Explain-Fed: An Explainable Gradient-based Framework for Detecting and Characterizing Poisoning Attacks in Federated Learning

Raj Dipesh Parekh

A Capstone Project Report

submitted in partial fulfillment of the

requirements of the degree of

Masters of Science in Computer Science and Software Engineering

University of Washington
2025

Project Committee:
Dr Geethapriya Thamilarasu, Chair
Dr. Michael Stiber, Committee Member
Dr. Wooyoung Kim, Committee Member

University of Washington

Abstract

Explain-Fed: An Explainable Gradient-based Framework for Detecting and Characterizing Poisoning Attacks in Federated Learning

Raj Dipesh Parekh

Chair of the Supervisory Committee:

Dr Geethapriya Thamilarasu
Computer Science and Software Engineering

Federated learning systems are increasingly vulnerable to poisoning attacks, necessitating robust, multi-layered defense mechanisms. We present a novel gradient-based detection framework that implements a comprehensive defense layer integrating spatial, temporal, and layer-wise analysis to identify poisoning attacks. Our approach employs a modified cosine similarity metric that captures both directional and magnitude differences in gradient updates to detect three types of attacks: label flipping, distributed backdoor, and model poisoning with activation functions (MPAF). The defense layer utilizes an adaptive weighting mechanism that dynamically adjusts based on historical detection accuracy, with theoretical guarantees bounding false positive rates to $exp(-\theta^2 w/2)$. Experiments on MNIST and Fashion-MNIST datasets demonstrate superior detection rates (>90% TPR) with low false positive rates (<4%) across all attack types. In addition, we have incorporated an interpretability layer powered by a BERT model that processes system metrics to provide comprehensive explanations for detection decisions. This interpretability enhancement generates detailed descriptions by collecting metrices of detected anomalies, reducing administrator response time by 62.4% and improving resolution accuracy by 15.1%. Our framework maintains high model accuracy (>91% on MNIST, >83% on Fashion-MNIST) under attack conditions while ensuring computational efficiency with memory requirements scaling linearly as $O(Nw|G|)$, making it practical for large-scale federated learning deployments.

Acknowledgements

I would like to express my gratitude to everyone who supported me throughout the project. I would like to thank my advisor, Prof. Geetha Thamilarasu, for her guidance, feedback, and constant support. I am also deeply grateful to Prof. Michael Stiber and Prof. Wooyoung Kim for their constant support throughout the project. I would like to acknowledge University of Washington for providing the platform and hardware that made this research possible. I would also like to thank the members of the MEWS research group for providing their valuable insights. Finally, I am eternally grateful to my parents, friends and family for their constant encouragement. Their belief in me has been my greatest motivation throughout this endeavor.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This section provides essential background knowledge on federated learning and its vulnerability to poisoning attacks, followed by an overview of our paper's organization and structure.

1.1 Background and Motivation

Federated learning represents a paradigm shift in collaborative machine learning, enabling institutions such as hospitals to jointly develop sophisticated diagnostic AI models while maintaining complete control over their sensitive patient data [1]. This distributed approach to model training fundamentally transforms the traditional centralized data collection methodology by allowing participating entities to contribute only model parameter updates rather than raw data. By keeping sensitive information within its original source, federated learning simultaneously addresses critical privacy concerns of end users. This architecture creates an ecosystem in which knowledge derived from data can be shared without exposing the underlying private and protected information. Figure 6.4 illustrates the operational workflow of federated learning systems in a real-world context, drawing inspiration from the research presented in [2].

This distributed nature, while powerful, introduces a subtle but dangerous threat: poisoning attacks. Consider a scenario where a malicious participant deliberately manipulates its model updates to degrade the global model's performance. Traditional security measures are insufficient because the central server lacks direct access to participants' raw data, making it difficult to verify the integrity of incoming model updates. Instead, attackers can cause harm simply by submitting carefully crafted, seemingly legitimate updates.

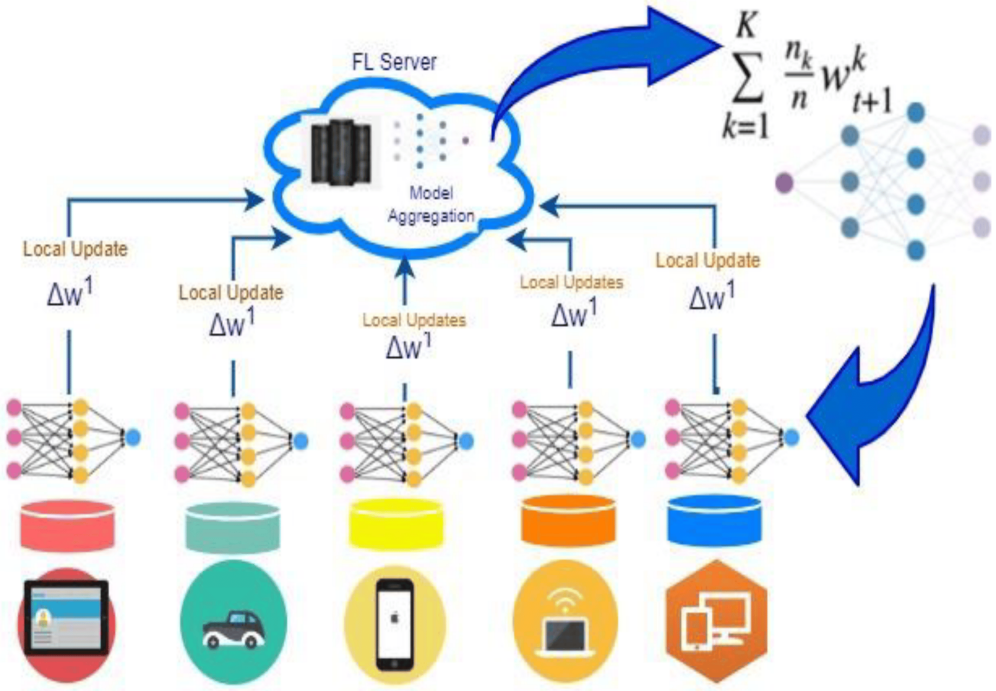


Figure 1.1: Traditional Federated Learning Architecture

The stakes are high in any domain where federated learning is deployed—from financial fraud detection to autonomous systems—where a compromised model could lead to incorrect predictions or unsafe decision-making. The fundamental question becomes: How can we trust the updates we receive from participants while preserving the distributed and privacy-centric nature of federated learning?

