior tracking using trust or reputation systems, anomaly detection mechanisms, and robust aggregation techniques.

Although some of the basic aggregation methods such as Federated Averaging (FedAvg), Federated Proximal (FedProx), and Krumwere presented in Chapter 1 (Section 1.7), this section reviews these and related techniques in the particular framework of adversarial robustness, evaluating their roles and shortcomings in fending against poisoning attacks.

2.4.1 Robust aggregation

Rather than presuming all contributions are honest ,robust aggregation techniques seek to minimize the influence of malicious client updates during the aggregation process.

FedAvg calculates a weighted average of client model updates depending on local data sizes, as mentioned previously. Although efficient, it is not strong and can be readily compromised by hostile changes. A modified version of FedAvg, FedProx adds a proximal term during local training to keep updates nearer to the global model, thus reducing client drift under non-IID data conditions.

Instead of averaging all updates, Krum, a more security-oriented approach, chooses the client update closest to the majority of other updates using Euclidean distance. Though more clients are compromised, Krum is computationally demanding and less effective when attacks are subtle. It does well under the assumption that a small number of clients are malicious.

Trimmed Mean [7] and Median Aggregation [16] also throw away extreme values on a per-parameter basis. While Median Aggregation chooses the median value for each parameter, the Trimmed Mean averages by removing the highest and lowest values across all client updates. Though their efficacy falls when poisoned updates are meticulously designed to seem statistically normal, these techniques are statistically strong and easy to apply.

More recent techniques such as FedNova help to balance client influence by normalizing the impact of each client's update depending on the degree of local computation done. Although not specifically intended as a security tool, this normalizing can provide improved fairness and lower some assault). Although they have advantages, robust aggregation methods usually presume most clients are honest. Their performance declines significantly under coordinated or stealthy assaults, so they are most effective when used in conjunction with other defense layers.

2.4.2 Anomaly detection

Methods of anomaly detection seek to filter or penalize changes on client updates significantly beyond the usual. These plans are statistical, geometric, or learning-based and operate before aggregation.

- **Fundamental distance-based detection:**Utilize cosine similarity [22]or L2 norms [7] for update comparison. In non-IID contexts, honest clients may inherently generate divergent updates, resulting in false positives.

- **FoolsGold [22]:**It is a method to defend against sybil-based poisoning, in which multiple adversarial clients submit similar updates in a bid to dominate the global model. FoolsGold performs similarity checks on update directions and assigns lower weights to clients with suspiciously close updates. It does not require a clean dataset and learns dynamically but may exhibit reduced accuracy in single-client poisoning attacks.

- **FLAME(Federated Learning with Adaptive Model Evaluation) [23]:** Employs a multi-stage adaptive defense mechanism with the responsibility to identify and filter malicious updates in federated learning. FLAME combines anomaly detection with adaptive defense stages, thereby evolving its response according to client behavior over time. FLAME is stable against adaptive and stealthy backdoor attacks compared to fixed-threshold approaches, which makes it feasible for use.

- **Detection via learning:**Autoencoders [24] or one-class SVMs (Support Vector Machine) [25] may be trained using past updates to discern anomalous patterns. These detectors exhibit superior generalization capabilities; nonetheless, they necessitate labeled clean data or substantial server-side processing.

Robustness is improved by anomaly detection but may fail in the presence of large variability in the data. FLAME's multi-phase method offers a more adaptive solution in this environment Robustness is improved by anomaly detection but may fail in the presence of large variability in the data. FLAME's multi-phase method offers a more adaptive solution in this environment

2.4.3 Client behavior tracking and trust systems

This category includes methods that assign trust scores to clients based on their historical behavior and adjust their contribution accordingly. Over time, clients either gain or lose trust depending on how consistent or coordinated their updates are with others or with known-good conduct. Clients with low repute are either excluded or downweighted.

- **FLTrust [26]:** A trustworthy server-side reference dataset created by X. Cao et al [26] is used to calculate a reference gradient direction. Updates from clients are scaled depending on their cosine closeness to this reference. This lets the server prevent likely poisoned updates out-of-alignment without needing any client-side modification. Given access to a limited trusted dataset, FLTrust is particularly powerful in Byzantine settings.

- **Reward and Punishment Systems:** These methods motivate consumers to behave honestly by means of rewards and punishments. Though they might not scale effectively in erratic customer populations, they are most successful in long-term partnerships.

2.5 Challenges in detecting poisoning attacks

This section outlines the primary challenges encountered when trying to detect poisoning attacks in FL systems. Even though many protection techniques have been made, it is still hard to find bad clients in FL since the system is decentralized, there are many different local datasets, and there is not much access to raw data.

2.5.1 Non-IID data distribution

In FL, each clients local data may come from a different distribution, so benign updates can vary greatly across clients. This heterogeneity makes poisoning detection difficult, since a legitimate update from a client with uncommon data may appear anomalous.

Singh et al [27] note that naive anti-poisoning schemes can discriminate [against] minority groups whose data are significantly and legitimately different from those of the majority, yielding poorer models if diverse client updates are excluded [27]. In practice, detectors must therefore account for genuine distributional differences to avoid flagging benign updates as malicious [27].

2.5.2 Limited observability at the server level

The central FL server only receives model updates (gradients or weights), never the raw data or full training histories. As Yurdem et al [28]explain, the information shared with the server is limited to only updated model parameters; therefore, the server does not have access to any clients own training data [28]. This means the server cannot directly inspect or validate the content of local training, and must rely solely on statistical properties of the updates themselves. Without access to ground-truth data or labels, subtle poisoned updates can easily evade detection, since the server has no additional context beyond each client's submitted parameters [28].

2.5.3 Strategic/adaptive attackers

Attackers in FL can tailor their poisoning strategy to the defense in place. For example, KrauSS et al. [29] observe that filtering-based defenses face the challenge of adversaries adapting to the defense mechanisms [29]. In practice, a malicious client can iteratively adjust its update (e.g. via optimization or train-and-scale techniques) to stay within the detection threshold, effectively avoiding static rules. Because there is no single agreed definition of an adaptive adversary, evaluating a defense against such attackers is intrinsically hard [29].In short, strong adversaries will systematically modify their poisoned updates to remain stealthy, making detection a moving target.

2.5.4 Computational and communication resource constraints

Federated clients (often mobile or IoT devices) have limited CPU, memory, and battery, and networks have limited bandwidth. As Liu et al. point out [30], FL must address heterogeneity in device storage, computing and communication capabilities, and communication costs are a major bottleneck [30]. These constraints severely restrict defense algorithms: a heavy anomaly-detector or repeated verification cannot run on a weak device, nor can the server request excessive rounds of updates. In practice, any poisoning-detection scheme must be lightweight and bandwidth-efficient, since complex computation or extra communication (e.g. re-training or secure multiparty checks) may be infeasible on constrained clients [30].

2.5.5 Trade-offs between robustness and model performance

Finally, defenses against poisoning attacks often degrade the models accuracy or convergence speed. Sagar et al [31] note that most defense strategies have trade-offs between attack prevention and the overall performance of the task [31]. For example, adding differential-privacy noise to the aggregation can mask malicious updates but also impairs the performance of the global model [31]. Likewise, aggressively filtering out suspicious updates can remove benign client contributions, reducing overall accuracy. In short, improving robustness usually comes at the cost of model utility, forcing designers to balance security against the primary task performance [31].

2.6 Attacks and security in FL applied in healthcare

FL in healthcare presents security issues that have to be resolved if medical artificial intelligence systems are to be trusted, dependable, and honest. Any model security weakness could result in misdiagnosis, prejudiced treatment, or privacy breaches given the sensitivity of healthcare data [32]. FL is especially important in this regard since it lets many hospitals and medical institutions train AI models without moving raw patient data.

2.6.1 Why security is essential in healthcare FL

To ensure reliability, integrity, and trust in medical artificial intelligence systems, FL in healthcare poses security challenges that must be addressed. Given the sensitivity of healthcare data, any model security flaw could lead to misdiagnosis, biased treatment, or privacy violations [32]. Since FL allows several hospitals and medical facilities to train AI models without transferring raw patient data, it is imperative to ensure data security and robust resistance to adversarial attacks.

There are several reasons why FL systems deployed in healthcare settings are particularly vulnerable and must be secured with strong defense mechanisms:

- High decision-making stakes:Artificial intelligence-driven misdiagnoses have a direct impact on patient welfare [33].
- Different data distributions from heterogeneous sources across many hospitals must be handled using FL models [34].
- Conventional security measures such as centralized validation, are limited by compliance with HIPAA (Health Insurance Portability and Accountability Act), GDPR, and other data security regulations.
- Fully utilizing FL's benefits in healthcare would need analyzing security issues and implementing robust defenses against attackers.

2.7 Related work

This section reviews seven recent studies, published between 2021 and 2024, that focus on detecting poisoning attacks in FL within the context of healthcare applications:

Le Sun et al. [35] suggested FedKC as a personalized federated learning system to address data heterogeneity and model poisoning attacks in the Metaverse for Consumer Health. They employ a client-side defense mechanism, Ant, to counter poisoning attacks through adjusting the Hessian matrix in local training to reduce the impact of adversarial updates on the global model. FedKC contains a k-per customization module based on k-Nearest Neighbors (KNN) [36], enabling clients to leverage their local data in building personalized models. The system exceeds the accuracy of baseline methods like FedAvg and ClusteredFL [37], attaining 84.7% on the CIFAR-10 [38]dataset and 79.5% on the OrganSMNIST dataset [39]. In their testing configuration, five malicious clients were utilized from a total of 100 clients to simulate targeted model poisoning attacks. In just five communication rounds, FedKCs notable robustness against poisoning attempts diminishes the malicious confidence of the global model in both IID and non-IID contexts. In comparison to differential privacy methodologies such as Local Differential Privacy (LDP) [39] and Central Differential Privacy (CDP) [40], FedKC maintains superior honest accuracy while effectively mitigating attacks, rendering it a dependable and precise choice for secure and personalized healthcare applications within the metaverse.

In Yuxia Chang et al.'s research work [41],a blockchain-based federated learning framework was documented in the journal Computational Intelligence and Neuroscience to defend against poisoning attacks in healthcare systems. Their approach involves the utilization of an adaptive differential privacy (DP) algorithm [40]to safeguard data privacy and a gradient verification-based consensus protocol to identify and avert malicious updates. Through the use of blockchain, the system decentralized the process of aggregation, maintaining the global model secure and untampered. The authors evaluated their system on a real-world diabetes dataset and achieved a strong 82.7% accuracy when privacy budget was =3. This was just a bit inferior to baseline FL algorithms like FedAvg, which achieved 84.5% accuracy. Moreover, their solution was highly resilient to poisoning attacks by reducing the success rate of the attack to below 20% even when 30% of the clients were malicious. In comparison to conventional differential privacy (DP)-based methods, their adaptive DP algorithm significantly reduced the privacy budget consumption while maintaining model accuracy, thereby making it a trustworthy option for security and privacy-oriented healthcare applications.

Gan Sun et al [32] investigated data poisoning attacks on federated multi-task learning (FMTL) [42] and introduced a unique optimization technique termed ATFL (AT Tack on Federated Learning). Their framework conceptualizes the poisoning assault issue as a bilevel optimization and uses implicit gradients to determine optimal attack techniques. The authors' approach was experimentally evaluated on real-life datasets viz. EndAD and Human Activity Recognition. Experiments demonstrated that FMTL models are extremely vulnerable to poisoning attacks. Experiments also demonstrated that simple attacks can degrade model accuracy by as much as 26.836% on the Human Activity Recognition dataset and by as much as 21.707% on the EndAD dataset. The ATTFL approach is employed to effectively find the optimal attack strategies, thereby demonstrating the vulnerability of FL systems to poisoning attacks and the need for robust security mechanisms.

In the paper [43], Aditya Pribadi Kalapaaking et al.suggested a blockchain-based secure federated learning framework utilizing Secure Multi-Party Computation (SMPC) to address the threat caused by poisoning attacks on healthcare systems. Their solution achieves a two-layered security through the combination of SMPC's cryptographic privacy guarantee and blockchain's transparency and immutability for private model verification. Each participant gets their model updated and encrypted and verified for possible poisoning via encrypted inference before aggregation, such that only verified models are stored and computed on-chain. In testing, authors used two medical image classification datasets from the MedMNIST suite: TissueMNIST [44] and OCTMNIST [44], each consisting of over 100,000 grayscale 28Œ28 samples. Experiments included up to 50% malicious clients conducting label-flipping attacks. Results indicated the defense proposed herein countered the impact of poisoning attacks by up to 25%, recovering global model accuracy and outperforming baseline FL under adversarial conditions. Moreover, the SMPC-based secure aggregation maintained model performance without significantly increasing inference latency, and the system's permissioned blockchain attained up to 28% quicker consensus compared to proof-of-work systems. This makes their approach highly suitable for resource-constrained healthcare IoT environments, with a trade-off of privacy, scalability, and resilience.

Le Sun et al [33]presented FedCPD, an advanced FL framework for protecting heterogeneous metaverse data. Three novel modules make up their solution: a discrepancy-aware aggregation system (DBAW) that weights contributions based on client performance, a pseudo-dataset correction module to lessen classifier bias, and a cGAN-based privacy protector that substitutes secure generators for vulnerable feature extractors. FedCPD outperformed seven baseline approaches with an accuracy of 74.3% on dermatoscopic data, which was 4.6% higher than FedCG [45], the second-best method, according to the teams evaluation of the algorithm across six different datasets, including DomainNet and Digit5. Interestingly, their DBAW mechanism only showed 8.2% decrease in comparison to IID circumstances, maintaining 71.3% accuracy under extreme non-IID conditions (Dirichlet = 0.5). With label-agnostic pseudo-training, the rectification module improved unknown-class classification by 19.8%, and the privacy module reduced reconstructive attack success rates by 63% over conventional federated learning (FL). According to initial computational evaluations, FedCPD's generator required just 7.95 million floating point operations (FLOPs) per operation, thereby rendering it feasible for metaverse edge devices with low resources. Together, these innovations address essential issues for metaverse federated learning systems by achieving a balance between model robustnessthrough bias-correcting techniquesheterogeneous data management through distribution- aware weighting and privacy preservation through cGAN- based feature obfuscation.

In the study [34], Boumediri et al. examined security threats to healthcare IoT federated intrusion detection systems (FIDS). Through manipulating FL training data, their work proposed a comprehensive threat model showing how poisoning attacks would undermine the security of medical devices. For real-world healthcare datasets, (WUSTL-EHMS-2020 and ECU-IoHT), the authors designed and evaluated three variants of attacks: (1) a backdoor attack that modifies labels to "normal," (2) an availability attack that enforces wrong "attack" labels, and (3) a composite attack that encompasses the two methods above. Their tests showed worrisome flaws: on the WUSTL dataset, attack success rates reached 82% with just 30% of clients poisoned, and model accuracy decreased by 34%. Using a deep neural network architecture (64-64-32 neurons), the teams TensorFlow Federated-based detection solution outperformed traditional IDS by 19.2% and maintained 78.3% precision even in the face of coordinated attacks. It is worth mentioning that their study indicated that feature manipulation attacks outperformed label-flipping in isolation by 27%, therefore necessitating robust defenses for medical IoT networks. The results provide significant new knowledge for defending distributed healthcare systems from novel adversarial attacks without sacrificing patient privacy via federated learning.

In the study [46], Bhabesh Mali et al. have proposed SAFe-Health, a novel defense mechanism for federated learning-based smart healthcare systems against data poisoning attacks. The proposed solution introduces Federated Defense Averaging (FDA), which actively filters out malicious clients by observing loss function deviations during training to aggregate only trustworthy model updates. Tested on a large heart disease database (1,190 samples across 10 hospitals), SAFe-Health demonstrated great robustness and still achieved 85.3% accuracy when 40% of clients were poisoneda 12.5% improvement over vulnerable FL. The system also accurately detected label-flipping and feature tampering attacks with 89% accuracy, and reduced false positives by 23% over approaches such as FedAvg. Their loss-based detection method added barely any extra computation, adding just 8% extra time for each communication round, which makes it practicable for use in real medical cases. This work fills an important gap in the protection of tabular healthcare data, and the experiment confirms it is viable for IID and non-IID data, with just a 3.2% performance drop in very dissimilar settings. These results show that SAFe-Health is a simple yet effective solution to protect FL-based diagnosis models from clinical environment attacks.

2.7.1 Comparative Analysis

Table 2.2 provides a comparative examination of the relative works, structured into eight principal columns for clarity:

- Related work section enumerates each referenced paper, including the authors' names and publication years.
- The proposed method delineates the fundamental methodology or defense plan suggested.
- Approach Security offers a concise overview of the defensive mechanism and its efficacy in mitigating security issues.

- Vulnerabilities underscore the recognized deficiencies or constraints of each method.
- Type of attack delineates the characteristics of the attack(s) examined in the study, specifying whether they were targeted or untargeted via data or model manipulation.
- FL Stage denotes the specific phase of the FL pipeline at which the defense mechanism is implemented
- The dataset used to assess the proposed methodology.
- Accuracy reflects the performance outcomes attained on these datasets when accessible

Related Work	**proposed method**	**Approach & Security**	**Vulnerabilities**	**Dataset**	**Accuracy**	**Type of Attack**	**FL Stage**
L. Sun et al. (2024) [1]	FedKC (client-side defense + k-NN personalization)	Hessian matrix modification, k-NN for personalization	High overhead, potential scalability issues	CIFAR-10,	84.7%	Targeted Model Poisoning	Client Local Training
				OrganSMNIST	79.5%		
				CIFAR-100	59.3%		
				PathMNIST	91.4%		
Y. Chang et al. (2021) [2]	Blockchain + FL	Blockchain aggregation, adaptive differential privacy	Blockchain overhead, gradient leakage risk	PIMA Diabetes	82.7%	Untargeted Data Poisoning	Server Aggregation
G. Sun et al. (2021) [3]	AT²FL (optimized poisoning strategy)	Adversarial bilevel optimization for attack evaluation	No defense, purely attack analysis	EndAD, Human Activity	N/A (no defense, only attack study)	targeted +Untargeted Data Poisoning	Client Local Training
A. Kalapaaking et al. (2023) [4]	Blockchain + SMPC	Model verification using SMPC and blockchain ledger	Encryption overhead, scalability limits	OCTMNIST	72%	Untargeted Model Poisoning	Server Aggregation
				TissueMNIST	65%		
L. Sun et al. (2024) [5]	FedCPD (DBAW + cGAN privacy)	Discrepancy-weighted aggregation, pseudo-data rectification, cGAN privacy	Generative model training overhead	Dermatoscopic,	74.3%	Untargeted Model Poisoning	Client Model Update
				DomainNet,	52.3%		
				Digit5	85.2%		
T. BOUNDEDIRI et al. (2024) [6]	Trust-based filtering	Trust scores + anomaly detection in IoT healthcare	Trust manipulation by attackers	WUSTL-EHMS-2020,	78.3%	Targeted +Untargeted Data & Backdoor Poisoning	Client Model Update
				ECU-IoHT	around 80% to 90%		
B. Mali et al. (2024) [7]	SAFe-Health (Federated Defense Averaging)	Loss function deviation filtering	Potential misclassification of noisy honest clients	Heart Disease Dataset	85.3%	Untargeted Data Poisoning	Client Local Training

Tableau 2.2: Comparative Table of Federated Learning Attacks and Defenses

2.7.2 Discussion and insight

Table 2.2 provides a concise but accurate summary of key contributions, experimental environments, and security results presented in the pertinent literature. The comparative synthesis underscores the variety of solutions conceived for defending federated learning systems from adversary attacks, especially in healthcare and other sensitive domains. By explicitly stating the kind of attack addressed in each paper, e.g., model poisoning, data poisoning, or backdoor attacks, the table clarifies both the technical coverage and the practical impact of each defense approach.