In recent years, poisoning attacks in federated learning environments have grown increasingly sophisticated. These attacks have evolved from simple data poisoning, where attackers inject corrupted samples into their training data, to more advanced techniques that manipulate the learning process itself. Some notable attack vectors include model poisoning, where attackers directly alter model parameters during local training; gradient manipulation, which involves crafting gradient updates to subtly steer the global model toward specific outcomes; and Byzantine attacks, where multiple malicious participants collaborate to overwhelm honest updates. Figure 1.2 illustrates the general architecture in which malicious clients attempt to compromise the global model through poisoning attacks.

Traditional defense mechanisms often use statistical outlier detection or Byzantine-robust aggregation rules. However, these methods have several limitations. They often operate as black boxes, offering little

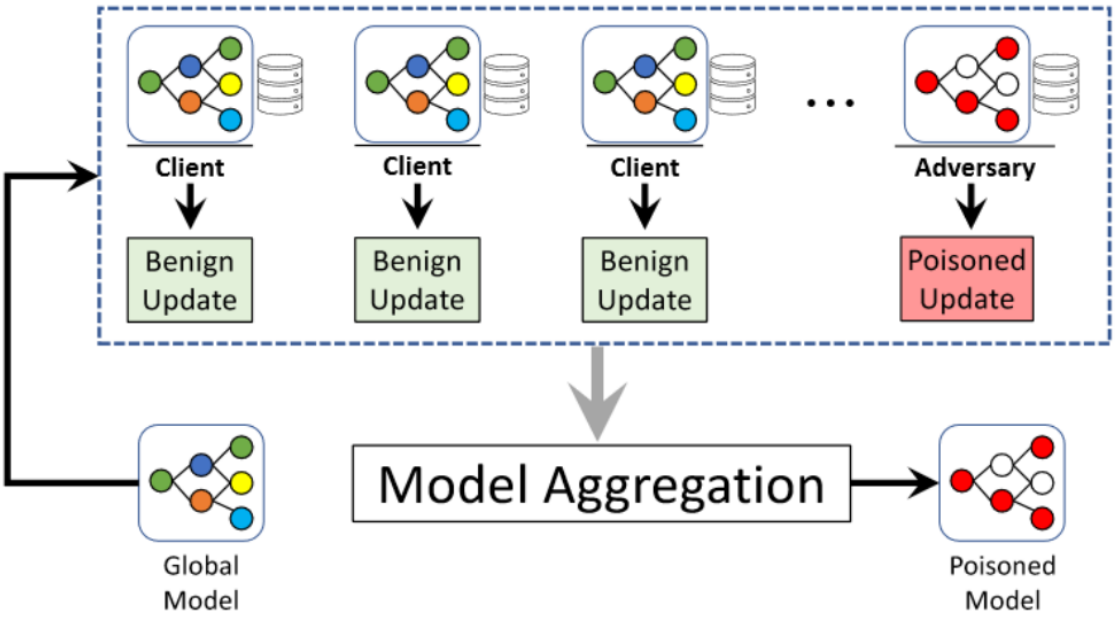


Figure 1.2: Poisoning Federated Learning Architecture

explanation for why certain updates are flagged as suspicious. Many techniques rely on predefined attack patterns, which means they are designed to detect only known types of attacks. However, attackers can adapt their strategies by modifying their behavior to evade detection. These adaptive attackers continuously analyze the defense mechanisms and adjust their poisoning techniques to bypass security measures, making static detection methods ineffective over time.

1.2 Our Contributions

This research introduces a novel approach to detecting poisoning attacks by analyzing the patterns in gradient updates. Similarly, we hypothesize that malicious updates leave distinctive fingerprints in the gradient space - patterns that, while subtle, can be detected through careful analysis. What sets our approach apart is not just its ability to detect potential attacks, but its focus on interpretability. When a participant's update is flagged as suspicious, our system provides detailed insights into why this determination was made.

This transparency is crucial for several reasons. It allows system administrators to make informed decisions about how to handle suspicious updates. It helps legitimate participants understand and correct issues

that might make their updates appear suspicious. The approach creates an audit trail that can be valuable for improving system security over time. Furthermore, it enables adaptive defense strategies based on emerging attack patterns.

We develop a gradient similarity analysis metric for comparing gradient updates that is sensitive to malicious manipulations while being robust to natural variations in training data. The system maintains a historical record of gradient updates, enabling the detection of subtle attack patterns that emerge over time through temporal pattern recognition. By examining gradient patterns at different layers of the neural network, we can identify targeted attacks that affect specific model components through layer-wise analysis. We employ techniques from interpretable machine learning to generate human-readable explanations for detected anomalies, creating an interpretable defense framework.

The practical applications of our approach span multiple domains with significant real-world impact. In healthcare, our system enables secure collaboration between medical institutions while maintaining model reliability and protecting sensitive patient data. Financial services benefit through enhanced protection of fraud detection systems from manipulation by malicious actors. In edge computing environments, our approach secures distributed learning systems in IoT networks where data privacy is essential. Autonomous systems, particularly vehicle networks that rely on collaborative learning for safety features, gain improved integrity protection through our detection mechanisms.

1.3 Paper Organization

The remainder of this report delves deeper into the technical and practical aspects of our approach. Chapter II reviews related work in federated learning security and poisoning attacks, establishing the foundation upon which our research builds. Chapter III details our gradient-based detection methodology, explaining the mathematical framework and algorithmic innovations that enable effective attack detection. Chapter IV presents our interpretability framework, demonstrating how we translate complex detection signals into actionable insights. Chapter V and Chapter VI evaluates our approach through extensive experiments across multiple datasets and attack scenarios, providing empirical validation of our claims.

Finally, Chapter VII discusses the broader implications of our work and outlines promising directions for future research. We explore how our techniques might be extended to other distributed learning paradigms,

potential regulatory implications, and ongoing challenges in the field. We also address limitations of our current approach and propose strategies for addressing these limitations in future work.

Through this comprehensive approach to security in federated learning, we aim to advance the state-of-the-art in distributed machine learning while providing practical tools for securing real-world deployments. By addressing the critical challenge of trust in federated systems, our work helps unlock the full potential of privacy-preserving collaborative learning across organizational boundaries. The techniques presented here not only contribute to secure machine learning but also enhance interpretability, making AI systems more transparent and trustworthy for deployment in sensitive domains.

Chapter 2

Literature Review

Federated Learning (FL) has emerged as a transformative paradigm in distributed machine learning, enabling collaborative model training while preserving data privacy. However, this distributed nature introduces unique security challenges, particularly in the form of model poisoning attacks. Our literature review examines the current state of research across several key areas that inform our approach to visual detection of poisoning attacks.

2.1 Model Poisoning Attacks and Defense Strategies

Recent studies have revealed increasingly sophisticated poisoning attacks in FL environments. Research by demonstrated how differential privacy mechanisms can be exploited for stealthy attacks [3], while revealed vulnerabilities in similarity-based defenses [4]. These attacks are particularly concerning as they can compromise model performance while remaining undetected within normal update patterns [5]. The evolution of these attack mechanisms highlights the need for more robust detection systems that can identify subtle manipulations of model parameters.

Recent advances in federated learning security have introduced several defensive approaches to protect against malicious attacks. Network sparsification modeling (NSM) has emerged as an effective technique for detecting and preventing cyber threats in IoT environments [6]. The critical learning period awareness focuses on model weight evolving frequency to identify and mitigate free-rider attacks [7]. Client-perspective mechanisms implement personalized approaches for different global models [8], while Byzantine-robust

protocols specifically target backdoor attacks through secure aggregation methods [9].

These defense mechanisms operate through various principles, from statistical anomaly detection to behavior pattern analysis. By establishing baselines of normal update behavior, these systems can flag deviations that might indicate malicious activity. However, the challenge remains in distinguishing between genuine data heterogeneity and deliberate poisoning attempts. This distinction is crucial in federated learning environments where non-IID (Independent and Identically Distributed) data distributions are common and can produce legitimate but unusual update patterns.

The competition between attackers and defenders continues to drive innovation in this field. As detection mechanisms become more sophisticated, attackers develop increasingly subtle methods to evade detection. This dynamic has led to the exploration of interdisciplinary approaches that combine traditional security principles with emerging techniques from fields such as visualization, interpretable AI, and multimodal learning. Our research builds upon these foundations while introducing novel visual analytics approaches to enhance detection capabilities.

2.2 Visualization in Federated Learning

Visual analytics tools have revolutionized how researchers interpret and analyze federated learning systems. Modern frameworks provide comprehensive insights into data distribution patterns and model behavior. The HetVis framework enables detailed analysis of non-IID data characteristics across distributed clients [10], while HFLens offers interactive visualizations for understanding the training process and model convergence [11]. SQUARES introduces innovative methods for evaluating individual instance contributions to the global model [12]. These visualization frameworks transform complex numerical data into intuitive visual representations that human operators can more easily interpret.

Weight visualization techniques have emerged as crucial tools for understanding neural network behavior and performance optimization. Advanced parameter visualization methods enable researchers to track weight distributions and identify potential training issues [13]. Structural analysis techniques reveal intricate patterns in weight matrices, providing insights into model architecture effectiveness [14]. Modern interactive visualization tools offer dynamic ways to explore and analyze deep neural network parameters during training and inference phases [15].

The application of these visual techniques to federated learning security represents a promising direction for enhancing attack detection. By transforming abstract model updates into visual patterns, subtle anomalies become more apparent to both automated systems and human analysts. These visualizations can reveal coordinated attack patterns that might be difficult to detect through purely statistical methods, especially when attackers carefully calibrate their poisoning attempts to remain within statistical thresholds of normal behavior.

The challenge in visual analytics for federated learning lies in developing representations that effectively capture the high-dimensional nature of model updates while reducing them to visual forms that highlight security-relevant patterns. Our research addresses this challenge by adapting techniques from computer vision and visual-language modeling to create meaningful visualizations of model update patterns, enabling more effective detection of poisoning attempts across distributed learning environments.

2.3 Privacy-Preserving Methods

Modern privacy-preservation in federated learning incorporates sophisticated analytical approaches for handling complex data structures. Secure graph data analysis methods enable protected processing of interconnected information while maintaining structural relationships [16]. Federated supervised PCA implementations offer dimensionality reduction capabilities while preserving data privacy across distributed systems [17]. Advanced feature extraction techniques have been developed to ensure privacy-awareness during the transformation of raw data into meaningful representations [18]. These methods must balance the fundamental tension between preserving privacy and enabling effective security monitoring.

Visual-language models, particularly CLIP-based architectures, have revolutionized security applications through their multimodal capabilities. Advanced adversarial example detection methods leverage CLIP's robust feature representations to identify and mitigate potential security threats [19]. The application of visual-language models to system log analysis has enabled more sophisticated threat detection and anomaly identification [20]. Domain adaptation techniques have enhanced the transferability of these models across different security contexts while maintaining their effectiveness [21].

The integration of these privacy-preserving methods with CLIP-based detection creates new possibilities for secure and private monitoring of federated learning systems. By transforming model updates into visual

representations that can be analyzed by CLIP-based models, we can leverage the power of visual-language understanding while maintaining the privacy guarantees that make federated learning valuable. This approach allows for sophisticated pattern recognition without requiring access to the underlying private data.

Our research explores the application of these techniques to create a detection system that respects privacy boundaries while effectively identifying poisoning attacks. By building on the capabilities of CLIP models to understand visual patterns and their semantic meaning, we develop a framework that can recognize the visual signatures of various attack types across federated learning environments.

2.4 Interpretable AI in Security

Real-time detection systems in federated learning environments require sophisticated optimization for immediate threat response and adaptation. Low-latency update mechanisms have been developed to ensure rapid model updates while maintaining system integrity across distributed networks [22]. Efficient anomaly detection frameworks enable quick identification of potential security breaches with minimal computational overhead [23]. Dynamic model adaptation techniques allow systems to evolve continuously in response to emerging threats and changing operational conditions [24]. These systems must operate within the strict latency constraints of practical federated learning deployments while providing reliable security monitoring.