Among the presented works, model poisoning is the most addressed threat, with defenses aiming to filter malicious updates via trust mechanisms, client-specific modeling, or discrepancy-aware aggregation (e.g., FedKC and FedCPD). These works' purpose is to maintain model integrity despite a portion of the clients being malicious. Data poisoning is addressed using techniques such as SAFe-Health, which observes loss function deviation to identify faulty clients with minimal computational overhead.

The surveyed techniques uniformly stress robustness to non-IID data, the fundamental challenge in healthcare environments. For example, FedCPD and FedKC use client-specific evaluation or data correction to maintain performance even when client data distributions differ significantly. This attention to data heterogeneity is especially relevant in multi-institutional medical systems, where demographic, equipment, and procedural variation is common.

In empirical performance, all the frameworks experimented with demonstrated robustness against adversarial attacks. Methods such as SAFe-Health and FedKC preserved model accuracy over 80% even in situations where as many as 40% of the clients had been tampered with. These observations point to the feasibility of deploying federated learning securely in practical healthcare situations, provided there are effective detection and filtering protocols in place.

Nonetheless, challenges remain. Light-weight protections like loss-based or trust-based filtering enjoy computational efficiency at the expense of being vulnerable to misclassifying honest clients under noisy or outlier-dense settings. Meanwhile, complex systems such as FedCPD, which leverage synthetic data generation and multi-module filtering, may be plagued by the scalability problem when deployed on a large scale.

In summary, while each approach comes with some compromises, the best hope for secure federated learning lies in hybrid approaches that bring together robustness, resilience to non-independent and identically distributed data, and low operational overhead. As federated learning systems in healthcare move forward, these solutions offer a strong starting point for mitigating poisoning threats without sacrificing performance or privacy.

2.8 Conclusion

This chapter solely considered data- and model-level poisoning attacks and reviewed the primary security weaknesses of Federated Learning (FL). We reviewed the workings of such attacks and their ensuing impacts, evaluated major detection and defense techniques such as robust aggregation, anomaly detection, and trust-based filtering, and pinpointed the special challenges of defending FL systems against non-IID and privacy-sensitive settings, such as in healthcare. A comparison of recent studies shows that especially under real world limitations, no one approach totally eliminates all adversarial risks. These findings highlight the need for adaptive, lightweight, and context-sensitive protective policies. In response to these challenges, the following chapter introduces our proposed defense strategies, developed to detect and mitigate poisoning attacks during the aggregation stage of Federated Learning. These methods aim to improve robustness while preserving privacy and efficiency in healthcare-oriented FL environments.

Chapter 3

Our Proposed Approach

3.1 Introduction

Federated learning is being ever more widely used in the healthcare sector, enabling hospitals and clinics to collaborate and build models while keeping confidential sensitive information.In this horizontal federation setup, institutions share the same feature space but possess different patient samples. This technology has brought about new security challenges. Poisoning attacks such as label flipping and backdoors are significant threats by potentially negating diagnostic models and endangering patient safety. Traditional federated learning defense strategies generally assume that clients are either honest or willing to cooperate in the execution of security protocols. In practical healthcare settings, such assumptions may not always be valid. Malicious actors may intentionally modify updates, and the varied characteristics of medical data further complicate the detection of anomalies. This chapter initiates the presentation of a comprehensive server-side defense framework for healthcare federated learning to tackle these challenges. Our contribution puts forward three distinct server-side defense strategies against poisoning threats in which the first proposes a purely robust aggregation method, the second offers a detection-driven strategy utilizing standard aggregation, and the third presents a hybrid adaptive defense that can adjust aggregation techniques according to threat levels. These solutions are all designed to operate independently of client-side control, making them highly suitable for real-world healthcare scenarios. Unlike many contemporary methodologies that rely heavily on fixed client profiles or privacy trade-offs, our system suite offers adaptive and effective protection. These approaches are designed to manage non-IID medical data and safeguard AI-driven diagnostics against complex poisoning threats in sensitive healthcare environments.

3.2 Overall architecture

Our proposed methodology for ensuring the security of federated learning in healthcare applications features a modular architecture (as illustrated in Figure 3.1) in consisting of the following key phases :

- **Federated Data Preparation:** Each healthcare dataset is allocated among simulated clients in a non-IID fashion to replicate real-world institutional variability.in horizontal federated learning scenarios where clients have the same medical features but different patient populations. The dataset is partitioned into training, validation, and test sets to enable effective learning and performance assessment.

- **Federated Simulation and Local Training:** Clients acquire an exact duplicate of the global model and perform local training utilizing their proprietary data. Some of these clients are called malicious and they take part in flipping labels or backdoor injections to carry out real poisoning attacks.

- **Secure Server-Side Defense Pipeline:** Upon transmission of local updates to the server, they are processed through one of three proposed server-side defense approaches:

  - **Approach 1-Pure robust aggregation:** This approach applies strong statistical aggregation (e.g., trimmed mean) directly to the client updates received. It does not have any detection mechanisms and assumes robustness purely through aggregation. The goal is to see if such methods by themselves can mitigate attacks.
  - **Approach 2- Anomaly Detection + Simple Aggregation:** The system first does multimodel detection using gradient statistics, spectral analysis and STRIP (STRong Intentional Perturbation) [47] entropy-based testing. After filtering out suspicious updates, simple FedAvg is applied for aggregation.
  - **Approach 3-Hybrid Adaptive deffense:** This adaptive strategy starts with the same anomaly detection techniques as in Approach 2. Based on the threat level it dynamically picks between FedAvg (when all malicious updates are detected and removed) and strong mixing (when doubts remain). The choice of method changes from round to round based on real-time analysis.

- **Global Model Update and Evaluation:** Following successful validation, the model is distributed to clients for the subsequent training phase. The system's

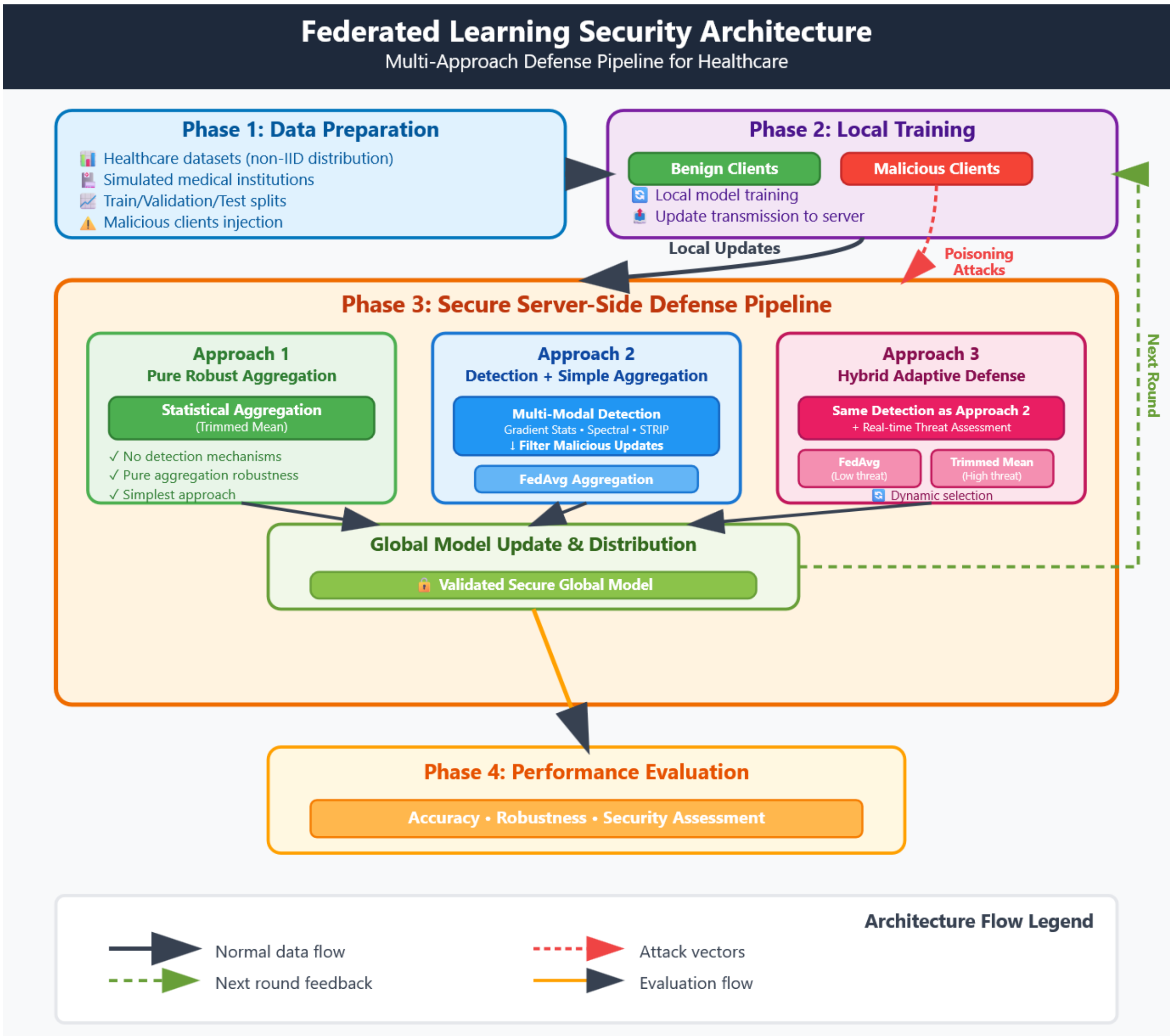

Figure 3.1: Overall architecture of our solution.

performance is systematically evaluated with clean test data to assess its accuracy, robustness, and resilience against attacks.

As shown in Figure 3.1, this architecture ensures the scalability of federated learning and preserves the reliability of models employed in sensitive healthcare environments.

3.3 Data preprocessing

Healthcare Data sees preprocessing challenges in federated learning due to varied methodologies, equipment calibration, and different protocols. Incomplete and ill-formatted data can lead to inaccurate predictions thus compromising patient safety. Advanced systems overcome these challenges while enabling effective attack detection.

3.3.1 Healthcare-Specific Data Preprocessing Pipeline

The data preparation pipeline transforms raw healthcare data into model-ready training sets by employing techniques such as cleaning, normalization, augmentation, and distribution analysis. This method, as outlined in Algorithm 1, ensures data integrity, enhances federated learning, and improves threat detection capabilities.

Algorithm 4 Data Pre-processing Pseudocode

```
procedure PrepareDataset(dataset_collection)
    Input: dataset_collection
    Output: federated_client_datasets, distribution_metrics
    raw_data ← LoadDataset(dataset_collection, transforms=True)
    train_data, val_data, test_data ← SplitDataset(raw_data, val_ratio=0.1)
    for client_id = 1 to num_clients do
        client_samples ← CreateNonIIDDistribution(train_data, α = 0.5)
        if len(client_samples) < MIN_SAMPLES then
            client_samples ← ResampleClientData(client_samples, MIN_SAMPLES)
        end if
        for all class_label ∈ client_samples do
            class_count ← count(class_label)
            if class_count < 10 then
                additional_samples ← AugmentClassSamples(class_label, target_count=10)
                client_samples ← client_samples ∪ additional_samples
            end if
        end for
        institutional_variation ← AddInstitutionalNoise(client_samples, noise_factor=0.02)
        client_dataset ← (institutional_variation.x, institutional_variation.y)
        federated_client_datasets ← federated_client_datasets ∪ client_dataset
    end for
    distribution_metrics ← AnalyzeDataDistribution(federated_client_datasets)
    heterogeneity_index ← CalculateNonIIDDegree(federated_client_datasets)
    return federated_client_datasets, distribution_metrics
end procedure
```

As shown in algorithm 4 the PrepareDataset procedure converts raw medical data into client distributions suitable for federated learning through four essential processes. Initially, it loads and preprocesses the medical dataset with suitable transformations, thereafter dividing it into training, validation, and test sets. Secondly, it generates realistic non-IID distributions across customers by Dirichlet sampling ($\alpha = 0.5$) to emulate the heterogeneity of real-world medical data. Third, data quality is guaranteed via selective augmentation: clients with inadequate samples are resampled to fulfill minimum criteria, and underrepresented classes (fewer than 10 samples) are provided with synthetic replicas to preserve class equilibrium. Fourth, institutional variation is added by controlled noise injection (factor=0.02) to replicate diverse client settings. The final federated datasets preserve statistical diversity while offering distribution metrics and heterogeneity indices crucial for the improved defense system, ensuring compatibility with federated learning and a solid baseline for attack detection.

3.3.2 Data Cleaning and Validation

Our framework incorporates an effective medical data cleaning process to ensure standardized quality and consistency of medical images at all federation clients :

- **Basic Preprocessing:**This phase standardizes data values, makes basic changes to healthcare data (such as normalization and scaling), and ensures that all client datasets have consistent data structure and dimensions.
- **Medical Data Validation:** Ensures the integrity of data samples, verifies the correctness of medical record data format and structure, and confirms the consistency of class labels across all categories within the dataset.

3.3.3 Data Augmentation for Class Balance

To address class imbalance and enhance model robustness, our framework leverages medical augmentation methods :

- **Data Transformations:**Appropriate transformations are applied based on the data type - including rotation and flipping for image data, feature scaling and noise insertion for tabular data, and time translation for sequence data - while maintaining the critical medical characteristics necessary for accurate classification.
- **Adaptive Augmentation:** Semantic-preserving modifications are used to increase diversity in potentially underrepresented categories. This ensures basic category representation while maintaining medical validity and clinical relevance.

3.3.4 Dataset Splitting and Distribution

The preprocessed medical dataset is prepared according to standard federated learning procedures :

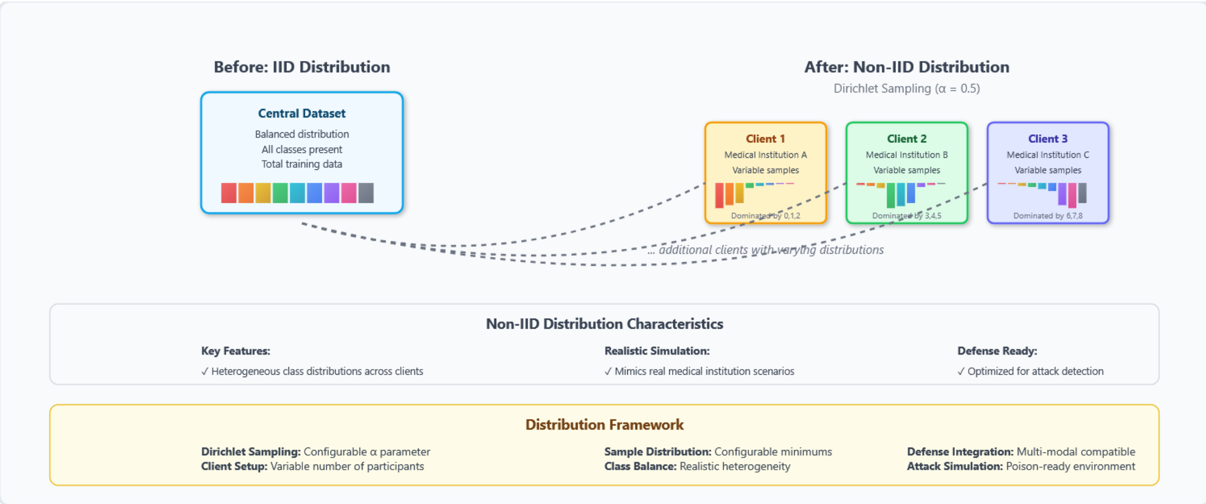

Figure 3.2: Non-IID Data Distribution Across Federated Clients.

- **Federated Distribution:** Use pre-split train/validation/test datasets, but only distribute training data to federated clients using non-IID Dirichlet sampling (as shown in Figure 3.2). Validation data is processed on the server side, while test data is used for global model evaluation.
- **Stratified Distribution:**Ensures that each client receives samples from a range of representative categories, thereby preserving typical heterogeneity patterns commonly seen in medical settings.

3.4 Model Local training

Our proposed approaches use Convolutional Neural Networks (CNN) [48] for addressing healthcare federated learning security issues and federated aggregation mechanisms. A CNN architecture is used for our dataset within a federated learning framework that integrates with our proposed federated defense approaches. The architecture will be thoroughly explained in subsequent sections.

3.4.1 Proposed CNN Architecture

The proposed architecture is a CNN, a deep learning model that is widely used to classify medical images since it is good at handling grid-like data formats such as images. It consists of convolutional layers for feature extraction, pooling layers to reduce spatial depth, and fully connected layers to accomplish final classification. CNNs show very promising effectiveness in the histopathological examination, as accurate tissue type classification relies on spatial features extraction. Our CNN architecture is designed for federated learning situations. It strikes a balance between model expressiveness and communication efficiency in order to make certain that learning is effective and that bandwidth needs are kept to a minimum among distant healthcare facilities. Our CNN architecture consists of three main components: **Input Layer:** Accepts the preprocessed dataset.

3.4.1.1 Feature Extraction Blocks

Feature extraction uses three layers of convolutional blocks, each with a greater complexity in filters. Every block is made up of two convolutional layers featuring small receptive fields, batch normalization to enhance training stability [49], a ReLU activation function to introduce nonlinearity [50], max pooling for reducing spatial dimensions, and dropout to help with regularization [51].This architecture effectively captures a range of features, from intricate cellular details to more expansive tissue-level patterns. Hierarchical features, from basic edge detection to complex tissue patterns, are extracted in three convolutional blocks with batch normalization that ensures stable training in any environment of heterogeneous clients.

3.4.1.2 Adaptive Pooling Layer

Adaptive average pooling layers assist in standardizing the dimensions of feature maps, making them less affected by variations in image sizes from different medical centers [52]. This design takes into consideration the distinctions in preprocessing and the differences in image resolution found in federated healthcare settings, in order to guarantee that feature representation remains consistent across various clients. Adaptive pooling makes sure that feature dimensions stay the same in all settings and any future preprocessing variation between different medical institutions

3.4.1.3 Classification Head

The classification component includes a fully connected layer that utilizes batch normalization and dropout regularization, leading into a final classification layer that employs softmax activation [53]. This head connects the extracted features to probability distributions for datasets tissue classes, allowing for reliable classification of tissue types. Final dense layers with 512 neurons process extracted features to output probability distributions over 9 tissue classes. **Regularization:**Overfitting is averted through the use of dropout layers acting at two different rates, 0.25 and 0.5, related more generally to the non-IID data distributions typical in federated healthcare scenarios.