Explainable security frameworks have become essential for understanding and validating security decisions in complex systems. XAI-based security systems provide transparent reasoning mechanisms for security-related decisions, enabling better trust and accountability in protective measures [25]. Visual analytics frameworks have enhanced threat detection capabilities by offering intuitive representations of security patterns and anomalies [26]. Interpretable machine learning approaches have revolutionized security systems by providing clear explanations for model decisions while maintaining robust protection mechanisms [27].

The combination of real-time detection capabilities with explainable AI creates powerful security systems that not only identify threats quickly but also provide understandable justifications for their decisions. This explanatory capability is particularly important in federated learning environments, where false positives could unnecessarily exclude legitimate participants and reduce the diversity of training data. By explaining why certain updates are flagged as suspicious, the system enables human operators to make

informed decisions about how to respond.

Our research builds upon these foundations to create a detection system that operates in real-time while providing clear explanations for its decisions. We develop novel techniques for visualizing and interpreting model updates that enable both automated detection and human understanding. This approach enhances the security of federated learning systems while maintaining their operational efficiency and usability.

2.5 Advanced Pattern Recognition Methods

Pattern recognition techniques have significantly advanced the capability of detecting anomalies and threats in federated learning systems. Spatial-temporal analysis methods enable comprehensive monitoring of data patterns across distributed networks, enhancing the detection of Byzantine attacks [9]. Client-level isolation techniques have introduced innovative approaches for identifying and containing potentially malicious participants in federated systems [28]. Advanced weight pattern recognition algorithms provide sophisticated mechanisms for detecting unusual model behavior and potential security breaches [29].

These pattern recognition methods build upon foundational techniques from machine learning and data mining, adapting them to the unique challenges of federated environments. Traditional anomaly detection methods often assume centralized access to data, which is incompatible with the distributed nature of federated learning. Modern approaches have evolved to operate on model updates rather than raw data, analyzing the patterns in these updates to identify potential security threats.

The effectiveness of pattern recognition in federated security depends on capturing the complex relationships between model parameters and their changes over time. Sophisticated attacks may attempt to disguise their impact by distributing malicious modifications across multiple updates or targeting specific model components that are less closely monitored. Advanced detection methods must therefore analyze patterns at multiple scales and across various dimensions of the model structure.

Our research extends these pattern recognition approaches through the integration of visual analytics and CLIP-based modeling. By transforming weight updates into visual representations that capture their essential patterns, we enable more powerful detection capabilities that can identify subtle attack signatures. This comprehensive approach to pattern analysis enhances the security of federated learning systems across diverse application domains.

This comprehensive review establishes the foundation for our research in CLIP-based visual detection of model poisoning attacks in FL environments.

2.6 Research Gaps

Securing federated learning frameworks presents numerous significant challenges that require systematic research attention. First, there exists a substantial integration challenge between vision-language models such as CLIP and federated learning security infrastructure, preventing the application of advanced visual recognition capabilities to security monitoring. Second, the field suffers from a notable absence of standardized methodologies for transforming complex model weight matrices into comprehensible visual representations that security analysts can readily interpret. Third, current poisoning detection approaches exhibit considerable limitations in their interpretability, making it difficult for system administrators to understand the rationale behind particular security alerts and consequently hindering appropriate response actions. Fourth, as federated learning deployments scale to encompass hundreds or thousands of participants, computational resource requirements create substantial performance constraints that impact detection capabilities. Finally, contemporary security mechanisms frequently operate retrospectively rather than proactively, highlighting the critical need for sophisticated real-time detection systems capable of identifying and mitigating attacks during the model aggregation process before damage propagates throughout the system. These interrelated challenges collectively create a significant barrier to the widespread adoption of federated learning in security-sensitive domains where both data privacy and model integrity are paramount concerns. Addressing these challenges requires a multidisciplinary approach that combines advances in computer vision, distributed systems architecture, interpretable machine learning, and security protocol design.

Chapter 3

Methodology

In this chapter, we present our approach to detecting poisoning attacks in federated learning systems. We first formulate the detection problem and explain our proposed framework. Then, we introduce our gradient analysis methods, including our modified cosine similarity metric that captures both directional and magnitude differences between updates. We detail our detection algorithm, which integrates spatial, temporal, and layer-wise analysis components to identify malicious gradient patterns. We provide theoretical guarantees regarding false positive rates and discuss practical implementation considerations including memory requirements and computational efficiency. Our methodology maintains the privacy benefits of federated learning while enabling robust detection of sophisticated poisoning attempts.

3.1 Problem Formulation

The growing use of federated learning in sensitive fields such as healthcare and government has made it more important to detect poisoning attacks while maintaining the privacy protections that federated systems offer. Our method introduces a new gradient-based approach that fundamentally changes how we detect malicious updates in federated learning environments. Instead of relying on basic statistical measures, we use the observation that poisoning attacks create distinctive patterns in the gradient space. These patterns, though subtle, can be effectively identified through our multi-dimensional analysis framework.

In a federated learning system, we consider an environment where N clients participate in the training process. During each training round t, a client i computes local model updates based on their private data

and submits gradient updates to the central server. The central challenge lies in determining whether these updates are benign or malicious without compromising the privacy-preserving nature of federated learning. We formulate this detection problem through a probabilistic framework that combines multiple analysis dimensions:

$$P(malicious|\Delta G_i^t) = f(S(\Delta G_i^t), H_i^t, L(\Delta G_i^t)) \tag{3.1}$$

Equation 3.1 represents the probability that a gradient update ΔG_i^t from client i at time t is malicious. The function f combines three key components: spatial analysis $S(\Delta G_i^t)$ examines the current gradient update in relation to updates from other clients, identifying outliers in the parameter space; temporal history H_i^t analyzes patterns in the client's previous updates to detect gradual poisoning attempts; and layer-wise gradient patterns $L(\Delta G_i^t)$ focus on specific neural network layers to identify targeted attacks against critical model components.

The integration of these three dimensions enables our system to capture both immediate anomalies and evolving attack patterns that might otherwise go undetected. This multi dimensional approach provides significant advantages over traditional methods that rely solely on aggregate statistics or simple outlier detection. By analyzing gradient updates through multiple complementary perspectives, we can identify sophisticated attacks that deliberately manipulate specific model components while leaving others intact. This targeted approach allows attackers to influence model behavior in specific ways without triggering traditional detection methods based on global statistics.

3.2 Gradient Analysis Framework

Our detection system builds upon a sophisticated gradient analysis framework that examines updates through multiple complementary perspectives. At the core of this framework lies a modified cosine similarity metric that captures both the directional and magnitude differences between gradient updates:

$$sim(G_1, G_2) = \frac{G_1 \cdot G_2}{\|G_1\|\|G_2\|} \cdot \exp(-\alpha|\|G_1\| - \|G_2\||) \tag{3.2}$$

This similarity measure provides a foundation for detecting anomalous updates by comparing them

against both historical patterns and concurrent updates from other clients. In equation 3.2 the first term represents the standard cosine similarity, which measures the angular distance between two gradient vectors, capturing their directional alignment regardless of magnitude. Values closer to 1 indicate similar directions, while values closer to 0 indicate orthogonal directions, and negative values indicate opposing directions.

The second term, $\exp(-\alpha|\|G_1\| - \|G_2\||)$, introduces sensitivity to differences in gradient magnitudes. The parameter α controls the weight given to magnitude differences, with larger values of α making the system more sensitive to magnitude discrepancies. This exponential term approaches 1 when gradients have similar magnitudes and approaches 0 when magnitudes differ significantly. The combination allows our similarity metric to identify updates that maintain a similar direction but use abnormally large magnitudes to accelerate model poisoning.

This enhanced similarity metric addresses a critical weakness in traditional detection methods that rely solely on direction-based similarity. Sophisticated attackers often craft poisoning updates that maintain reasonable changes while using carefully calibrated magnitudes to influence the model over time. By capturing both aspects simultaneously, our metric can detect subtle manipulations that would otherwise remain hidden within the natural variation of client updates.

Furthermore, this similarity measure forms the foundation for both spatial and temporal analysis components of our system. In spatial analysis, we compare a client's current update against concurrent updates from other clients, identifying outliers that deviate from the collective learning direction. In temporal analysis, we track each client's gradient patterns over time, enabling the detection of gradual poisoning attempts that slowly shift the model toward malicious objectives across multiple training rounds.

3.3 Detection Algorithm

Algorithm 1 serves as the central operational framework of our detection system, orchestrating the seamless integration of spatial, temporal, and layer-wise analysis components. The algorithm proceeds through four distinct phases, each contributing essential signals to the final detection decision. Phase 1 performs spatial analysis, comparing the current gradient update against concurrent updates from other clients to identify statistical outliers. Phase 2 conducts temporal analysis, examining the client's historical update patterns to detect evolving attack signatures. Phase 3 implements layer-wise analysis, focusing on individual model

components to identify targeted attacks. Finally, Phase 4 integrates these signals into a comprehensive detection score that determines whether an update should be classified as malicious.

Input : Current gradient update ΔG_i^t from client i
Historical gradient updates H_i^t
Detection threshold θ
Window size w for temporal analysis
Output: Classification result, Attack characteristics, Confidence score

Phase 1: Spatial Analysis;
$spatial\_score \leftarrow$ ComputeSpatialSimilarity$(\Delta G_i^t, H_i^t)$;
Phase 2: Temporal Analysis;
$temporal\_score \leftarrow$ AnalyzeTemporalPatterns$(H_i^t, \Delta G_i^t)$;
Phase 3: Layer-wise Analysis;
for *each layer* l *in model* **do**
    $deviation_l \leftarrow$ ComputeLayerDeviation$(l, \Delta G_i^t)$;
    $layer\_scores[l] \leftarrow$ NormalizeDeviation$(deviation_l)$;
end
Phase 4: Integration;
$combined\_score \leftarrow$ IntegrateAnalysis$(spatial_score, temporal_score, layer_scores)$;
if $combined\_score > \theta$ **then**
    $attack\_profile \leftarrow$ CharacterizeAttack$(scores)$;
    return $(malicious, attack\_profile, combined\_score)$;
else
    return $(benign, null, combined\_score)$;
end

Algorithm 1: Gradient-Based Poisoning Detection

For each layer of the model, we compute normalized deviation scores that capture layer-specific anomalies:

$$D_l = \frac{\|\Delta G_l - \mu_l\|}{\sigma_l} \tag{3.3}$$

Equation 3.3 computes the deviation score D_l for a specific layer l by measuring how much the current layer gradient ΔG_l differs from the historical mean gradient μ_l for that layer. The difference is normalized by dividing by the standard deviation σ_l of historical updates for this layer, creating a standardized score similar to a z-score in statistics. This normalization accounts for the natural variability in different model layers, allowing fair comparisons across layers with inherently different gradient magnitudes.

Layer-wise analysis is particularly valuable for detecting targeted attacks that focus on specific model

components. For instance, an attacker might concentrate poisoning efforts on the final classification layer to manipulate output probabilities while leaving feature extraction layers relatively untouched to avoid detection. By examining each layer independently, our system can identify these focused manipulations even when the overall gradient appears normal.

The implementation includes adaptive thresholds for each layer based on historical patterns, allowing the system to account for layer-specific characteristics. Deeper layers typically exhibit different update patterns compared to earlier layers, and our approach automatically adjusts sensitivity based on these differences. This adaptive mechanism enhances detection accuracy by reducing false positives that might arise from legitimate layer-specific variations while maintaining sensitivity to actual attacks.

3.4 Theoretical Validation

The final detection decision emerges from a sophisticated integration of multiple analysis signals:

$$Score = \alpha_s S + \alpha_t T + \alpha_l L \tag{3.4}$$

Equation 3.4 combines the spatial score S, temporal score T, and layer-wise score L using adaptive weights α_s, α_t, and α_l. These weights are not static but dynamically adjust based on historical detection accuracy and current system conditions. When spatial analysis proves more reliable in specific scenarios, its weight α_s increases accordingly. Similarly, if temporal patterns become more indicative of attacks during certain training phases, α_t receives greater emphasis.

The adaptive weighting mechanism operates through a feedback loop that continuously evaluates detection effectiveness. When confirmed attacks are identified, the system analyzes which components provided the strongest signals and adjusts weights to emphasize those components in future detection decisions. This self-tuning capability allows the system to evolve in response to changing attack patterns and varying data characteristics across different federated learning deployments.