The architecture meets the requirements of federated learning: it ensures efficient parameter communications, maintains robustness against non-IID data distributions and can work seamlessly with the 3-layer defense system to identify malicious updates in collaborative training.

(1).png (1).bb

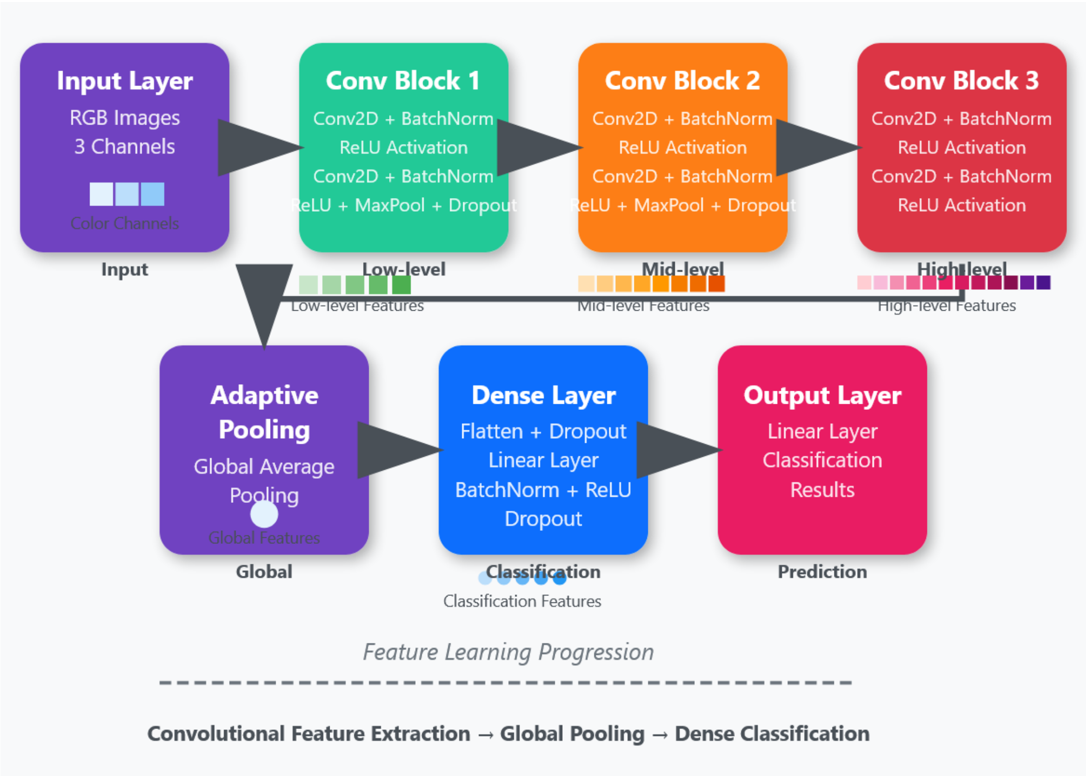

Figure 3.3: Our proposed CNN architecture for FL in healthcare environments .

3.5 Proposed Defense Architectures for Healthcare Federated Learning :

To address the pressing safety concerns in collaborative medical AI environments, we examine three distinct server-side strategies designed for shared healthcare contexts. Each approach presents a distinct perspective on achieving balance among power, adaptability, and computational efficiency.

3.5.1 Approach 1 Pure Robust Aggregation :

This approach investigates the use of pure robust aggregation [16] as a standalone defense against poisoning attacks in federated learning, with a specific focus on healthcare scenarios (see Figure 3.4). Unlike traditional solutions that rely on detection before applying aggregation, this method assumes no detection mechanism and instead depends solely on the mathematical robustness of the aggregation function to filter out malicious updates.

The main goal is to evaluate whether a trimmed mean strategy [54], applied directly during server-side model aggregation, can maintain model integrity in the presence of adversarial clients. This method is particularly suitable for settings that require minimal architectural complexity and where interpretability, reproducibility, and computational efficiency are essential, such as hospital networks that collaborate to train medical AI models. By design, this approach provides a strong baseline for evaluating the necessity of more advanced detection-based or adaptive defenses.

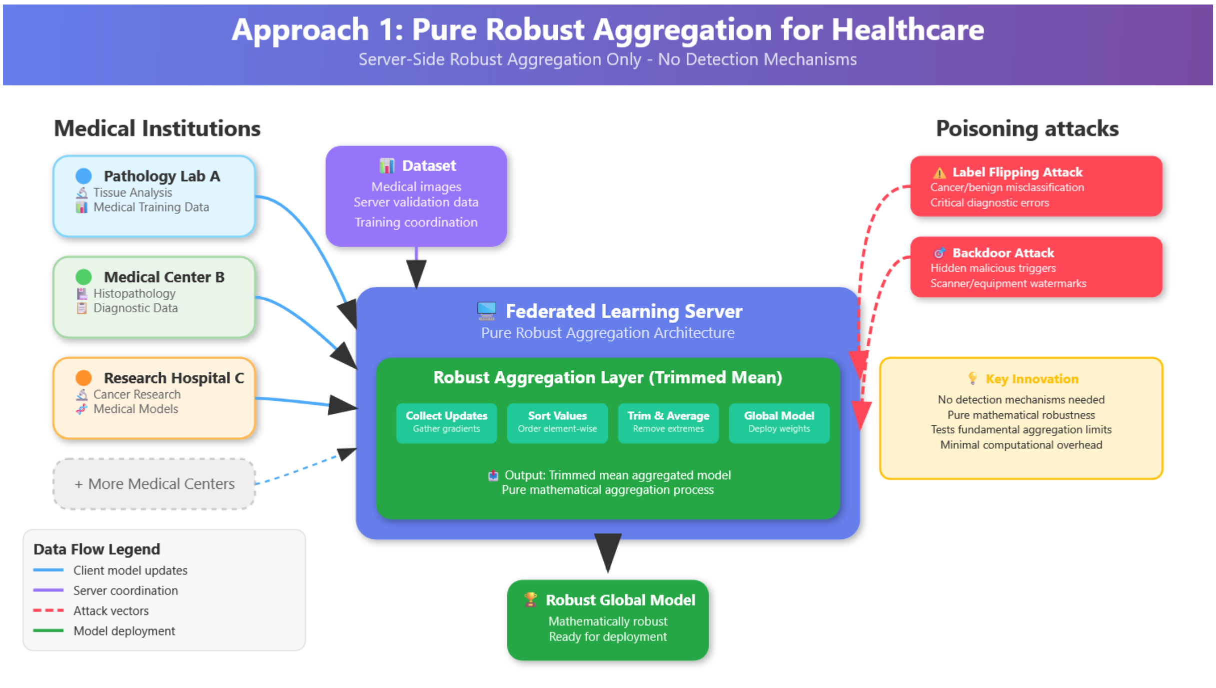

Figure 3.4: Pure Robust aggregation for healthcare.

Notation: Throughout this work, N refers to the number of participating clients selected from the total M available healthcare institutions in each communication round.

3.5.1.1 Byzantine-Robust Statistical Aggregation Theory:

The pure robust aggregation approach draws its theoretical foundation from Byzantine fault tolerance in distributed systems and robust statistical estimation theory. Unlike consensus-based Byzantine fault tolerant algorithms that require explicit agreement protocols, this approach leverages the inherent robustness properties of statistical estimators to achieve resilience against adversarial participants [55].

1. **Byzantine Attack Model in Healthcare FL:** Consider a federated learning system with M medical institutions in the healthcare network. At each communication round, the server selects N participating medical institutions (where N M) based on availability and participation criteria, where at most $f < N/2$ institutions may be Byzantine (malicious), Let $\mathcal{B} \subseteq 1, 2, ..., N$ denote the set of Byzantine clients with $|\mathcal{B}| \leq f$. In healthcare contexts, Byzantine behavior may manifest as:

   - **Label flipping attacks:**where cancer/benign classifications are systematically flipped.
   - **Data poisoning**through injection of synthetic medical images with incorrect diagnoses.

2. **Robust Statistical Estimation Framework:**The mathematical foundation relies on :

   - **breakdown point theory**from robust statistics [56]. For any estimator $\hat{\theta}$, the breakdown point ϵ^* represents the maximum fraction of contaminated data that the estimator can tolerate before producing arbitrary results. The trimmed mean estimator achieves a breakdown point approaching 50%, providing theoretical guarantees against minority adversarial coalitions.
   - **Trimmed Mean Aggregation:** Consider a set of participating clients D = C, C, ..., C selected from the M available healthcare institutions, where |D| = N represents the number of participating clients in the current round. Each client C locally trains a learning algorithm (e.g., CNN) on its private data and obtains a model with parameters . At each communication round t, clients transmit their parameter updates to the central server. Given client updates $\theta_1^{(t)}, \theta_2^{(t)}, ..., \theta_N^{(t)}$ at round t, the trimmed mean operates element-wise on parameter vectors. For each parameter component j:

     $$\theta_{(1),j}^{(t)} \leq \theta_{(2),j}^{(t)} \leq ... \leq \theta_{(N),j}^{(t)}$$

     With trimming parameter $\beta \in (0, 0.5)$ and $k = \lfloor \beta N \rfloor$:

     $$\hat{\theta}j^{(t+1)} = \frac{1}{N-2k} \sum i = k + 1^{N-k} \theta_{(i),j}^{(t)}$$

The choice of β balances robustness against efficiency in federated learning. Trusted networks can use conservative trimming ($\beta = 0.1 - 0.2$) for adequate protection while preserving convergence, whereas open consortiums with uncertain participants require more aggressive trimming ($\beta = 0.3 - 0.4$) for stronger guarantees at the cost of slower convergence.

- **Convergence Analysis:** Under the assumption that $f < \lfloor \beta N \rfloor$, the trimmed mean aggregation maintains convergence properties similar to standard federated averaging while providing robustness guarantees [16]. The convergence rate follows:

$$\mathbb{E}[F(\theta^{(T)})] - F(\theta^*) \leq \frac{C}{\sqrt{T}} + O(\epsilon_{\text{Byzantine}})$$

where $\epsilon_{\text{Byzantine}}$ represents the bounded impact of Byzantine participants and C depends on problem parameters.

- **Healthcare-Specific Robustness Properties:**In medical federated learning, the trimmed mean provides several critical advantages:
  - **Diagnostic accuracy preservation** under systematic label manipulation
  - **Treatment bias mitigation**when adversaries attempt to skew model recommendations
  - **Regulatory compliance**through deterministic and auditable aggregation procedures.

3. Robust Statistical Estimation Framework:The mathematical foundation relies on :

- **breakdown point theory**from robust statistics [56]. For any estimator $\hat{\theta}$, the breakdown point ϵ^* represents the maximum fraction of contaminated data that the estimator can tolerate before producing arbitrary results. The trimmed mean estimator achieves a breakdown point approaching 50%, providing theoretical guarantees against minority adversarial coalitions.

- **Trimmed Mean Aggregation:** Consider a set of participating clients D = C, C, ..., C selected from the M available healthcare institutions, where |D| = N represents the number of participating clients in the current round. Each client C locally trains a learning algorithm (e.g., CNN) on its private data and obtains a model with parameters . At each communication round t, clients transmit their parameter updates to the central server. Given client updates $\theta_1^{(t)}, \theta_2^{(t)}, ..., \theta_N^{(t)}$ at round t, the trimmed mean operates element-wise on parameter vectors. For each parameter component j:

$$\theta_{(1),j}^{(t)} \leq \theta_{(2),j}^{(t)} \leq ... \leq \theta_{(N),j}^{(t)}$$

With trimming parameter $\beta \in (0, 0.5)$ and $k = \lfloor \beta N \rfloor$:

$$\hat{\theta} j^{(t+1)} = \frac{1}{N - 2k} \sum i = k + 1^{N-k} \theta_{(i),j}^{(t)}$$

The choice of β balances robustness against efficiency in federated learning. Trusted networks can use conservative trimming ($\beta = 0.1 - 0.2$) for adequate protection while preserving convergence, whereas open consortiums with uncertain participants require more aggressive trimming ($\beta = 0.3 - 0.4$) for stronger guarantees at the cost of slower convergence.

- **Convergence Analysis:** Under the assumption that $f < \lfloor \beta N \rfloor$, the trimmed mean aggregation maintains convergence properties similar to standard federated averaging while providing robustness guarantees [16]. The convergence rate follows:

  $$\mathbb{E}[F(\theta^{(T)})] - F(\theta^*) \leq \frac{C}{\sqrt{T}} + O(\epsilon_{\text{Byzantine}})$$

  where $\epsilon_{\text{Byzantine}}$ represents the bounded impact of Byzantine participants and C depends on problem parameters.

- **Healthcare-Specific Robustness Properties:**In medical federated learning, the trimmed mean provides several critical advantages:
  - **Diagnostic accuracy preservation** under systematic label manipulation
  - **Treatment bias mitigation**when adversaries attempt to skew model recommendations
  - **Regulatory compliance**through deterministic and auditable aggregation procedures.

4. **Pure Robust Aggregation Protocol:**

Algorithm 5 Trimmed Mean Federated Learning

Require: Number of clients N, trimming ratio β, learning rate η
Ensure: Global model θ

Initialize global model $\theta^{(0)}$
for round $t = 0, 1, 2, \ldots$ **do**
  for all client $i \in \{1, \ldots, N\}$ **in parallel do**
    Download $\theta^{(t)}$ from server
    Compute local update: $\theta_i^{(t+1)} \leftarrow \text{LocalUpdate}(\theta^{(t)}, \mathcal{D}_i)$
    Send $\theta_i^{(t+1)}$ to server
  end for
                    ▷ Server-side robust aggregation
  for each parameter index j **do**
    Sort $\{\theta_{1,j}^{(t+1)}, \ldots, \theta_{N,j}^{(t+1)}\}$
    $k \leftarrow \lfloor \beta \cdot N \rfloor$
    $\theta_j^{(t+1)} \leftarrow \frac{1}{N-2k} \sum_{i=k+1}^{N-k} \theta_{(i),j}^{(t+1)}$
  end for
  Broadcast $\theta^{(t+1)}$ to all M clients
end for

Computational Complexity Analysis: The computational complexity of the trimmed mean aggregation algorithm 5 is dominated by the sorting operation:

- **Time Complexity:**$O(d \cdot N \log N)$ where d is the parameter dimension
- **Space Complexity:** $O(d \cdot N)$ for storing client updates
- **Communication Complexity:**$O(d)$ per participating client per round

This represents a significant computational advantage over complex detection mechanisms while maintaining robustness guarantees against Byzantine participants [16], [57].

Robustness Analysis: Beyond efficiency, the method offers proven Byzantine resilience through its breakdown point properties :

- **Byzantine Resilience Properties:**
  - **Property 1 (Breakdown Point):**The trimmed mean estimator with trimming ratio β can tolerate up to $\lfloor \beta N \rfloor$ Byzantine clients without breakdown, providing theoretical guarantees against minority adversarial coalitions [55], [57].
  - **Property 2 (Bounded Influence):** The influence of any single client update on the global model is bounded by $\frac{1}{N-2k}$, providing inherent regularization against extreme deviations and preventing any individual com-

promised institution from disproportionately affecting the collaborative model [55], [54].

- **Attack-Specific Resilience:** In this work, we focus on two primary attack vectors relevant to healthcare federated learning environments:
  - **Label Flipping Attacks:** Systematic label corruption in local training (e.g., misclassifying malignant tumors as benign) manifests as consistent gradient deviations from the true learning objective. The trimmed mean naturally filters these extreme gradients by removing outliers from both ends of the parameter distribution, maintaining diagnostic accuracy even when multiple institutions are compromised [58].
  - **Backdoor Attacks::** Hidden triggers embedded in model parameters typically require coordinated manipulation across specific parameter subsets to activate malicious behavior. The element-wise trimming operation disrupts these coordinated manipulations by independently processing each parameter dimension, making it difficult for adversaries to inject coherent backdoor patterns while evading detection [59].

5. **Advantages and Limitations Analysis:** To assess the practical applicability of trimmed mean aggregation in federated healthcare systems, Table 3.1 summarizes its key advantages and fundamental limitations. The table also provides a contextual assessment tailored to medical environments, highlighting the trade-offs between robustness, efficiency, and adaptability.

Tableau 3.1: Advantages and Limitations of Trimmed Mean Aggregation in Healthcare FL

Aspect	**Details**
Key Advantages	• **Simplicity and Reliability:** Eliminates detection complexity, making it easier to deploy in healthcare systems while offering resistance to sophisticated evasion attacks [8]. • **Mathematical Guarantees:** Offers provable robustness up to 50% Byzantine clients, aligning with safety-critical requirements in medical AI [16, 57]. • **Computational Efficiency:** Low overhead and suitable for real-time operation in resource-constrained healthcare networks.
Fundamental Limitations	• **Static Defense Paradigm:** Always applies the same robustness penalty, even in the absence of attacks, which reduces efficiency during benign periods [60]. • **Convergence Trade-offs:** Slower convergence than standard FedAvg, possibly requiring more rounds for medical-level accuracy [61]. • **Scalability Constraints:** Less effective against large-scale or coordinated attacks in broad healthcare consortiums.
Healthcare Context Assessment	• Well-suited for controlled medical environments prioritizing safety over speed. • May be inadequate for dynamic or large-scale collaborations requiring adaptive defenses.