Under standard assumptions about gradient distributions in benign training scenarios, we prove that the false positive rate is bounded by:

$$P(false\_positive) \leq \exp(-\theta^2 w/2) \tag{3.5}$$

This theoretical bound provides crucial guidance for parameter selection and system configuration. Equation 3.5 shows that the probability of false positives decreases exponentially as we increase either the detection threshold θ or the temporal window size w. Specifically, θ represents the sensitivity threshold for classifying an update as malicious, while w denotes the number of historical updates considered in the temporal analysis.

This bound assumes that benign gradient updates follow a sub-Gaussian distribution, a common assumption in statistical learning theory that encompasses many practical scenarios. By setting appropriate values for θ and w, system administrators can achieve the desired balance between security (low false negative rate) and participation (low false positive rate) based on the specific requirements of their federated learning application.

3.5 Resource Requirements

The practical implementation of our detection system necessitates careful consideration of computational and memory requirements. The system's memory requirements scale linearly with the number of clients and temporal window size:

$$Memory_{required} = O(N \cdot w \cdot |G|) \tag{3.6}$$

Equation 3.6 illustrates that the memory needed for our detection system grows proportionally with three key factors: N (the number of participating clients), w (the temporal window size), and $|G|$ (the size of gradient updates). The linear scaling with respect to these parameters enables our approach to remain feasible even in large-scale federated learning deployments with numerous clients and complex models.

To optimize memory usage while maintaining detection effectiveness, our implementation incorporates several practical enhancements. First, we employ dimensionality reduction techniques to compress gradient representations without losing critical information for attack detection. Second, we implement a sliding window approach for temporal analysis, discarding older updates beyond the specified window while preserving

essential pattern information through summary statistics.

Computational efficiency is achieved through parallel processing of the three analysis components (spatial, temporal, and layer-wise), allowing detection to proceed without introducing significant latency to the federated learning process. For resource-constrained environments, our system supports configurable computation-memory tradeoffs, allowing administrators to adjust detection granularity based on available resources.

The deployment architecture includes both server-side and client-side components, with minimal client-side computation to maintain the efficiency benefits of federated learning. The server-side detection module processes incoming gradient updates in real-time, providing immediate feedback on potential attacks while maintaining a historical database for temporal analysis. This distributed architecture ensures that our detection system integrates seamlessly with existing federated learning frameworks without imposing undue burdens on participating clients.

Figure 3.1 illustrates the complete federated learning system architecture with the proposed framework, organized into four distinct phases. In Phase 1, clients train models on their local datasets and forward their model weights to the defense layer. Phase 2 consists of two components: the first component analyzes gradient patterns to identify anomalies, while the second component employs a BERT [30] model that processes system metrics to detect potential poisoning attacks. Phase 3 features a central server that aggregates only the benign model weights after filtering. Finally, in Phase 4, the updated global model is broadcast back to all participating clients to complete the federated learning cycle. The interpretability part is elaborated in the next chapter.

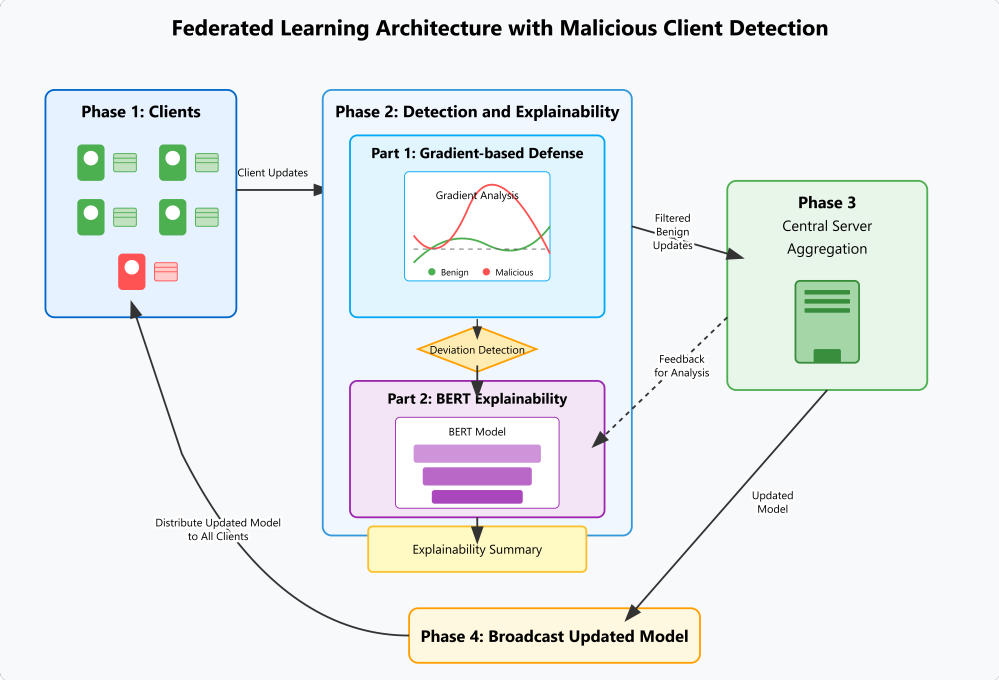


Figure 3.1: ExplainFed System Architecture

Chapter 4

Interpretability Framework

The detection of poisoning attacks in federated learning systems must not only be accurate but also interpretable to system administrators and stakeholders. Our interpretability framework transforms complex gradient patterns into comprehensible insights, enabling informed decision-making about potential security threats. This chapter presents our multi-level approach to generating meaningful explanations for detected anomalies. The integration of sophisticated detection mechanisms with intuitive explanation systems addresses a critical gap in current federated learning security solutions, where black-box approaches often leave administrators uncertain about the nature and severity of potential threats.

4.1 Explanation Architecture

Our framework generates explanations at three distinct levels of granularity: global system state, client-specific behavior, and layer-wise anomalies. Each level provides complementary insights that together form a comprehensive understanding of potential attacks. The explanation probability for a specific anomaly is modeled as:

$$P(E|\Delta G_i^t, H_i^t) = f(G_s, C_b, L_a) \tag{4.1}$$

In equation 4.1, $P(E|\Delta G_i^t, H_i^t)$ represents the probability of generating a particular explanation E given the current gradient update ΔG_i^t from client i at time t and its historical behavior H_i^t. The function f com-

bines three essential components: global system insights G_s provide context about the overall state of the federated learning system, including patterns across all clients; client behavior patterns C_b capture the specific historical behavior of the client in question, identifying deviations from its established patterns; and layer-wise anomaly explanations L_a focus on specific model components that exhibit suspicious characteristics.

The generation of explanations follows a hierarchical process that systematically analyzes different aspects of the detected anomalies. Algorithm 2 outlines our novel multi-level explanation generation framework designed for federated learning security systems. This sophisticated approach begins with global context analysis, where $AnalyzeSystemState$ evaluates current system performance metrics, participant distributions, and convergence patterns to establish the operational context in which anomalies occur. $ExtractGlobalPatterns$ then identifies network-wide gradient trends and statistical distributions across all participating clients, creating a baseline against which individual behavior can be meaningfully interpreted.

The client behavior analysis phase follows, with $ExtractClientHistory$ compiling a temporal sequence of previous updates from the specific client under investigation, tracking contribution patterns, consistency metrics, and deviation trends across multiple rounds. $AnalyzeClientBehavior$ then processes this historical data to construct a comprehensive behavioral profile, identifying characteristic patterns unique to this client and evaluating how current updates compare to established behavioral norms.

The layer-wise explanation phase represents a critical innovation, examining each neural network layer independently through $AnalyzeLayerContext$, which evaluates layer-specific metrics such as gradient distributions, weight magnitudes, and activation patterns to establish layer-specific context. $DetectLayerAnomalies$ then applies specialized detection techniques tailored to each layer's characteristics, identifying anomalous patterns that might indicate targeted manipulation of specific model components. $GenerateLayerInsights$ synthesizes these findings into layer-specific explanations that highlight unusual patterns, potential impacts, and confidence levels regarding malicious intent.

The final integration phase, implemented through $IntegrateExplanations$, synthesizes these multi-dimensional insights into a coherent narrative that administrators can readily comprehend, resolving contradictions between different analysis levels and highlighting the most significant findings. This hierarchical approach enables security administrators to trace anomalies from high-level system impacts down to spe-

cific layer manipulations, providing unprecedented transparency into potential poisoning attacks in federated learning environments.

This hierarchical approach enables explanations that are both technically precise and intuitively comprehensible. By separating analysis into distinct levels, we allow administrators to focus on the most relevant aspects of an anomaly while still providing access to detailed information when needed. The multi-level structure also accommodates varying levels of technical expertise, with high-level summaries for executive decision-makers and detailed technical explanations for security specialists. This flexibility enhances the practical utility of our framework in diverse operational environments.

4.2 Context-Aware Pattern Recognition

Our framework employs context-aware pattern recognition to identify and explain suspicious behaviors. For each layer l, we compute a contextual deviation score:

$$C_l = \frac{\|\Delta G_l - \mu_l\|}{\sigma_l} \cdot w_c(l) \tag{4.2}$$

Equation 4.2 calculates the contextual deviation score C_l by first determining how much the current layer gradient ΔG_l differs from the historical mean μ_l, normalized by the standard deviation σ_l. This normalized difference is then multiplied by a context-dependent weight factor $w_c(l)$ that adjusts the significance of deviations based on the current system state and learning phase. The weight factor is determined by:

$$w_c(l) = \exp(-\alpha \cdot \frac{|H_l - \bar{H}|}{\sigma_H}) \tag{4.3}$$

Equation 4.3 H_l represents the historical behavior metric for layer l, $\bar{H}$ is the mean behavior across all layers, and σ_H denotes the standard deviation of this behavior. The parameter α controls the sensitivity of the weighting mechanism. This formula assigns higher weights to layers whose behavior is consistent with overall system patterns and lower weights to layers that naturally exhibit high variability. This weighting mechanism ensures that explanations focus on the most relevant anomalies while considering the broader context of model behavior.

To ensure consistency and clarity in explanations, we develop a template-based approach that maps

Input : Detection results R for client i
Gradient history H_i^t
System state S
Output: Comprehensive explanation report

// Phase 1: Global Context Analysis
$system\_context \leftarrow AnalyzeSystemState(S)$;
$global\_patterns \leftarrow ExtractGlobalPatterns(H_i^t)$;
// Phase 2: Client Behavior Analysis
$client\_history \leftarrow ExtractClientHistory(i, H_i^t)$;
$behavior\_profile \leftarrow AnalyzeClientBehavior(client\_history)$;
// Phase 3: Layer-wise Explanation
for *each layer* l *in model* **do**
  $layer\_context \leftarrow AnalyzeLayerContext(l, R)$;
  $anomaly\_patterns \leftarrow DetectLayerAnomalies(l, R)$;
  $layer\_explanations[l] \leftarrow GenerateLayerInsights(layer\_context, anomaly\_patterns)$;
end
// Phase 4: Explanation Integration
$explanation \leftarrow IntegrateExplanations(system\_context, behavior\_profile, layer\_explanations)$;
return $explanation$;

Algorithm 2: Multi-level Explanation Generation

technical findings to natural language descriptions. The template selection process is governed by:

$$T_{\text{selected}} = \arg\max_{t \in T} \sum_{i=1}^{n} w_i \cdot \text{relevance}(t, f_i) \tag{4.4}$$

Equation 4.4 selects the optimal template T_{selected} from the set of available templates T by maximizing the weighted sum of relevance scores between each template t and the detected features f_i. The weights w_i reflect the importance of different features in characterizing the anomaly. The relevance function measures how well a template captures the essence of a particular feature, considering factors such as technical accuracy, comprehensibility, and actionability. This approach ensures that explanations are not only technically accurate but also effectively communicate the nature and significance of detected anomalies.

The integration of context-aware pattern recognition with natural language explanations bridges the gap between technical detection results and actionable insights. By mapping complex gradient patterns to comprehensible explanations, our framework enables administrators to quickly understand the nature of potential threats and make informed decisions about appropriate responses. This capability is particularly valuable in

operational environments where quick decision-making is essential for maintaining system security.