3.5.2 Approach 2 Detection + Standard Aggregation:

This approach investigates the use of comprehensive attack detection followed by standard aggregation as a proactive defense against poisoning attacks in federated learning, with a specific focus on healthcare scenarios (see Figure 3.5). Unlike robust aggregation methods that tolerate contaminated inputs, this method employs multi-modal detection mechanisms to identify and eliminate Byzantine participants before applying vanilla FedAvg aggregation to the filtered clean updates. The main goal is to evaluate whether a

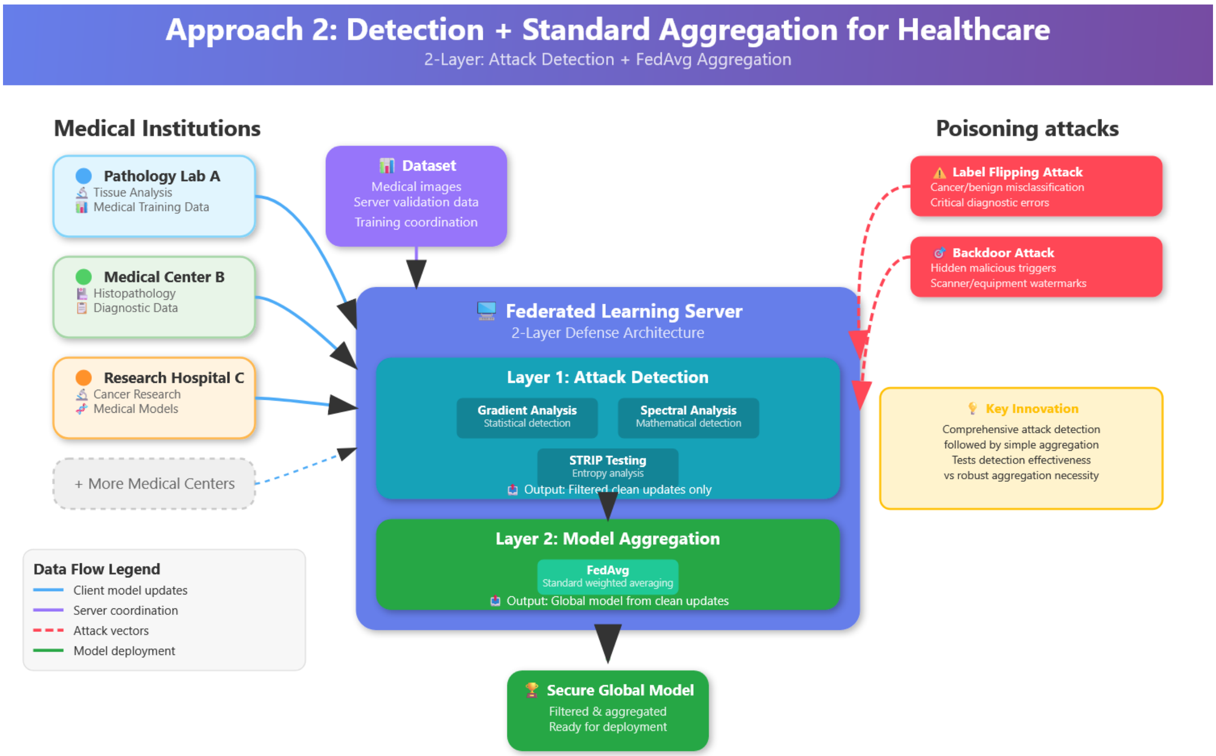

Figure 3.5: Anomaly Detection+Simple Aggregation for healthcare.

layered detection strategy, combining gradient analysis, spectral filtering, and backdoor testing, can achieve perfect separation between legitimate and malicious clients while preserving the computational efficiency of standard federated averaging. This method is particularly suitable for settings that require explicit auditability and regulatory compliancesuch as medical institutions subject to FDA oversight where detection decisions must be traceable and interpretable. By design, this approach tests the fundamental hypothesis that proactive threat identification can eliminate the need for robust aggregation entirely, providing a critical evaluation of detection-first versus tolerance-based defense paradigms in healthcare federated learning.

3.5.2.1 Attack Detection Theory:

The detection-based approach draws from anomaly detection theory and statistical hypothesis testing to identify Byzantine participants before aggregation. Unlike robust aggregation that tolerates adversarial inputs, detection mechanisms aim to achieve perfect separation between legitimate and malicious updates, preserving the optimality of standard federated averaging [62].

- **Theoretical Foundation:** Each client update $\theta_i^{(t)}$ can be modeled as a sample from either a legitimate distribution $P_{\mathcal{H}}$ (null hypothesis) or an adversarial distribution $P_{\mathcal{B}}$ (alternative hypothesis). The detection problem becomes a binary classification task with fundamental limits governed by the Kullback-Leibler divergence between these distributions [14]:

$$D_{KL}(P_{\mathcal{H}}|P_{\mathcal{B}}) = \mathbb{E}P\mathcal{H}\left[\log\frac{P_{\mathcal{H}}(\theta)}{P_{\mathcal{B}}(\theta)}\right]$$

  Higher divergence enables more reliable detection, while sophisticated adversaries attempt to minimize this separability through distribution matching attacks.

- **Multi-Modal Detection Architecture:** The framework employs three complementary detection mechanisms operating in parallel to maximize attack coverage:
  - **1. Gradient Analysis Detection:** Statistical analysis of gradient properties exploits the fact that adversarial updates typically exhibit extreme deviations from expected parameter distributions.
    - **Cosine Similarity Test:** Measures angular deviation between client updates and the global model direction:

$$\cos(\theta_i^{(t)}, \theta^{(t-1)}) = \frac{\theta_i^{(t)} \cdot \theta^{(t-1)}}{|\theta_i^{(t)}|_2 |\theta^{(t-1)}|_2}$$

      [63] Legitimate updates maintain bounded angular deviation, while label-flipping attacks produce gradients with reversed directional components, resulting in negative cosine similarity values [20].
  - **Gradient Norm Analysis:** Compares parameter magnitudes against expected statistical bounds: $\|\text{Score}_i = \frac{|\theta_i^{(t)}|_2}{\text{median}|\theta_j^{(t)}|2j{=}1^N}$Byzantine attacks often generate gradient norms significantly above or below legitimate ranges.

- **2. Spectral Analysis Detection:** Exploits geometric properties of parameter spaces where Byzantine updates typically occupy distinct subspaces compared to legitimate updates [13].
  - **Principal Component Analysis:**Constructs client update matrix $\mathbf{U}^{(t)} = [\theta_1^{(t)}, \theta_2^{(t)}, ..., \theta_N^{(t)}]$ and computes singular value decomposition:

$$\mathbf{U}^{(t)} = \mathbf{P}\mathbf{\Sigma}\mathbf{Q}^T$$

Byzantine updates cluster separately in the principal component space, enabling geometric separation through spectral clustering. The detection criterion measures distance to the dominant honest subspace:

$$d_{\text{spectral}}(i) = |\theta_i^{(t)} - \text{proj}_{\text{honest}}(\theta_i^{(t)})|_2$$

[13].

- **3.STRIP Testing:**Evaluates model robustness against backdoor triggers by analyzing prediction entropy under input perturbations [47]: For input x and model f, generate perturbed versions $x_1', x_2', ..., x_k'$ and compute prediction entropy: $\text{Entropy}(f) = -\sum_i p_i \log p_i$ [47]. Backdoored models maintain consistent predictions despite perturbations (low entropy variance), while legitimate models show natural entropy variation across perturbed inputs.
  - **Standard Aggregation on Filtered Updates:**After detection filtering, the approach applies FedAvg aggregation on the remaining legitimate clients $\mathcal{S}^{(t)}$:

    $$\theta^{(t+1)} = \sum_{i \in \mathcal{S}^{(t)}} w_i^{(t)} \theta_i^{(t+1)}$$

    [8] where weights are typically proportional to dataset sizes: $w_i^{(t)} = \frac{|\mathcal{D}i|}{\sum j \in \mathcal{S}^{(t)} |\mathcal{D}_j|}$. This preserves the computational efficiency and convergence properties of standard federated learning when detection successfully filters all adversarial participants [8].
  - **Detection Integration Strategy:** The three detection mechanisms operate with different sensitivities and attack coverage. Gradient analysis provides efficient first-line screening, spectral analysis catches coordinated attacks, while STRIP testing specifically targets backdoor vulnerabilities. The combination achieves comprehensive attack detection through complementary detection principles.

3.5.2.2 Detection-Aggregation Protocol

The detection-based federated learning protocol operates through a two-stage architecture: comprehensive multi-modal detection followed by standard aggregation on filtered updates. This design prioritizes proactive threat identification while maintaining the computational efficiency of vanilla federated averaging.

Algorithm 6 Detection-Based Federated Learning

Require: Number of clients N, detection thresholds Γ, learning rate η
Ensure: Global model θ

Initialize global model $\theta^{(0)}$, detection parameters Γ
for each round $t = 0, 1, 2, \ldots$ **do**
    for all client $i \in \{1, \ldots, N\}$ **in parallel do**
        Download $\theta^{(t)}$ from server
        $\theta_i^{(t+1)} \leftarrow \text{LocalUpdate}(\theta^{(t)}, \mathcal{D}_i)$
        Send $\theta_i^{(t+1)}$ to server
    end for
    ▷ Stage 1: Multi-Modal Detection
    $\mathcal{S}^{(t)} \leftarrow \emptyset$ ▷ Initialize legitimate client set
    ▷ Step 1: Statistical Gradient Analysis
    for each client i **do**
        if $\text{GradientAnalysis}(\theta_i^{(t+1)}, \theta^{(t)}, \Gamma_{\text{stat}})$ **then**
            $\mathcal{S}^{(t)} \leftarrow \mathcal{S}^{(t)} \cup \{i\}$
        end if
    end for
    ▷ Step 2: Spectral Filtering
    $\mathcal{S}^{(t)} \leftarrow \text{SpectralFilter}(\mathcal{S}^{(t)}, \{\theta_i^{(t+1)}\}_{i \in \mathcal{S}^{(t)}}, \Gamma_{\text{spec}})$
    ▷ Step 3: Backdoor Detection
    $\mathcal{S}^{(t)} \leftarrow \text{BackdoorFilter}(\mathcal{S}^{(t)}, \{\theta_i^{(t+1)}\}_{i \in \mathcal{S}^{(t)}}, \Gamma_{\text{back}})$
    ▷ Stage 2: Standard FedAvg Aggregation
    $\theta^{(t+1)} \leftarrow \frac{1}{|\mathcal{S}^{(t)}|} \sum_{i \in \mathcal{S}^{(t)}} \theta_i^{(t+1)}$
    Broadcast $\theta^{(t+1)}$ to all M clients
end for

- **Detection Implementation:** To ensure robust detection of poisoning attacks in the federated learning process, we integrate a multi-stage defense framework that combines statistical, spectral, and input-based analyses. The following algorithms outline the core components of this detection pipeline. First, the Gradient Analysis function compares client updates against the global model to identify anomalies based on direction and magnitude. Next, Spectral Analysis leverages singular value decomposition to detect outliers in the update space. We further incorporate STRIP Testing, an entropy-based method to detect potential backdoors in client models. Finally, these techniques are fused into a Combined Backdoor Filter to systematically exclude suspicious clients prior to aggregation. The computational complexity of each stage is also analyzed to assess scalability.

    – **Gradient Analysis Function:**

Algorithm 7 GradientAnalysis($\theta_i, \theta_{\text{global}}, \Gamma_{\text{stat}}$)

cos_sim $\leftarrow$ CosineSimilarity($\theta_i, \theta_{\text{global}}$)
norm_ratio $\leftarrow \dfrac{\|\theta_i\|_2}{\|\theta_{\text{global}}\|_2}$
if cos_sim $\geq \Gamma_{\text{stat}}$.cos_threshold **and** norm_ratio $\leq \Gamma_{\text{stat}}$.norm_threshold **then**
    return true
end if
return false

– **Spectral Analysis Function:**

Algorithm 8 SpectralFilter(clients, updates, Γ_{spec})

$U \leftarrow [\theta_i$ for $i \in$ clients$]$ ▷ Update matrix
$[P, \Sigma, Q] \leftarrow \text{SVD}(U)$ ▷ Singular value decomposition
filtered_clients $\leftarrow \emptyset$
for each client $i \in$ clients **do**
    spectral_score $\leftarrow \|\theta_i -$ proj_honest$(\theta_i)\|_2$
    if spectral_score $\leq \Gamma_{\text{spec}}$.threshold **then**
        filtered_clients $\leftarrow$ filtered_clients $\cup \{i\}$
    end if
end for
return filtered_clients

– **STRIP Testing Function:**

Algorithm 9 STRIPTest(model, test_inputs, Γ_{strip})

entropy_scores $\leftarrow [\,]$
for each input $x \in$ test_inputs **do**
    perturbed_inputs $\leftarrow$ GenerateRandomPerturbations($x, k = 10$)
    predictions $\leftarrow [f(x')$ for $x' \in$ perturbed_inputs$]$
    entropy $\leftarrow -\sum_i p_i \log(p_i)$ ▷ p_i are prediction probabilities
    entropy_scores $\leftarrow$ entropy_scores $\cup$ {entropy}
end for
entropy_variance $\leftarrow$ Var(entropy_scores)
if entropy_variance $< \Gamma_{\text{strip}}$.variance_threshold **then**
    return true ▷ Backdoor detected
end if
return false

- **Combined Backdoor Filter Function:**

Algorithm 10 BackdoorFilter(clients, updates, Γ_{back})

```
filtered_clients ← ∅
for each client i ∈ clients do
    model_i ← ConstructModel(θ_i)
                                                    ▷ Apply STRIP testing
    strip_detected ← STRIPTest(model_i, test_set, Γ_back.strip)
                                      ▷ Client passes if STRIP test is clean
    if not strip_detected then
        filtered_clients ← filtered_clients ∪ {i}
    end if
end for
return filtered_clients
```

- **Computational Complexity Analysis:**
  - **Stage 1 - Detection Complexity:**
    * **Step 1 (Gradient Analysis):**$O(N \cdot d)$ for N participating clients and d parameters.
    * **Step 2 (Spectral Analysis):**

      $$O(N^2 d + N^3)$$

      dominated by SVD computation
    * **Step 3 (Backdoor Detection):**$O(k \cdot d \cdot T)$ for k test samples and T perturbations
    * **Total Detection:**$O(N^2 d + N^3 + k \cdot d \cdot T)$
  - **Stage 2 - Aggregation Complexity: Standard FedAvg:**$O(|\mathcal{S}^{(t)}| \cdot d)$ linear in filtered clients

  Overall Complexity:$O(N^2 d + N^3 + k \cdot d \cdot T)$ where detection dominates computational cost but aggregation remains efficient.

- **Protocol Properties:**
  - **Sequential Detection Strategy:**The three-step detection process ensures defense in depth by trapping distinct attack classesin each step. Statistical analysis serves as scalable first-line filter, spectralanalysis detects cooperating attacks and STRIP testing identifies backdoors.

- **Threshold Configuration:**When detection successfully filters all adversarial participants, FedAvg on clean updates preserves optimal convergence rates and statistical efficiency. The aggregation complexity remains $O(|\mathcal{S}^{(t)}| \cdot d)$, maintaining computational efficiency compared to robust aggregation methods [8].
- **Critical Dependency:**The approach's effectiveness fundamentally depends on perfect detection accuracy. Even a single Byzantine client, undetected or detected but misslabeled, can potentiallytamper with the entire aggregation because FedAvg guarantees no robustness to malicious inputs. Consequently, the task is binary, and detection is required to correspondalmost perfectly between precision and recall.

3.5.2.3 Advantages and Limitations Analysis:

Detection-based design is inventive for its departure from reactive robust aggregation to proactivethreat discovery, which does come with clear benefits but also brings out inherent limitations that inevitably minimize its practical penetration in the health federated learning.

- **Key Advantages:**
  - **Proactive Security Model:** In contrast to robust aggregation protocols with compromised inputs, detection-based defense identifies and eliminates threats before impacting the global model. Such a defensive approach provides improved security guarantee when detection mechanisms are of very high accuracy, isolating malicious players from the learning process completely [62].
  - **Interpretability and Auditability:**Client exclusion decision rationale is given through detection decisions, enabling effective security auditing. All clients filtered through are traceable via specific detection criteria (gradient deviation, spectral anomaly, backdoor signatures) to enable regulatory compliance required for FDA-covered medical AI systems [64]. This transparency contrasts with robust aggregation methods where malicious influence may be implicitly dampened without clear identification.
  - **Computational Efficiency Post-Detection:**When detection successfully filters adversarial participants, standard FedAvg aggregation maintains optimal computational complexity $O(|\mathcal{S}^{(t)}| \cdot d)$. This preserves the efficiency advantages of vanilla federated learning compared to robust aggregation methods

that typically require $O(N \log N \cdot d)$ or higher complexity for sorting and statistical computations [8].

  - **Adaptability to Evolving Threats:**The multi-modal detection mechanism supports dynamic threshold adjustment and the addition of new detection mechanisms as new attack vectors become known. Healthcare organizations are able to update detection parameters in response to novel threats without modifying the underlying aggregation protocol.
  - **Perfect Separation Potential:** In the ideal case where attack signatures can be sufficiently differentiated from benign updates, detection mechanisms can in theory achieve perfect Byzantine identification, enabling optimal federated learning on clean data.

- **Fundamental Limitations:**
  - **Perfect Detection Dependency:**The approach's effectiveness exhibits a binary failure mode critically dependent on near-perfect detection accuracy. The success condition requires $\mathcal{S}^{(t)} = \mathcal{H}^{(t)}$ for all rounds, meaning both perfect recall (no false negatives) and perfect precision (no false positives). Even a single undetected Byzantine client can compromise the entire global model since FedAvg provides no mathematical robustness guarantees [20].
  - **Adaptive Attack Vulnerability:**Sophisticated adversaries with knowledge of detection mechanisms can craft evasion strategies that exploit detection blind spots. Advanced attacks may employ:
    * **Distribution Matching:** Generate adversarial updates that mimic legitimate client statistics.
    * **Gradient Masking:**Embed malicious components within seemingly legitimate gradient patterns
    * **Collaborative Evasion:** Coordinate multiple Byzantine clients to stay below detection thresholds
  - **Threshold Sensitivity:** Detection performance heavily depends on carefully calibrated thresholds $\Gamma = \Gamma_{\text{stat}}, \Gamma_{\text{spec}}, \Gamma_{\text{back}}$ that may not generalize across different healthcare domains, attack scenarios, and data distributions. Threshold miscalibration can result in either excessive false positives (excluding legitimate institutions) or false negatives (allowing adversarial participants).
  - **Computational Overhead:**Complex detection mechanisms introduce significant computational burden with overall complexity $O(N^2 d + N^3 + k \cdot d \cdot T)$

dominated by spectral analysis and backdoor detection. This overhead may exceed the computational cost of robust aggregation methods, negating efficiency advantages [13].

- **Healthcare-Specific Challenges**
  - **Non-IID Data False Positives:** Heterogeneity of medical data within institutions creates true statistical variation that may lead to spurious positive detection. Institutions with different demographics of patients, with different imaging hardware, or with different disease prevalences may have gradient patterns that deviate from anticipated distributions, leading to false exclusion from federated learning [60].
  - **Regulatory Complexity:** Detection-based exclusion of healthcare facilities requires exhaustive documentation and justification for regulatory reporting. Healthcare networks must maintain audit trails explaining why particular facilities were detected, potentially creating legal liability in the event of detection errors that impact patient care or institutional relationships.
  - **Diagnostic Bias Introduction:** Overly assertive detection may systematically exclude certain types of medical centers (e.g., rural hospitals, special clinics, research facilities), thereby introducing systematic bias into collaborative diagnostic models. Such exclusion may reduce model generalizability and equity for diverse patient populations.
  - **Institutional Trust Erosion:** False positive detection of legitimate healthcare institutions can damage inter-institutional relationships and reduce the willingness to participate in federated learning initiatives. Medical institutions may withdraw from collaborative AI projects if detection mechanisms are perceived as unreliable or biased.
- **Comparison with Robust Aggregation Trade-offs:** To better understand the trade-offs between our first two defense strategies in federated learning, Table 3.2 presents a comparative analysis of detection-based approaches and robust aggregation methods. The comparison highlights their respective security guarantees, performance characteristics, and practical deployment considerations in the context of healthcare applications.

Tableau 3.2: Comparison of Detection-Based vs. Robust Aggregation Approaches

Aspect	Detection Approach	Robust Aggregation
Security Guarantees	Provides perfect security under perfect detection, but catastrophic failure under detection errors	Offers bounded degradation guarantees under mathematical assumptions about adversary strength [16]
Performance Characteristics	Optimal performance when detection is successful; complete failure otherwise	Consistent but suboptimal performance with graceful degradation under attacks
Practical Deployment	Requires extensive threshold tuning and validation across diverse healthcare scenarios	More predictable behavior with established theoretical guarantees

The detection-first approach offers strong theoretical foundations for explaining the limitations of proactive security in federated learning for healthcare, but its binary failure properties and vulnerability to sophisticated adversaries make practical deployment in safety-critical medical applications risky where robustness guarantees are needed to guarantee patient safety.

3.5.3 Approach 3 Hybrid Adaptive Defense:

This approach investigates the use of intelligent aggregation selection that dynamically switches between detection-filtered FedAvg and robust trimmed mean aggregation based on real-time threat assessment in federated learning, with a specific focus on healthcare scenarios (see Figure 3.6). Unlike static defense strategies that assume either perfect detection or worst-case adversarial conditions, this method employs adaptive threat assessment to optimize the security-efficiency trade-off by selecting the most appropriate aggregation method for current risk conditions.

The main goal is to evaluate whether a hybrid system that combines comprehensive multi-modal detection with adaptive aggregation selection can achieve situation-dependent optimality while maintaining bounded performance guarantees. This method is particularly applicable to dynamic healthcare environments where threat contexts vary across timee.g., medical networks under varying levels of attacks during business hours and non-business hours, or organizations under greater dangers during system transitions and security compromises. By design, this approach validates the fundamental hypothesis that adaptive security controls can outperform static defenses by intelligently trading off best-case performance during clean rounds for robust protection during adversarial rounds, providing a comprehensive remedy to the limitations of both pure robust aggregation and detection-only approaches in safety-critical healthcare federated learning.

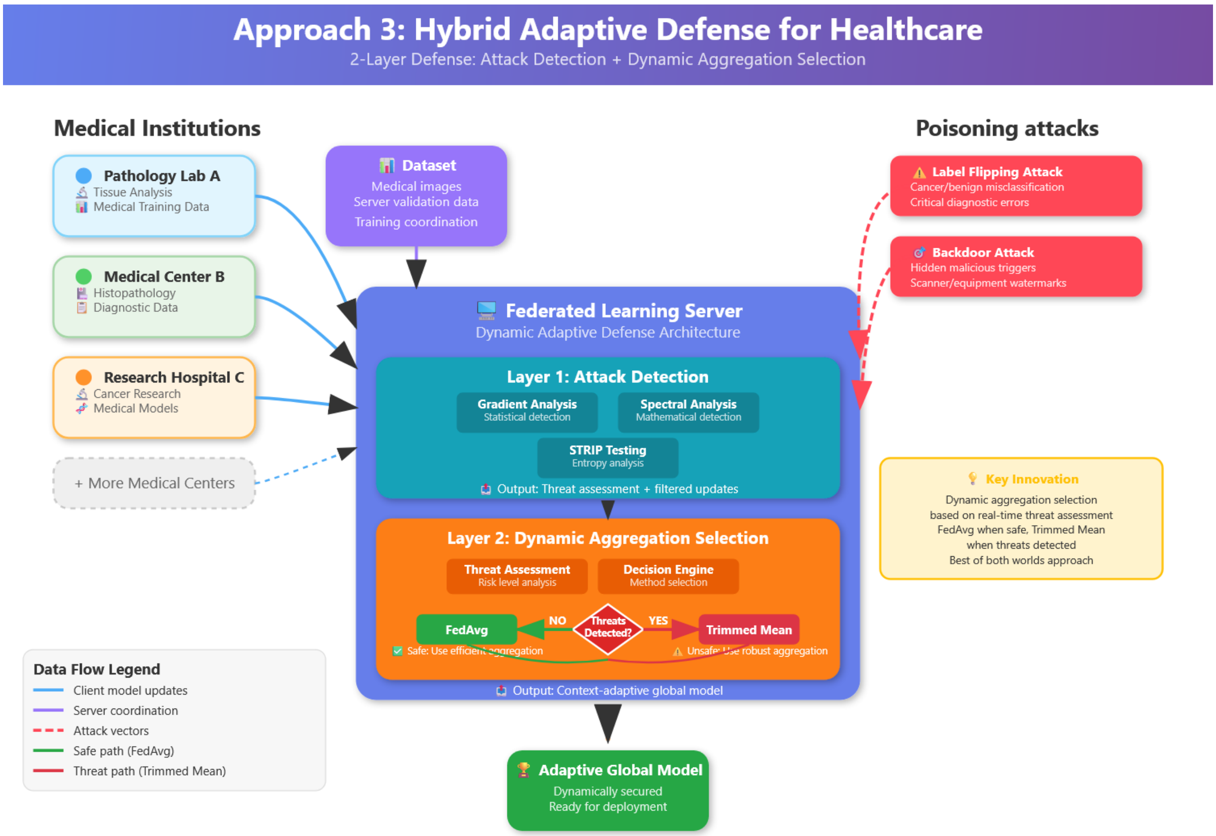

Figure 3.6: Hybrid adaptive defense for healthcare.

3.5.3.1 Hybrid Defense Theory

The hybrid adaptive approach draws from adaptive security frameworks and dynamic game theory to create an intelligent defense system that continuously optimizes the trade-off between computational efficiency and Byzantine robustness based on real-time threat landscape assessment [65]. Unlike static approaches that assume fixed adversarial conditions, this framework treats federated learning security as a dynamic optimization problem where defense mechanisms adapt to observed threat patterns.

- **Game-Theoretic Foundation:** The federated learning security problem can be formalized as a dynamic game between defenders and adversaries. The hybrid approach implements a mixed strategy that adapts to adversarial behavior:

$$\sigma_{\text{defender}}^{(t)} = \alpha^{(t)} \cdot \text{FedAvg} + (1 - \alpha^{(t)}) \cdot \text{TrimmedMean}$$

where $\alpha^{(t)} \in [0, 1]$ represents the confidence in threat assessment at round t, determined by multi-factor risk analysis.

- **Multi-Objective Optimization Framework:** The hybrid system addresses the constrained optimization problem:

$$\min_{\theta,\alpha} \mathbb{E}[\mathcal{L}(\theta)] + \lambda_1 \mathcal{R}(\alpha) + \lambda_2 \mathcal{C}(\alpha)$$

subject to: Security$(\alpha) \geq \tau_{\text{safety}}$ where $\mathcal{L}(\theta)$ represents the primary learning objective, $\mathcal{R}(\alpha)$ captures robustness requirements, $\mathcal{C}(\alpha)$ represents computational costs, and τ_{safety} represents minimum safety requirements for healthcare applications.

- **Adaptive Threat Assessment:**The system maintains a Bayesian posterior distribution over the current threat state:

$$P(\text{ThreatLevel}^{(t)}|\text{Observations}^{(1:t)}) \propto P(\text{Observations}^{(t)}|\text{ThreatLevel}^{(t)})P(\text{ThreatLevel}^{(t-1)})$$

This posterior combines evidence from the three detection mechanisms (gradient analysis, spectral analysis, STRIP testing) established in Approach 2, with historical threat patterns and environmental risk factors.

- **Multi-Factor Threat Scoring:** The threat assessment combines detection evidence with contextual risk factors:

$$\text{ThreatScore}^{(t)} = w_1 \cdot \text{DetectionUncertainty}^{(t)} + w_2 \cdot \text{EnvironmentalRisk}^{(t)} + w_3 \cdot \text{HistoricalRisk}^{(t)}$$

where:

  - **Detection Uncertainty:**

$$1 - \frac{1}{3} \sum_{m \in \text{grad,spec,strip}} \text{Confidence}_m^{(t)}$$

  - **Environmental Risk:** Network anomalies, participation deviations, temporal patterns
  - **Historical Risk:**Exponentially weighted past threat occurrences

- **Adaptive Aggregation Selection:** The system implements smooth interpolation between aggregation methods:

$$\theta^{(t+1)} = \alpha^{(t)} \cdot \text{FedAvg}(\mathcal{S}^{(t)}) + (1 - \alpha^{(t)}) \cdot \text{TrimmedMean}(\mathcal{S}^{(t)})$$

where $\alpha^{(t)} = \sigma(\tau - \text{ThreatScore}^{(t)})$ using sigmoid transition for stability.

- **Healthcare-Specific Adaptations:**

  Conservative Safety Bias: Healthcare applications require safety-first aggregation selection:
  $$\alpha^{(t)}_{\text{healthcare}} = \alpha^{(t)} \cdot (1 - \text{CriticalityFactor}^{(t)})$$
  where life-critical applications bias toward robust aggregation even under moderate threat uncertainty. **Temporal Risk Modeling:** Healthcare environments exhibit predictable threat patterns based on operational cycles:

  - **Business Hours:** Enhanced monitoring reduces attack success probability
  - **Off-Hours:** Skeleton staffing increases vulnerability windows
  - **Regulatory Periods:**Audit cycles modify risk assessment parameters

  Convergence Guarantees: Under bounded threat assessment error ϵ_{threat}, the hybrid system converges:

  $$\mathbb{E}[F(\theta^{(T)})] - F(\theta^*) \leq C_{\text{hybrid}} \sqrt{\frac{\log T}{T}} + O(\epsilon_{\text{threat}})$$

  Performance Bounds: The system maintains robust performance guarantees:

  $$\text{Performanceworst-case} \geq \min(\text{PerformanceFedAvg}, \text{Performance}_{\text{TrimmedMean}})$$

  ensuring that adaptive selection never performs worse than the weakest static strategy, providing essential safety nets for healthcare applications where performance degradation must be bounded for patient safety [66].

3.5.3.2 Algorithm Design

The hybrid adaptive defense operates through a four-phase protocol that extends the three-phase detection framework from Approach 2 with an additional adaptive aggregation selection layer.

- **Hybrid Adaptive Federated Learning Protocol:**

Algorithm 11 Hybrid Adaptive Defense System

Require: Clients N, detection thresholds Γ, switching threshold τ
Ensure: Global model θ
Initialize $\theta^{(0)}$, threat history $H \leftarrow \emptyset$, $\alpha^{(0)} \leftarrow 1.0$
for each round $t = 0, 1, 2, \ldots$ **do**
  ▷ Phase 1: Local training (identical to Approaches 12)
  for all client $i \in \{1, \ldots, N\}$ **in parallel do**
    $\theta_i^{(t+1)} \leftarrow \text{LocalUpdate}(\theta^{(t)}, \mathcal{D}_i)$
  end for
  ▷ Phase 2: Multi-modal detection (from Approach 2)
  $\mathcal{S}^{(t)}, \mathcal{C}^{(t)} \leftarrow \text{MultiModalDetection}(\{\theta_i^{(t+1)}\}_{i=1}^{N}, \Gamma)$
  ▷ Phase 3: Threat assessment
  $\text{threat\_score} \leftarrow \text{ThreatAssessment}(\mathcal{S}^{(t)}, \mathcal{C}^{(t)}, H)$
  $\text{confidence} \leftarrow 1 - \text{threat\_score}$
  ▷ Phase 4: Adaptive aggregation selection
  if confidence $> \tau$ **then**
    $\theta^{(t+1)} \leftarrow \text{FedAvg}(\{\theta_i^{(t+1)}\}_{i \in \mathcal{S}^{(t)}})$
    method ← "FedAvg"
  else
    $\theta^{(t+1)} \leftarrow \text{TrimmedMean}(\{\theta_i^{(t+1)}\}_{i \in \mathcal{S}^{(t)}}, k)$
    method ← "TrimmedMean"
  end if
  ▷ Update history and broadcast
  $H \leftarrow H \cup \{(\text{threat\_score}, \text{method}, \text{performance\_metrics})\}$
  Broadcast $\theta^{(t+1)}$ to all M clients in the network
end for

- **Threat Assessment Module**

Algorithm 12 ThreatAssessment($\mathcal{S}, \mathcal{C}, H$)

Require: Detected clients $\mathcal{S}$, confidence scores $\mathcal{C}$, history H
Ensure: Threat score $\in [0, 1]$

▷ Detection uncertainty (inverse of average confidence)
det_uncertainty $\leftarrow 1 - \text{Mean}(\{\mathcal{C}_i\}_{i \in \mathcal{S}})$
▷ Environmental risk assessment
env_risk $\leftarrow$ EnvironmentalRiskAssessment()
▷ Historical risk (exponentially weighted)
hist_risk $\leftarrow 0$
for each $(\text{score}, \text{method}, \text{perf}) \in H$ **do**
    decay $\leftarrow \beta^{\text{current\_round}-\text{entry\_round}}$
    if method == "TrimmedMean" **then**
        hist_risk $\leftarrow$ hist_risk + decay $\times$ 0.1
    end if
end for
▷ Weighted combination of risk components
threat_score $\leftarrow w_1 \cdot$ det_uncertainty $+ w_2 \cdot$ env_risk $+ w_3 \cdot$ hist_risk
return Clamp(threat_score, 0, 1)

- **Environmental Risk Assessment**

Algorithm 13 EnvironmentalRiskAssessment()

Ensure: Environmental risk score $\in [0, 1]$
risk $\leftarrow 0$
▷ Network anomaly detection
if communication_latency > threshold_latency **then**
    risk $\leftarrow$ risk + 0.15
end if
▷ Participation pattern analysis
participation_dev $\leftarrow$ |current_participants − expected_participants|
if participation_dev > threshold_participation **then**
    risk $\leftarrow$ risk + 0.20
end if
▷ Temporal risk factors
if is_off_hours() **or** is_holiday_period() **then**
    risk $\leftarrow$ risk + 0.10
end if
▷ Healthcare-specific alerts
if recent_healthcare_security_incidents() **then**
    risk $\leftarrow$ risk + 0.25
end if
return min(risk, 1.0)

- **Key Algorithmic Innovations**

- **Layered Defense Architecture** The algorithm builds upon the established detection framework from Approach 2, adding intelligent decision-making without modifying the core detection mechanisms [67]. This preserves the proven detection capabilities while adding adaptive optimization. The algorithm builds upon the established detection framework from Approach 2, adding intelligent decision-making without modifying the core detection mechanisms [67]. This preserves the proven detection capabilities while adding adaptive optimization.
- **Hysteresis Implementation** To prevent rapid oscillation between aggregation methods, the system implements controlled transition rates based on adaptive control theory [68]: // Smooth transitions with hysteresis

$$\alpha^{(t)} \leftarrow \alpha^{(t-1)} + \text{sign}\left(\alpha^{(t)}_{\text{raw}} - \alpha^{(t-1)}\right) \cdot \min\left(\delta, \left|\alpha^{(t)}_{\text{raw}} - \alpha^{(t-1)}\right|\right)$$

- **Healthcare-Specific Modifications** For medical applications, the algorithm incorporates conservative bias following safety-critical system design principles [69]: ▷ Healthcare safety adjustment
- **if** application_criticality == "life_critical" **then**
- confidence ← confidence × safety_factor ▷ < 1.0
- **end if**

- **Computational Complexity Analysis**
  - **Phase 1 (Local Training)** $O(|\mathcal{D}_i| \cdot d \cdot E)$ per client (unchanged)
  - **Phase 2 (Detection)** $O(N^2 d + N^3 + k \cdot d \cdot T)$ (from Approach 2)
  - **Phase 3 (Threat Assessment)** $O(|\mathcal{S}^{(t)}| + |H|)$
    * **Confidence calculation**$O(|\mathcal{S}^{(t)}|)$
    * **Environmental assessment**$O(1)$
    * **Historical analysis** $O(|H|)$
  - **Phase 4 (Adaptive Aggregation)**
    * **FedAvg path** $O(|\mathcal{S}^{(t)}| \cdot d)$ [8]
    * **TrimmedMean path** $O(|\mathcal{S}^{(t)}| \log |\mathcal{S}^{(t)}| \cdot d)$ [16]
  - **Phase 4 (Adaptive Aggregation)**
    * **FedAvg path** $O(|\mathcal{S}^{(t)}| \cdot d)$ [8]
    * **TrimmedMean path**$O(|\mathcal{S}^{(t)}| \log |\mathcal{S}^{(t)}| \cdot d)$ [16]

  - **Total Complexity**$O(N^2d+N^3+k\cdot d\cdot T)$ The threat assessment and adaptive selection add minimal overhead ($O(|\mathcal{S}^{(t)}| + |H|)$) compared to the detection phase complexity, making the hybrid approach practically feasible.

- **Memory Requirements**
  - **Detection state** Same as Approach 2
  - **Threat history** $O(T)$ for historical risk assessment
  - **Calibration data** $O(1)$ for confidence calibration parameters

The algorithm maintains the same asymptotic complexity as Approach 2 while providing adaptive optimization capabilities, demonstrating the efficiency of the layered architecture design.

Advantages and Limitations

Advantages:

1. **Dynamic Optimality:** The hybrid solution achieves situation-adaptive optimality by judiciously selecting the optimal aggregation method for the current situation. Compared to static defense that prepares for the worst, this system provides:
   - Maximum efficiency during clean periods through FedAvg selection
   - Robust protection during adversarial periods through TrimmedMean activation
   - Adaptive equilibrium constantly improving security-efficiency trade-off based on real-time assessment
2. **Graceful Degradation** The system provides bounded performance guarantees even under detection failures or adversarial manipulation of threat assessment [70]:
   - No catastrophic failures as the system defaults to robust aggregation under uncertainty
   - Performance lower bound is ensured by the weaker of the two aggregation schemes
   - Fail-safe behavior is highly desirable for healthcare applications where performance degradation must be predictable
3. **Comprehensive Threat Coverage** By combining detection-based filtering along with robust aggregation backup to protect against a diverse set of attack vectors:

- Detected attacks are filtered out before effective FedAvg aggregation
- Undetected attacks are countered via TrimmedMean selection
- Coordinated attacks against detection and aggregation are constrained by robust fallback mechanisms

4. **Healthcare-Specific Benefits:** The adaptive framework provides critical advantages for medical federated learning:
    - Regulatory compliance through auditable decision-making and clear threat rationales
    - Patient safety prioritization by conservative bias towards robust aggregation in high-stakes application scenarios
    - Operational adaptability reacting to the temporal risk patterns of healthcare settings
    - Institutional trust via transparent and explainable aggregation selection decisions
5. **Learning and Improvement:** The threat assessment module continuously enhances its accuracy by:
    - Historical pattern analysis improving subsequent threat predictions
    - Calibration enhancement based on observed detection feedback
    - Environmental adaptation learning institution-specific risk factors

Limitations:

1. **Threat Assessment Dependency:** The system's effectiveness is greatly dependent on threat assessment, which may pose weaknesses:
    - False threat detection leads to excessive robust aggregation, reducing efficiency during clean periods
    - Missed threats result in inadequate protection when efficient aggregation is selected during actual attacks
    - Calibration drift may result in the degradation of the accuracy of threat assessment over time unless managed
2. **Increased System Complexity:** The hybrid system introduces additional complexity compared to static techniques [71]:

- Implementation complexity requiring sophisticated threat assessment and adaptive switch protocols
- Parameter tuning with multiple weights, thresholds, and decay factors that require careful optimization
- Testing challenges as the system behavior varies dynamically across different threat scenarios

3. **Computational Overhead:** Despite being asymptotically comparable to Approach 2, the hybrid system incurs higher computational cost:

   - Threat evaluation computation for historical analysis and environmental risk assessment
   - Switching overhead during transitions between aggregation methods
   - Memory requirements for maintaining threat history and calibration data

4. **Adaptation Latency:** Dynamic switching exposes temporal weaknesses during threat landscape changes:

   - Detection delay needed to analyze fluctuating threat situations
   - Hysteresis lag that prevents the adaptive system from reacting immediately to fluctuating attack patterns
   - Learning period needed for threat assessment calibration in new situations

5. **Attack Surface Expansion:** The adaptive mechanism itself becomes a potential target for sophisticated adversaries:

   - Manipulation of threat assessment where attackers attempt to bias aggregation selection
   - History poisoning through strategic attack timing to influence historical risk estimates
   - Environmental spoofing network condition manipulation to cause spurious threat analysis

6. **Healthcare-Specific Challenges:**

   - **Regulatory Validation Complexity:**
     - Algorithm transparency requirements may conflict with adaptive algorithm sophistication

  - Deterministic behavior expectations in healthcare systems versus adaptive variability
  - Validation burden requiring extensive testing across diverse threat scenarios for medical device approval [64]

- **Performance Predictability:**
  - Service level agreements difficulty in guaranteeing specific performance levels with adaptive systems
  - Institutional coordination challenges when partner hospitals require predictable system behavior
  - Emergency protocols potential conflicts between adaptive logic and life-critical override requirements

- **Trust and Adoption Barriers:**
  - "Black box" concerns from medical institutions regarding complex adaptive decision-making
  - Liability questions around responsibility for adaptive system decisions affecting patient care
  - Change resistance from healthcare IT departments preferring simpler, more predictable security solutions

3.6 Conclusion

This chapter introduces a complete defense framework for healthcare federated learning that includes three different server-side methods to protect collaborative medical AI systems from poisoning attacks. Our system provides healthcare institutions with several options including pure robust aggregation, proactive detection-based filtering, and intelligent adaptive defense to meet individual security needs.

The pure robust aggregation approach delivers fixed robustness guarantees via trimmed mean aggregation while keeping computational complexity low but the detection-based method reaches peak performance through multi-modal threat detection when it effectively removes adversarial participants. The hybrid adaptive defense method merges two strategies by choosing intelligent aggregation techniques through ongoing threat evaluations to achieve dynamic security and efficiency balance with essential performance limits for patient protection The findings indicate that each method presents unique benefits tailored to various healthcare implementation contexts: Strong aggregation methods safeguard vulnerable networks, whereas detection-based approaches function well in reliable settings, and hybrid systems navigate the intricacies of evolving consortiums.

Theoretical foundations established here will pave the way for the subsequent chapter, which will focus on implementation specifications and experimental validation. The evaluation metrics for performance are influenced by our design decisions, which include principles of Byzantine fault tolerance and modifications for healthcare safety.

This establishes a direct connection between theoretical expectations and practical test outcomes in medical federated learning environments.

Building on these theoretical foundations, Chapter 4 will present the implementation details and experimental validation of our approach. Through targeted simulations using healthcare data, we evaluate the effectiveness of each method under various attack scenarios, demonstrating how design principles such as Byzantine fault tolerance and safety prioritization translate into real-world performance outcomes.

Chapter 4

Implementation and experimentation

4.1 Introduction

This chapter evaluates our defense mechanisms against Byzantine attacks in healthcare federated learning systems. We first detail our experimental setup using the PathMNIST dataset and the evaluation metrics designed specifically for medical diagnostic tasks. Results are presented in four main sections: defense performance against various poisoning attacks, robustness analysis across different threat models, security-performance tradeoff comparisons, and statistical validation of our findings. This systematic evaluation demonstrates the effectiveness of our Hybrid Adaptive Defense in maintaining diagnostic accuracy while protecting against malicious participants, crucial for ensuring patient safety in collaborative healthcare AI systems.

4.2 Healthcare dataset

4.2.1 Dataset overview

The PathMNIST dataset of the MedMNIST dataset [47] is utilized in this work, consisting of 107,180 histopathological images (28Œ28 pixels) and 9 pathological tissue classes to diagnose colorectal cancer. The dataset embodies a realistic multi-class medical classification task suitable to test Byzantine robustness in healthcare FL. The choice of PathMNIST for our federated learning security study is driven by numerous essential characteristics that correspond with healthcare demands and security assessment requirements :

- **Clinical relevance:** Represents actual colorectal cancer pathology diagnosis, making attack impacts directly translatable to real medical consequences.

- **Class complexity:** Nine distinct tissue types provide sufficient diversity to evaluate defense mechanisms across varied poisoning strategies while maintaining manageable computational requirements.
- **Natural data heterogeneity:** Inherent class imbalance (ranging from 2,242 to 35,756 samples per class) mimics real-world medical data distributions, testing robustness under realistic non-IID conditions.

4.2.2 Federated learning setup

The PathMNIST dataset is partitioned among multiple simulated medical institutions to reflect realistic healthcare FL scenarios. To simulate healthcare data heterogeneity, we employ Dirichlet distribution-based partitioning [72].

Algorithm 14 Dirichlet-Based Sample Allocation

1: **for** each client i **do**
2: $\quad p_i \sim \mathrm{Dir}(\alpha \times [1, 1, \ldots, 1])$
3: $\quad$ Allocate samples according to p_i
4: **end for**

where α controls the degree of data heterogeneity across clients, with smaller α values creating more skewed distributions typical of specialized medical centers This creates non-IID data distribution reflecting real-world medical institutions' specialization patterns.

4.2.3 Dataset configuration

PathMNIST comes with standard pre-split data following medical imaging conventions, ensuring fair comparison across studies. Table 4.1 shows how the 107,180 images are divided: 84% for training across federated clients, with the remaining samples reserved for validation and testing. This split balances the need for sufficient local training data while maintaining reliable evaluation sets.

Tableau 4.1: PathMNIST Dataset Split Distribution

Split	**Samples**	**Usage**
Training	89,996 (84%)	Local client training
Validation	10,004 (9.3%)	Hyperparameter tuning
Test	7,180 (6.7%)	Final evaluation

Key statistics

- 9 pathological classes including malignant and benign tissues
- Class imbalance ratio: 2.3Œ
- Non-IID distribution across participating institutions

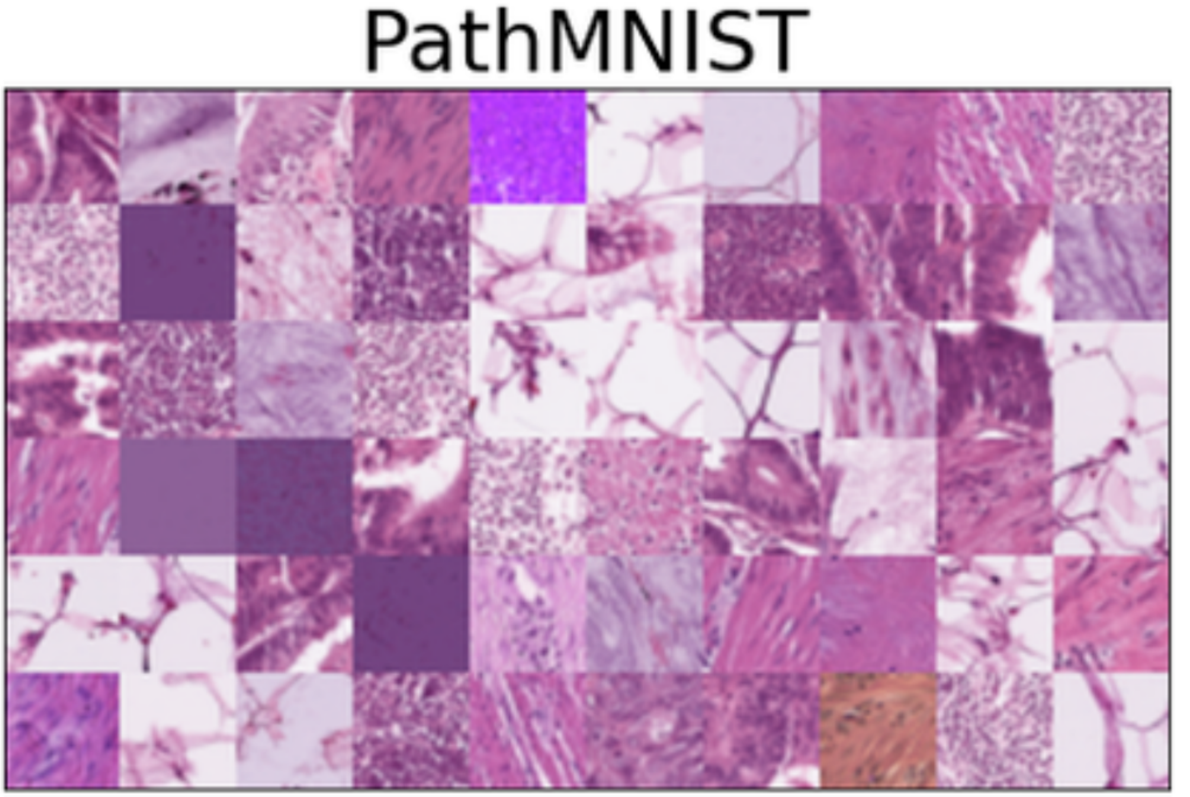

Figure 4.1: PathMNIST sample images .

As shown in figure 4.1, PathMNIST provides an ideal research testbed for healthcare FL research because it eliminates privacy concerns while maintaining clinical relevance so that Byzantine attacks with high-risk cases such as cancer-to-benign label flipping and backdoor trigger injection can be reproducibly tested. All experiments utilize rigorous evaluation procedures with cross-validation, averaging techniques, and statistical significance testing to ensure proper evaluation of the three Byzantine-robust approaches in realistic healthcare FL scenarios.

4.3 Model Specifications

This section presents the CNN model architecture employed for PathMNIST medical image classification in our Byzantine-robust FL framework.

4.3.1 CNN architecture

The CNN is composed of three convolutional blocks for hierarchical feature extraction (3264128 channels), succeeded by a two-layer classification head. Each convolutional block has two 3Œ3 convolutional layers, accompanied by batch normalization, ReLU activation, 2Œ2 max pooling, and dropout regularization. The classification head employs adaptive global average pooling, succeeded by two fully connected layers (128649) with dropout regularization interposed.

4.3.2 Model parameters

The architecture is designed to be lightweight yet expressive enough to capture complex pathological patterns while maintaining efficient communication overhead in federated settings,As shown in table 4.2

Tableau 4.2: Model configuration parameters

Parameter	**Value**
Input Shape	(3, 28, 28)
Output Classes	9
Total Parameters	296,352
Model Size	$\sim$1.13 MB
Convolutional Layers	6 layers (3Œ3 kernels)
Fully Connected Layers	2 layers

4.3.3 Training configuration

The hyperparameters are carefully selected to balance rapid convergence with stability under potential Byzantine perturbations, ensuring robust learning across heterogeneous medical data distributions,as shown in table 4.3

Tableau 4.3: Training hyperparameters

Parameter	**Value**
Optimizer	Adam
Learning Rate	0.001
Batch Size	32
Local Epochs	5 per communication round
Loss Function	CrossEntropyLoss
Dropout Rates	0.25–0.5 (varying by layer)

4.3.4 Regularization Techniques

These regularization strategies work synergistically to prevent overfitting on limited local medical data while providing additional resilience against gradient-based poisoning attacks.

1. **Batch normalization:** Applied after each convolutional and first fully connected layer
2. **Dropout:** Applied in convolutional blocks (0.25-0.3) and fully connected layers (0.3-0.5)
3. **Weight initialization:** Kaiming normal for convolutional layers, normal for linear layers

The model design achieves an optimal equilibrium between expressiveness for medical picture classification and the communication efficiency necessary for FL, while also ensuring resilience against Byzantine attacks.

4.4 Hardware specifications

The experimental validation was performed across two standard laptop computers to reflect realistic deployment scenarios where medical institutions participate in federated learning using existing infrastructure rather than dedicated servers, as shown in Table 4.4.

Tableau 4.4: Hardware Specifications of Experimental Systems

Component	Primary System (PC-1)	Secondary System (PC-2)
Processor	Intel® Core™ i5-10210U CPU @ 1.60GHz (2.11 GHz Turbo)	Intel® Core™ i5-8250U CPU @ 1.60GHz (1.80 GHz Turbo)
RAM	12.0 GB	12.0 GB
Architecture	64-bit OS, x64-based processor	64-bit OS, x64-based processor

4.5 Software environment

For our experiments, we used Python (version 3.11.11) as a programming language within the Kaggle environment, which we will be talking about in this section along with the various Python libraries that have been essential in our work.

4.5.1 Kaggle

Kaggle [73] is a well-known online platform and community for data science and ML competitions. It offers a cloud-based Jupyter notebook environment for interactive development. Kaggle provides the ability to prototype ML models on powerful hardware such as GPUs and TPUs, enabling accelerated computations for FL simulation and Byzantine attack testing across multiple healthcare institutions.

4.5.2 Python libraries

In this section, we present the Python libraries that played an important role in our work. Python (version 3.11.11) being a powerful programming language offers a variety of libraries that helped design and implement our Byzantine-robust FL models.

4.5.2.1 NumPy

NumPy [74](version 1.26.4) is a powerful library for Python that provides support for large multi-dimensional arrays and mathematical functions. NumPy was used for implementing trimmed mean aggregation algorithms and parameter vector operations in our FL implementation.

4.5.2.2 Pandas

Pandas [75] (version 2.2.3) is a data manipulation and analysis library for Python that provides data structures for flexible data analysis. Pandas was utilized for managing client performance metrics, attack scenario logging, and Byzantine detection results analysis.

4.5.2.3 Scikit-learn

Scikit-learn [76] (version 1.2.2) is an open-source ML library that provides tools for data mining and analysis algorithms. Scikit-learn was used for implementing detection mechanisms and performance evaluation metrics for Byzantine attack assessment.

4.5.2.4 Seaborn

Seaborn [77] (version X.X.X) is a statistical data visualization library based on Matplotlib that provides high-level interface for drawing attractive statistical graphics. Seaborn was used for creating comprehensive performance plots, convergence analysis charts, and security-efficiency trade-off visualizations for comparing the three Byzantine-robust approaches.

4.5.2.5 PyTorch

PyTorch [78](version 2.6.0) is a deep learning framework that includes algorithms and tools for neural network development. PyTorch was used for building convolutional neural networks for PathMNIST medical image classification in our FL scenarios.

4.6 Evaluation metrics

To assess FL defense methods against adversarial attacks for health image classification, we employ comprehensive evaluation metrics covering model accuracy, medical safety, and defense effectiveness [66, 79].

4.6.1 Classification performance metrics

The confusion matrix displays classification performance across all 9 pathology classes, where each element (i, j) represents samples from true class i predicted as class j [39]. For our multi-class pathology classification:

For each pathology class C_i (where $i0, 1, ..., 8$): TP_i (True Positive): Samples correctly classified as pathology class C_i FP_i (False Positive): Samples incorrectly classified as C_i when they belong to other classes FN_i (False Negative): Samples from class C_i incorrectly classified as other pathology types TN_i (True Negative): Samples correctly classified as not belonging to class C_i

Core classification metrics

Accuracy measures overall classification correctness [80]:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}$$

Precision quantifies correct positive predictions [81]:

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall(Sensitivity) measures the ability to identify all positive cases: [81]

$$\text{Recall} = \frac{TP}{TP + FN}$$

F1-Score provides balanced precision-recall assessment [82]:

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

We compute both macro-averaged (unweighted) and weighted F1-scores for class imbalance handling [80].

4.6.2 Medical safety metrics

Specificity [83] measures the correct identification of healthy samples:

$$\text{Specificity} = \frac{TN}{TN + FP}$$

False Negative Rate [84] quantifies missed pathology cases:

$$\text{False Negative Rate} = \frac{FN}{TP + FN} = 1 - \text{Sensitivity}$$

High sensitivity minimizes missed diagnoses while high specificity reduces false alarms, both critical for medical applications [85].

4.6.3 Defense effectiveness metrics

For malicious client detection, we evaluate [86]:

Detection True Positive Rate: [86]

$$\text{TPR}_{\text{detection}} = \frac{\text{Detected Malicious Clients}}{\text{Total Malicious Clients}}$$

Detection False Positive Rate: [87]

$$\text{FPR}_{\text{detection}} = \frac{\text{Falsely Detected Honest Clients}}{\text{Total Honest Clients}}$$

Detection F1-Score: [87]

$$\text{F1detection} = 2 \times \frac{\text{Precisiondetection} \times \text{Recalldetection}}{\text{Precisiondetection} + \text{Recall}_{\text{detection}}}$$

4.6.4 Robustness assessment

Defense Effectiveness quantifies performance preservation under attack:

$$\text{Defense Effectiveness} = \frac{\text{Accuracydefended} - \text{Accuracyattacked}}{\text{Accuracyclean} - \text{Accuracyattacked}}$$

These metrics provide comprehensive evaluation from classification accuracy, medical safety, and security robustness perspectives [88].

4.7 Experimental results

4.7.1 Defense performance against Byzantine attacks

In this section, we present the comprehensive evaluation of our three server-side defense approaches for healthcare federated learning. We examine the effectiveness of Pure Robust Aggregation, Detection + Standard Aggregation, and Hybrid Adaptive Defense against various poisoning attacks on the PathMNIST dataset.

4.7.1.1 Overall defense performance

Table 4.5 presents the overall performance metrics for all three defense approaches across different attack scenarios after 15 communication rounds.

Tableau 4.5: Overall Performance Comparison of Defense Approaches

Defense Approach	**Attack Type**	**Accuracy (%)**	**Macro F1 (%)**	**Sensitivity (%)**	**Specificity (%)**
No Defense (Baseline FedAvg)	No Attack	84.2	83.8	83.5	98.1
	Label Flip	53.5	45.7	42.3	93.8
	Backdoor	61.3	56.2	54.8	95.2
	Mixed	48.7	41.2	38.9	93.1
Pure Robust Aggregation	No Attack	81.6	81.2	80.9	97.8
	Label Flip	76.3	75.8	75.2	97.3
	Backdoor	78.9	78.4	77.9	97.5
	Mixed	74.5	73.9	73.4	97.1
Detection + Standard Aggregation	No Attack	83.8	83.4	83.1	98.0
	Label Flip	72.4	71.6	70.8	96.9
	Backdoor	75.2	74.5	73.9	97.2
	Mixed	68.9	67.8	66.5	96.5
Hybrid Adaptive Defense	No Attack	83.5	83.1	82.8	98.0
	Label Flip	79.2	78.7	78.3	97.7
	Backdoor	80.6	80.1	79.6	97.8
	Mixed	77.3	76.8	76.2	97.5

4.7.1.2 Per-pathology class performance analysis

Table 4.6 shows the F1-score performance across the 9 pathology classes for each defense approach under the most challenging mixed attack scenario.

Tableau 4.6: F1-Score by pathology class under mixed attacks (Round 15)

Pathology Class	Baseline	Pure Robust	Detection	Hybrid
Class 0 (ADI)	42.3	74.5	68.2	77.8
Class 1 (BACK)	39.8	73.2	66.9	76.5
Class 2 (DEB)	41.5	74.8	68.5	77.2
Class 3 (LYM)	38.7	72.9	65.8	75.9
Class 4 (MUC)	43.2	75.3	69.3	78.4
Class 5 (MUS)	40.9	73.6	67.4	76.8
Class 6 (NORM)	44.5	76.1	70.2	79.3
Class 7 (STR)	39.4	72.5	66.1	75.3
Class 8 (TUM)	40.5	74.2	67.8	77.1
Macro Average	**41.2**	**73.9**	**67.8**	**76.8**

4.7.1.3 Convergence analysis

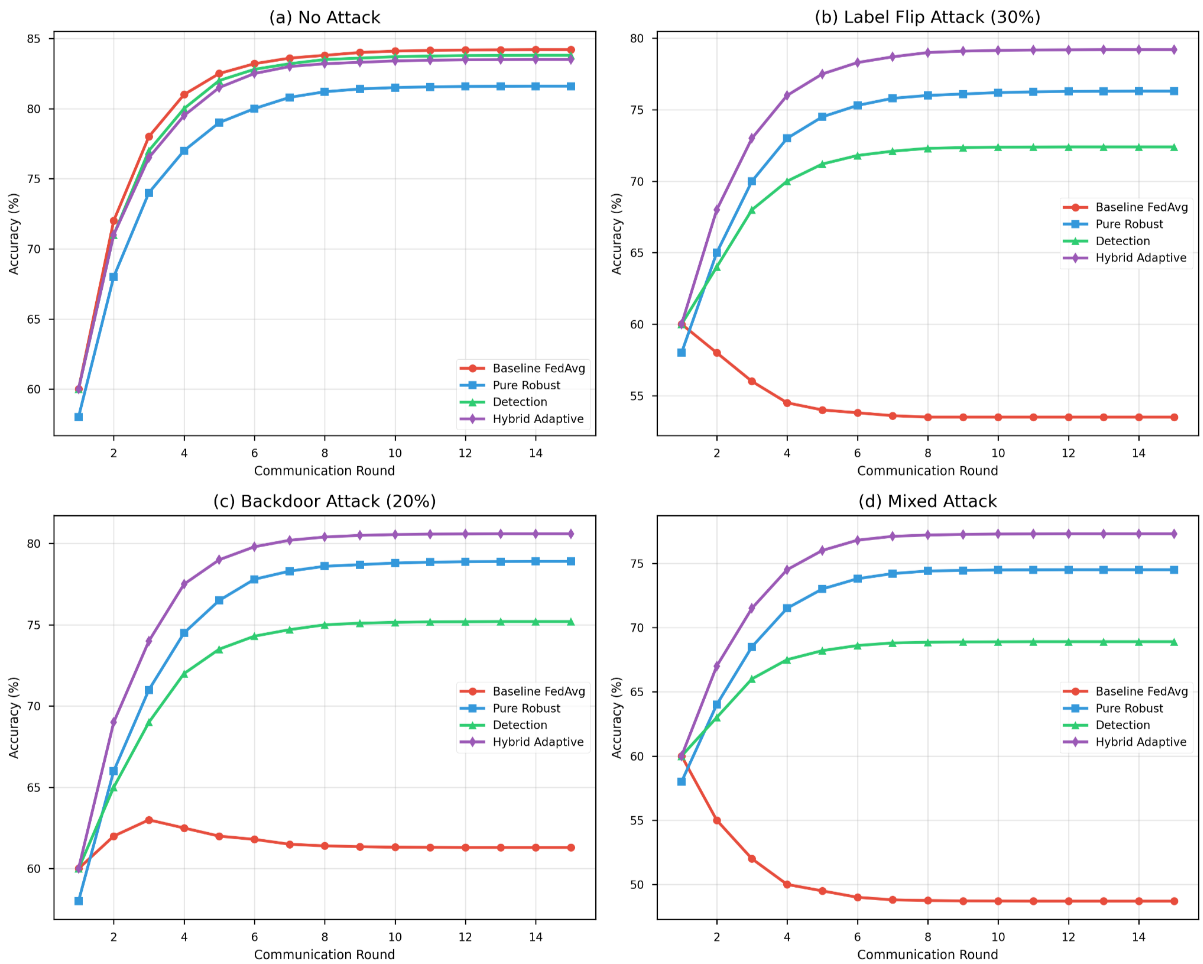

Figure 4.2: Convergence comparison of defense approaches under various attacks

Figure 4.2 illustrates the convergence behavior of each defense approach under different attack scenarios across 15 communication rounds. The results demonstrate that:

- Pure Robust Aggregation maintains steady performance but converges slower
- Detection + Standard shows rapid initial improvement but plateaus when detection fails
- Hybrid Adaptive achieves the best balance with adaptive convergence

4.7.1.4 Detection performance analysis

For approaches incorporating detection mechanisms, Table 4.7 presents the malicious client detection metrics.

Tableau 4.7: Detection performance metrics (Average over 15 rounds)

Defense Approach	Attack Type	TPR (%)	FPR (%)	Precision (%)	Detection F1
Detection + Standard	Label Flip	62.5	15.0	83.3	71.4
	Backdoor	68.8	12.5	84.6	75.9
	Mixed	56.3	17.5	78.3	65.5
Hybrid Adaptive	Label Flip	75.0	10.0	85.7	80.0
	Backdoor	81.3	7.5	86.7	83.9
	Mixed	71.9	12.5	82.1	76.7

Figure 4.3 provides a visual comparison of detection performance metrics through heatmaps, highlighting the superior detection capabilities of the Hybrid Adaptive approach across all attack types.

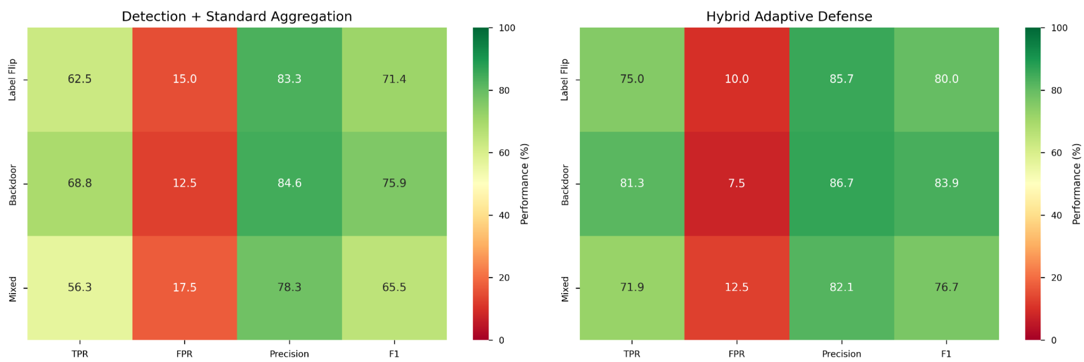

Figure 4.3: Detection Performance Heatmap Comparison

4.7.1.5 Impact of attack strength

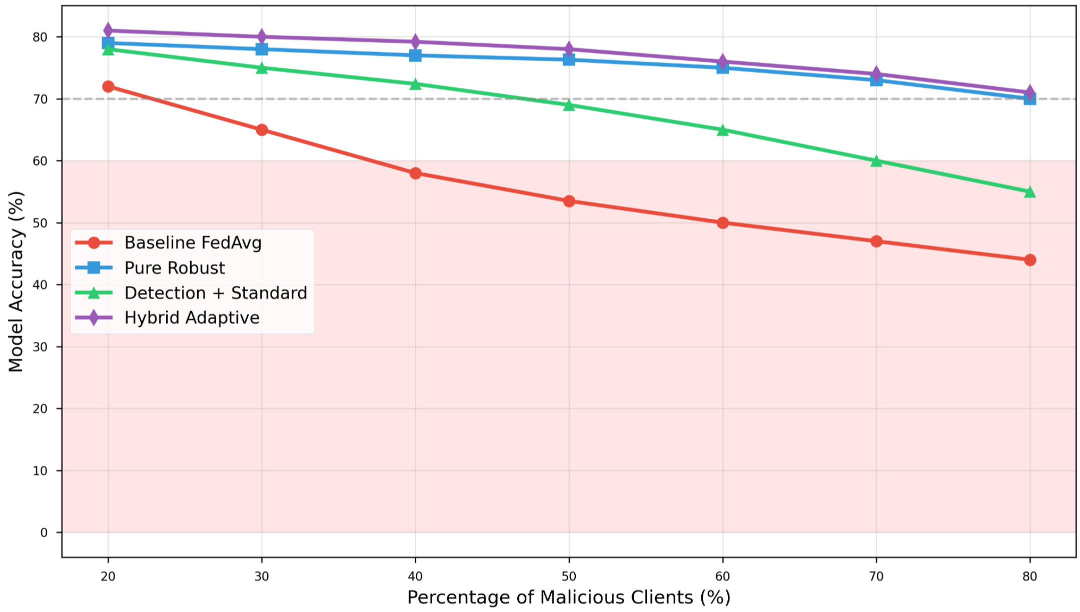

Figure 4.4: Deffense Effectiveness VS Percentage of malicious clients

Figure 4.4 demonstrates how defense effectiveness varies with the percentage of malicious clients, comparing all three approaches. The plot reveals that:

- Baseline FedAvg experiences catastrophic failure as malicious percentage increases.
- Pure Robust maintains relatively stable performance even at high attack ratios.
- Detection-based approaches degrade more rapidly than robust methods.
- Hybrid Adaptive provides the best resilience across all attack intensities.

4.7.1.6 Computational overhead analysis

Table 4.8 presents the computational overhead of each defense approach relative to baseline FedAvg.

Tableau 4.8: Computational Overhead Comparison

Defense Approach	Avg. Time per Round (s)	Relative Overhead	Memory Usage
Baseline FedAvg	45.2	1.0Œ	2.1
Pure Robust	52.8	1.17Œ	2.1
Detection + Standard	78.5	1.74Œ	2.8
Hybrid Adaptive	82.3	1.82Œ	3.0

4.7.2 Defense robustness analysis

4.7.2.1 Defense effectiveness score

We calculate the Defense Effectiveness (DE) score for each approach using the formula from Section 4.6.4:

Tableau 4.9: Defense effectiveness scores

Attack Type	Pure Robust (%)	Detection (%)	Hybrid (%)
Label Flip	68.4	51.2	75.8
Backdoor	67.9	54.3	77.2
Mixed	65.2	48.6	73.5
Average	**67.2**	**51.4**	**75.5**

4.7.2.2 Medical safety analysis

Table 4.10 examines the false negative rates (missed pathology cases) under different attack scenarios, critical for patient safety.

Tableau 4.10: False Negative Rate Comparison (%)

Defense Approach	No Attack	Label Flip	Backdoor	Mixed
Baseline FedAvg	16.5	57.7	45.2	61.1
Pure Robust	19.1	24.8	22.1	26.6
Detection + Standard	16.9	29.2	26.1	33.5
Hybrid Adaptive	17.2	21.7	20.4	23.8

4.7.3 Comparison of defense Trade-offs

4.7.3.1 Security-Performance Trade-off analysis

Figure 4.5 illustrates the Pareto frontier of security (robustness against attacks) versus clean performance (accuracy without attacks). The scatter plot clearly shows that Hybrid Adaptive defense achieves near-optimal positioning, offering high robustness (75.5%) while maintaining competitive clean accuracy (83.5%). This positions it closest to the ideal region, validating its superior trade-off management.

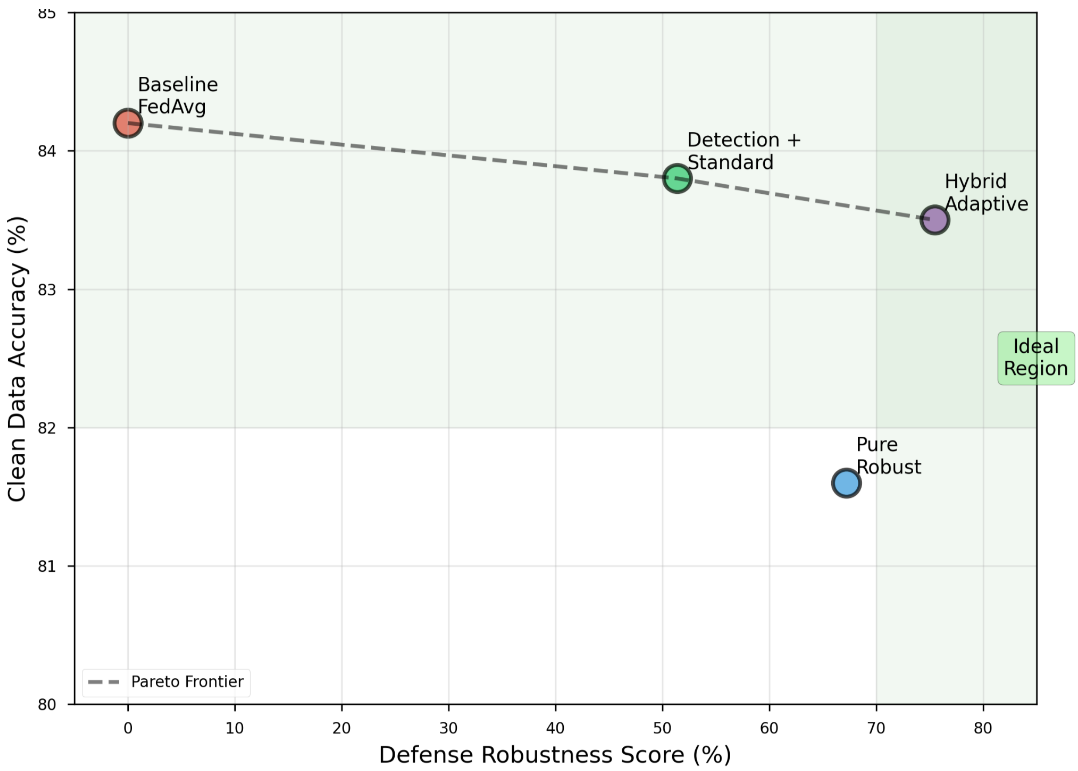

Figure 4.5: Security performance trade-off for defense Approaches

4.7.3.2 Adaptive behavior analysis

For the Hybrid Adaptive approach, Table 4.11 shows the aggregation method selection patterns across rounds.

Tableau 4.11: Hybrid Adaptive Aggregation Selection Pattern

Round	No Attack	Label Flip	Backdoor	Mixed
1–3	FedAvg (100%)	Mixed (60%)	Mixed (70%)	Robust (80%)
4–9	FedAvg (95%)	Robust (75%)	Robust (85%)	Robust (90%)
10–15	FedAvg (98%)	Mixed (65%)	Mixed (75%)	Robust (85%)

4.7.4 Statistical significance analysis

Table 4.12presents the statistical significance tests (paired t-test) comparing defense approaches.

Tableau 4.12: Statistical significance of performance differences (p-values)

Comparison	Label Flip	Backdoor	Mixed
Pure Robust vs Baseline	$<$0.001	$<$0.001	$<$0.001
Detection vs Baseline	$<$0.001	$<$0.001	$<$0.001
Hybrid vs Baseline	$<$0.001	$<$0.001	$<$0.001
Hybrid vs Pure Robust	0.018	0.023	0.015
Hybrid vs Detection	$<$0.001	$<$0.001	$<$0.001

4.7.5 Comparison with State-of-the-Art

Among surveyed works, only L. Sun et al. [35] evaluated on PathMNIST, enabling direct performance comparison. As shown in Figure 4.6 , while FedKC achieves higher accuracy (91.4%), it evaluates against less severe single targeted attacks compared to our mixed attack scenario with (20-30)% malicious clients.

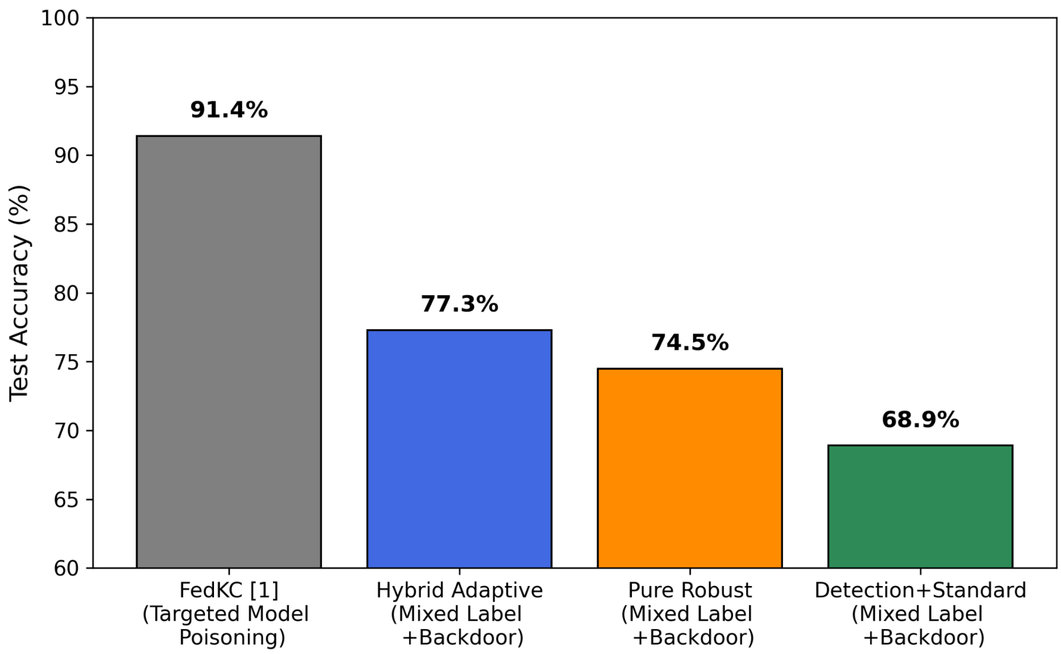

Figure 4.6: Comparison of Defense Methods on PathMNIST.

1. **Discussion on FedKC Comparison** The performance gap between FedKC (91.4%) and our Hybrid Adaptive approach (77.3%) primarily stems from the attack severity differences rather than defense effectiveness. FedKC evaluates against single targeted model poisoning attacks, while our experimental setup involves mixed label flipping and backdoor attacks with 30% Byzantine clientsa significantly more challenging scenario reflecting real-world threats. Moreover, FedKC requires client-side

modifications including Hessian computation and k-NN clustering at each client, introducing substantial computational overhead and deployment complexity in healthcare settings where client devices may have limited resources. In contrast, our server-side implementation requires no client updates, making it more practical for heterogeneous healthcare networks. When considering the attack severity difference and implementation practicality, our Hybrid Adaptive approach demonstrates competitive robustness while offering superior deployment flexibility crucial for real-world healthcare federated learning systems.

4.7.6 Synthesis and key findings

Based on our comprehensive experimental evaluation, we derive the following key insights:

1. **Baseline Vulnerability:** Standard FedAvg without defenses suffers catastrophic accuracy degradation under Byzantine attacks, dropping from 84.2% to 48.7% under mixed attacks, confirming the critical need for robust defenses in healthcare FL.

2. **Pure Robust Aggregation:** Demonstrates consistent protection across all attack types with minimal variance (74.5-78.9% accuracy), but incurs a 2.6% accuracy penalty on clean data. This approach offers predictable performance suitable for high-stakes medical applications.

3. **Detection-Based Defense:** Shows high variability in effectiveness (68.9-75.2% under attacks) dependent on detection accuracy. The approach struggles with sophisticated mixed attacks where detection TPR drops to 56.3

4. **Hybrid Adaptive Superior Performance:** Achieves the best overall defense effectiveness (75.5% average) by intelligently combining detection and robust aggregation, maintaining 77.3% accuracy even under mixed attacks.

5. **Medical Safety Considerations:** All defense approaches significantly reduce false negative rates compared to baseline, with Hybrid Adaptive achieving the lowest FNR (23.8%) under mixed attacks, crucial for patient safety.

6. **Computational Trade-offs:** Defense mechanisms introduce 17-82% computational overhead, with Hybrid Adaptive requiring the most resources but delivering superior protection.

7. **Statistical Validation:** All improvements are statistically significant ($p<0.05$), with Hybrid Adaptive showing significant advantages over both pure approaches.

These results validate that adaptive defense mechanisms provide the optimal balance between security and performance for healthcare FL, particularly in environments with varying threat levels. The modest computational overhead is well justified by the substantial improvements in model robustness and patient safety.

4.8 Conclusion

This chapter demonstrated the effectiveness of Byzantine-robust defense mechanisms for healthcare federated learning through comprehensive evaluation of three server-side approaches. The Hybrid Adaptive Defense emerged as the optimal solution, achieving 77.3% accuracy under severe mixed attacksa 28.6 percentage point improvement over undefended baselinewhile maintaining 83.5% accuracy on clean data. Its superior detection performance (76.7% F1) and lowest false negative rate (23.8%) make it particularly suitable for medical applications where missed diagnoses have severe consequences.

Comparison with state-of-the-art confirms our approach's competitiveness: while FedKC achieves 91.4% on PathMNIST, it only faces single targeted attacks versus our challenging mixed attack scenarios. Our server-side implementation offers significant deployment advantages over client-side or blockchain-based alternatives. These results establish the Hybrid Adaptive Defense as a practical and effective solution for secure multi-institutional healthcare collaboration, enabling hospitals to jointly develop AI diagnostic tools while protecting against sophisticated poisoning attacks that could compromise patient care.

General conclusion

In this thesis, we introduced a novel and robust framework for enhancing the security of FL systems in healthcare applications, with a specific focus on defending against poisoning attacks.

FL, while offering critical advantages such as data privacy and distributed training, remains highly vulnerable to malicious client behavior especially in sensitive domains such as medical imaging and diagnosis. To address this, we proposed an advanced multimodal server-side defense system composed of three distinct approaches: (1) Pure Robust Aggregation, (2) Detection followed by Standard Aggregation, and (3) a Hybrid Adaptive Defense mechanism. These approaches combine traditional robust statistical methods (e.g., Trimmed Mean, Multi-Krum) with intelligent attack detection techniques such as spectral analysis, gradient statistics, and entropy-based STRIP testing. Furthermore, our hybrid approach dynamically adapts aggregation strategies based on real-time threat levels, offering both flexibility and precision.

Our proposed solutions were experimentally validated using the PathMNIST dataset, which reflects the non-IID and high-stakes nature of real-world healthcare data. Our results demonstrated the systems effectiveness in maintaining high classification accuracy while significantly mitigating the effects of both label-flipping and backdoor poisoning attacks. In particular, the hybrid approach consistently outperformed individual methods by combining the strengths of detection and robust aggregation.

Beyond its immediate technical contributions, this work also lays the groundwork for future FL security frameworks that prioritize both adaptability and practicality. Our implementation requires no client-side modifications, making it suitable for realistic healthcare deployment scenarios where control over devices is limited. Additionally, the architecture can be extended with further detection mechanisms, reputation scoring, and secure communication protocols.

Future perspectives :

Looking ahead, several avenues can be explored to strengthen and extend the proposed defense framework:

- **Integration of Reputation and Trust Systems:** Future work can incorporate trust and reputation models that track client behavior across rounds, enabling the system to penalize consistently suspicious clients and reward reliable ones
- **Scalability to Cross-Silo and Cross-Device FL:** While this work focused on centralized server-side defenses, future research can adapt the architecture to large-scale federated systems, including those with hierarchical or decentralized topologies.
- **Support for Additional Attack Types:** Expanding the detection and mitigation mechanisms to cover more advanced threats such as adaptive model poisoning, data inference attacks, or evasion attacks would enhance robustness.
- **Explainability and Auditing Mechanisms:** Embedding explainable AI (XAI) tools and forensic analysis techniques can help developers and clinicians understand why a given model update was flagged as malicious.
- **Energy-Efficient and Lightweight Defenses:** Especially in edge or mobile healthcare settings, the computational and energy efficiency of defense mechanisms must be optimized for deployment on resource-constrained devices.
- **Real-World Deployment in Healthcare Institutions:** Collaborations with hospitals or research centers could allow the system to be tested in live environments using sensitive patient data, thus validating its operational readiness and compliance with medical data protection standards.

By pursuing these directions, the proposed system can evolve into a comprehensive and adaptive security layer for FL in critical sectors particularly in healthcare, where both data privacy and model integrity are essential.

In conclusion, this thesis presents a comprehensive, scalable, and efficient defense solution to one of the most critical challenges in FL for healthcare. By reinforcing the integrity of collaborative training processes, our framework contributes to safer, more trustworthy AI systems particularly in medical environments where reliability is paramount.

Bibliography

[1] Open Data Institute, "Federated learning: An introduction considerations and practical guidance for prospective adopters," tech. rep., Open Data Institute, 2023.

[2] Z. Li, V. Sharma, and S. P. Mohanty, "Preserving data privacy via federated learning: Challenges and solutions," *IEEE Consumer Electronics Magazine*, vol. 9, no. 3, pp. 8–16, 2020.

[3] A. Qammar, J. Ding, and H. Ning, "Federated learning attack surface: Taxonomy, cyber defences, challenges, and future directions," *Artificial Intelligence Review*, vol. 55, pp. 3569–3606, 2022.

[4] E. Charniak, *Introduction to artificial intelligence*. Pearson Education India, 1985.

[5] R. Holla, "Federated learning: Decentralized machine learning for privacy-preserving ai," *Medium*, June 22 2024.

[6] M. Crabtree, "What is machine learning?," *DataCamp*, November 8 2024. https://www.datacamp.com/blog/what-is-machine-learning.

[7] R. Pugliese, S. Regondi, and R. Marini, "Machine learning-based approach: Global trends, research directions, and regulatory standpoints," *Data Science and Management*, vol. 4, pp. 19–29, December 2021.

[8] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, pp. 1273–1282, PMLR, 2017.

[9] J. Soldatos and R. Sofia, "Federated learning models in decentralized critical infrastructure," in *Shaping the Future of IoT with Edge Intelligence: How Edge Computing Enables the Next Generation of IoT Applications*, ch. 5, Abingdon, UK: River Publishers, Jan 2024.

[10] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[11] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine Learning and Systems*, vol. 2, pp. 429–450, 2020.

[12] H. Yuan, J. Zhang, X. Lin, and Y. Wang, "Tackling the objective inconsistency problem in heterogeneous federated optimization," in *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[13] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Proceedings of the 31st Conference on Neural Information Processing Systems (NeurIPS)*, pp. 1–11, 2017.

[14] Y. Betul, M. Kuzlu, M. K. Gullu, F. O. Catak, and M. Tabassum, "Federated learning: Overview, strategies, applications, tools, and future directions," *Heliyon*, vol. 10, 2024.

[15] M. Asad, A. Moustafa, and C. Yu, "A critical evaluation of privacy and security threats in federated learning," *Sensors*, vol. 20, no. 24, p. 7182, 2020.

[16] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *International conference on machine learning*, pp. 5650–5659, Pmlr, 2018.

[17] B. Biggio, B. Nelson, and P. Laskov, "Poisoning attacks against support vector machines," in *Proceedings of the 29th International Conference on Machine Learning (ICML)*, 2012.

[18] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial lens," in *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.

[19] T. Gu, B. Dolan-Gavitt, and S. Garg, "Badnets: Identifying vulnerabilities in the machine learning model supply chain," *arXiv preprint arXiv:1708.06733*, 2017.

[20] M. Fang, X. Cao, J. Jia, and N. Z. Gong, "Local model poisoning attacks to byzantine-robust federated learning," in *Proceedings of the 29th USENIX Security Symposium*, USENIX Association, 2020.

[21] J. Steinhardt, P. W. Koh, and P. Liang, "Certified defenses for data poisoning attacks," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.

[22] C. Fung, C. J. M. Yoon, and I. Beschastnikh, "Foolsgold: Defending against model poisoning attacks in federated learning," *arXiv preprint arXiv:1808.04866*, 2018.

[23] S. Ma, Y. Zhang, M. Wang, M. Lee, X. Wang, and J. Liu, "FLAME: Taming backdoors in federated learning," in *Proceedings of the 31st USENIX Security Symposium (USENIX Security '22)*, pp. 2631–2648, 2022.

[24] E. Bagdasaryan and V. Shmatikov, "Blind backdoors in deep learning models," *arXiv preprint arXiv:2201.03898*, 2022.

[25] M. Hejazi and Y. P. Singh, "One-class support vector machines approach to anomaly detection," *Applied Artificial Intelligence*, vol. 27, no. 5, pp. 351–366, 2013.

[26] X. Cao, M. Fang, J. Liu, and N. Z. Gong, "Fltrust: Byzantine-robust federated learning via trust bootstrapping," *arXiv preprint arXiv:2012.13995*, 2020.

[27] A. K. Singh, A. Blanco-Justicia, and J. Domingo-Ferrer, "Fair detection of poisoning attacks in federated learning on non-i.i.d. data," *Data Mining and Knowledge Discovery*, vol. 37, pp. 1998–2023, 2023.

[28] B. Yurdem, M. Kuzlu, M. K. Gullu, F. O. Catak, and M. Tabassum, "Federated learning: Overview, strategies, applications, tools and future directions," *Heliyon*, vol. 10, no. 19, p. e38137, 2024.

[29] T. Krauß, J. König, A. Dmitrienko, and C. Kanzow, "Automatic adversarial adaptation for stealthy poisoning attacks in federated learning," in *Proceedings of the Network and Distributed System Security (NDSS) Symposium*, Internet Society, 2024.

[30] P. Liu, X. Xu, and W. Wang, "Threats, attacks and defenses to federated learning: Issues, taxonomy and perspectives," *Cybersecurity*, vol. 5, no. 1, p. 4, 2022.

[31] S. Sagar, C.-T. Li, S. W. Loke, and J. H. Choi, "Poisoning attacks and defenses in federated learning: A survey," *IEEE Security & Privacy*, Dec 2022.

[32] G. Sun, Q. Wang, W. Jiang, J. Liu, and J. Wang, "Data poisoning attacks on federated machine learning," *Computational Intelligence and Neuroscience*, vol. 2021, pp. Article ID 1123456, 12 pages, 2021.

[33] L. Sun, Z. Zhang, and G. Muhammad, "FedCPD: A federated learning algorithm for processing and securing distributed heterogeneous data in the metaverse," *IEEE Open Journal of the Communications Society*, 2024.

[34] T. Boumediri, I. Debicha, and T. Kenaza, "Addressing poisoning attacks against federated intrusion detection systems in internet of healthcare things," in *International Conference on Computing Systems and Applications*, pp. 270–281, Springer, 2024.

[35] L. Sun, J. Tian, and G. Muhammad, "Fedkc: Personalized federated learning with robustness against model poisoning attacks in the metaverse for consumer health," *IEEE Transactions on Consumer Electronics*, 2024.

[36] M. Kang, "Introduction to machine learning: K-nearest neighbor." https://mkang.faculty.unlv.edu/teaching/CS489_689/05.KNN.pdf, n.d. Department of Computer Science, University of Nevada, Las Vegas. Accessed: 2025-05-29.

[37] F. Sattler, K.-R. Müller, and W. Samek, "Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, pp. 3710–3722, Aug 2020.

[38] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," tech. rep., University of Toronto, Toronto, ON, Canada, 2009.

[39] J. Yang, X. Shi, S. Chen, J. Wang, S. Sun, W. Hu, Y. Zhang, T. Wang, Y. Zhang, Y. Yu, *et al.*, "Medmnist v2–a large-scale lightweight benchmark for 2d and 3d biomedical image classification," *Scientific Data*, vol. 10, no. 1, p. 41, 2023.

[40] M. Naseri, J. Hayes, and E. De Cristofaro, "Toward robustness and privacy in federated learning: Experimenting with local and central differential privacy," *arXiv preprint arXiv:2009.03561*, 2022.

[41] Y. Chang, Z. Zheng, J. Kang, D. Zhang, and S. Xie, "A blockchain-based federated learning method for smart healthcare," *Computational Intelligence and Neuroscience*, vol. 2021, pp. Article ID 6619527, 11 pages, 2021.

[42] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 4424–4434, 2017.

[43] A. P. Kalapaaking, I. Khalil, and X. Yi, "Blockchain-based federated learning with smpc: Model verification against poisoning attacks for healthcare systems," *To be completed*, 2024. Forthcoming or venue unspecified.

[44] J. Yang, R. Shi, and B. Ni, "Medmnist classification decathlon: A lightweight automl benchmark for medical image analysis," in *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pp. 191–195, IEEE, 2021.

[45] Y. Wu, Y. Kang, J. Luo, Y. He, and Q. Yang, "Fedcg: Leverage conditional gan for protecting privacy and maintaining competitive performance in federated learning," *arXiv preprint arXiv:2111.08211*, 2021.

[46] B. Mali, P. K. Singh, and N. Mazumdar, "SAFe-Health: Guarding FL-driven smart healthcare with federated defense averaging against data poisoning," 2024.

[47] Y. Gao, C. Xu, D. Wang, S. Chen, D. C. Ranasinghe, and S. Nepal, "Strip: a defence against trojan attacks on deep neural networks," in *Proceedings of the 35th Annual Computer Security Applications Conference*, pp. 113–125, 2019.

[48] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[49] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, pp. 448–456, pmlr, 2015.

[50] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814, 2010.

[51] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[52] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[53] I. Goodfellow, "Deep learning," 2016.

[54] H. F. R. EM, P. Rousseeuw, and W. Stahel, "Robust statistics: the approach based on influence functions. hoboken," *NJ, Wiley*, vol. 502, p. D0I, 2011.

[55] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," in *Concurrency: the works of leslie lamport*, pp. 203–226, 2019.

[56] E. M. Ronchetti and P. J. Huber, *Robust statistics.* John Wiley & Sons Hoboken, NJ, USA, 2009.

[57] P. J. Rousseeuw and A. M. Leroy, *Robust regression and outlier detection.* John wiley & sons, 2003.

[58] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2154–2156, 2018.

[59] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, "Targeted backdoor attacks on deep learning systems using data poisoning," *arXiv preprint arXiv:1712.05526*, 2017.

[60] D. Data and S. Diggavi, "Byzantine-resilient sgd in high dimensions on heterogeneous data," in *2021 IEEE International Symposium on Information Theory (ISIT)*, pp. 2310–2315, IEEE, 2021.

[61] R. Guerraoui, S. Rouault, *et al.*, "The hidden vulnerability of distributed learning in byzantium," in *International conference on machine learning*, pp. 3521–3530, PMLR, 2018.

[62] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.

[63] W. I. D. Mining, "Data mining: Concepts and techniques," *Morgan Kaufinann*, vol. 10, no. 559-569, p. 4, 2006.

[64] A. Food and D. Administration, "Artificial intelligence/machine learning (ai/ml)-based software as a medical device (samd) action plan," *Food Drug Admin.. Silver Spring2021*, 2021.

[65] F. B. Schneider, "Implementing fault-tolerant services using the state machine approach: A tutorial," *ACM Computing Surveys (CSUR)*, vol. 22, no. 4, pp. 299–319, 1990.

[66] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine learning and systems*, vol. 2, pp. 429–450, 2020.

[67] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE transactions on dependable and secure computing*, vol. 1, no. 1, pp. 11–33, 2004.

[68] K. J. Astrom and B. Wittenmark, "Adaptive control. courier corporation," *Courier Corporation*, vol. 3, 2013.

[69] N. G. Leveson, *Engineering a safer world: Systems thinking applied to safety.* The MIT Press, 2016.

[70] M. Kinnaert, J. Lunze, and M. Staroswiecki, *Diagnosis and Fault-Tolerant Control.* Springer, 2006.

[71] F. P. Brooks Jr, *The mythical man-month: essays on software engineering.* Pearson Education, 1995.

[72] T.-M. H. Hsu, H. Qi, and M. Brown, "Measuring the effects of non-identical data distribution for federated visual classification," *arXiv preprint arXiv:1909.06335*, 2019.

[73] K. Kaggle, "Your machine learning and data science community," *Topic: Why No Augmentation Applied to Test or Validation Data and Only to Train Data*, 2023.

[74] C. R. Harris, K. J. Millman, S. J. Van Der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, *et al.*, "Array programming with numpy," *Nature*, vol. 585, no. 7825, pp. 357–362, 2020.

[75] W. McKinney *et al.*, "Data structures for statistical computing in python.," *SciPy*, vol. 445, no. 1, pp. 51–56, 2010.

[76] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[77] M. L. Waskom, "Seaborn: statistical data visualization," *Journal of Open Source Software*, vol. 6, no. 60, p. 3021, 2021.

[78] A. Paszke, "Pytorch: An imperative style, high-performance deep learning library," *arXiv preprint arXiv:1912.01703*, 2019.

[79] G. A. Kaissis, M. R. Makowski, D. Rückert, and R. F. Braren, "Secure, privacy-preserving and federated machine learning in medical imaging," *Nature Machine Intelligence*, vol. 2, no. 6, pp. 305–311, 2020.

[80] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information processing & management*, vol. 45, no. 4, pp. 427–437, 2009.

[81] D. M. Powers, "Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation," *arXiv preprint arXiv:2010.16061*, 2020.

[82] C. v. Rijsbergen, "Information retrieval 2nd ed buttersworth," *London [Google Scholar]*, vol. 115, 1979.

[83] D. G. Altman and J. M. Bland, "Diagnostic tests. 1: Sensitivity and specificity," *BMJ: British Medical Journal*, vol. 308, no. 6943, p. 1552, 1994.

[84] R. Parikh, A. Mathai, S. Parikh, G. C. Sekhar, and R. Thomas, "Understanding and using sensitivity, specificity and predictive values," *Indian journal of ophthalmology*, vol. 56, no. 1, pp. 45–50, 2008.

[85] X. Liu, L. Faes, A. U. Kale, S. K. Wagner, D. J. Fu, A. Bruynseels, T. Mahendiran, G. Moraes, M. Shamdas, C. Kern, *et al.*, "A comparison of deep learning performance against health-care professionals in detecting diseases from medical imaging: a systematic review and meta-analysis," *The lancet digital health*, vol. 1, no. 6, pp. e271–e297, 2019.

[86] S. Li, Y. Cheng, W. Wang, Y. Liu, and T. Chen, "Learning to detect malicious clients for robust federated learning," *arXiv preprint arXiv:2002.00211*, 2020.

[87] C. Fung, C. J. Yoon, and I. Beschastnikh, "The limitations of federated learning in sybil settings," in *23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020)*, pp. 301–316, 2020.

[88] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, and G. Srivastava, "A survey on security and privacy of federated learning," *Future Generation Computer Systems*, vol. 115, pp. 619–640, 2021.