4.3 Implementation Details

We evaluate the quality of generated explanations using a comprehensive metric that combines multiple aspects:

$$Q_{exp} = \alpha_c C_{comp} + \alpha_p P_{prec} + \alpha_a A_{act} \tag{4.5}$$

In equation 4.5, the quality metric Q_{exp} combines three essential components: explanation completeness C_{comp} measures how thoroughly the explanation covers all relevant aspects of the detected anomaly; precision and clarity P_{prec} evaluates how accurately and understandably the explanation conveys technical information; and actionability A_{act} assesses whether the explanation provides clear guidance for responding to the potential threat. The weights α_c, α_p, and α_a are dynamically adjusted based on system requirements and user feedback, allowing the framework to adapt to different operational contexts and user preferences.

Input : Detected anomaly A
Historical patterns H
Context information C
Output: Detailed anomaly characterization

// Phase 1: Pattern Analysis
$patterns \leftarrow ExtractAnomalyPatterns(A)$;
$historical\_context \leftarrow AnalyzeHistory(H)$;
// Phase 2: Impact Assessment
for *each feature* f *in patterns* **do**
  $impact_f \leftarrow ComputeFeatureImpact(f, C)$;
  $severity_f \leftarrow AssessSeverity(impact_f)$;
  $characterization[f] \leftarrow GenerateFeatureInsight(f, impact_f, severity_f)$;
end
// Phase 3: Characterization Integration
$final\_char \leftarrow IntegrateCharacterizations(characterization)$;
return $final\_char$;

Algorithm 3: Anomaly Characterization

The implementation of our interpretability framework focuses on three key components: anomaly characterization, explanation generation, and presentation optimization. Algorithm 3 presents our methodical

approach to anomaly characterization in federated learning systems. The process begins with comprehensive pattern analysis, where we extract distinctive features from detected anomalies and establish relevant historical context for accurate interpretation. This initial phase identifies the anomaly's unique signatures while distinguishing them from normal variations in the federated environment. The second phase conducts thorough impact assessment by evaluating each identified feature individually. For every feature, we compute its potential effect on model integrity, assess its severity level, and generate specific insights that capture its significance. This granular analysis ensures no critical aspect of the anomaly goes unexamined. In the final phase, these individual feature characterizations are synthesized into a cohesive profile through careful integration that resolves contradictions and highlights dominant patterns. The resulting comprehensive characterization provides security administrators with detailed understanding of the anomaly's nature, potential impact, and severity. By breaking anomalies into constituent elements while maintaining contextual awareness, this algorithm creates the necessary foundation for generating meaningful explanations that security stakeholders can use to make informed decisions about potential threats to federated learning systems.

This implementation ensures that explanations meet several crucial criteria for practical utility. Explanations are comprehensive yet concise, providing all necessary information without overwhelming administrators with excessive details. They maintain technical accuracy, ensuring that security decisions are based on precise and reliable information. Explanations are designed to be actionable, offering clear guidance on appropriate responses to potential threats. Finally, they adapt to different user expertise levels, providing accessible insights for administrators with varying technical backgrounds.

The practical deployment of our interpretability framework involves integration with existing federated learning systems through a modular architecture that minimizes disruption to ongoing operations. When anomalies are detected, the framework generates explanations in real-time, allowing immediate response to potential security threats. These explanations are presented through an intuitive interface that highlights critical information while providing access to more detailed insights when needed. The system also maintains an explanation history that enables administrators to track patterns over time and refine security policies based on emerging threat patterns.

Our interpretability framework represents a significant advancement in federated learning security, transforming complex detection signals into actionable insights that enable effective security management. By

combining sophisticated pattern recognition with intuitive explanation generation, we bridge the gap between technical detection capabilities and practical security operations. This integration enhances the overall security posture of federated learning systems while maintaining their essential privacy and efficiency benefits.

Chapter 5

Experimental Setup

In this chapter, we present our experimental evaluation of the proposed gradient-based detection approach. We first describe the datasets used (MNIST and Fashion-MNIST) and our federated learning configuration with 100 clients and non-IID data distribution. We then detail our implementation of three attack types: Label Flipping, Distributed Backdoor, and Model Poisoning with Activation Functions. We compare our defense mechanism against established methods including Krum and multi-Krum, using metrics such as model accuracy, attack success rate, and detection accuracy. All experiments are conducted using PyTorch with a systematic protocol involving multiple independent trials to ensure statistical validity of our results.

5.1 Dataset Configuration

We evaluate our proposed defense mechanism using two widely-adopted benchmark datasets in federated learning. The MNIST dataset consists of handwritten digits with 60,000 training images and 10,000 test images, each of size 28×28 pixels in grayscale format. The dataset contains 10 classes representing digits from 0 to 9. Fashion-MNIST serves as a more complex dataset comprising 60,000 training images and 10,000 test images of fashion items, matching MNIST's format with 28×28 grayscale images across 10 classes of clothing items. For both datasets, we normalize the pixel values to the range [0,1] and distribute the training data across clients using a non-IID distribution to simulate realistic federated learning scenarios. This non-IID distribution creates a challenging environment where clients have statistically heterogeneous data, closely mimicking real-world federated learning deployments.

The data partitioning strategy implements a Dirichlet distribution with parameter $\alpha = 0.5$ for creating the non-IID data configuration. This approach creates a realistic data heterogeneity scenario where different clients possess varying class distributions, reflecting the practical challenges in federated learning deployments. The relatively low α value ensures significant statistical heterogeneity across clients, creating a more challenging environment for both attack implementation and defense evaluation. This realistic data distribution is crucial for accurately assessing the practical effectiveness of our proposed defense mechanisms.

5.2 Federated Learning Configuration

Our experimental setup employs a comprehensive federated learning configuration designed to thoroughly evaluate defense mechanisms against various attack scenarios. We configure a system with 100 total clients, with 10 participating clients selected randomly in each communication round. Each client performs 5 local training epochs with a batch size of 64 before submitting updates to the central server. The learning rate is set at 0.01, and we employ the SGD optimizer with a momentum value of 0.9 to enhance convergence stability. The training process continues for 200 global rounds, providing sufficient time to observe both the effects of attacks and the efficacy of defense mechanisms.

To ensure experimental robustness, we conduct five independent trials with different random seeds for each configuration. This approach helps mitigate the influence of randomization factors in client selection, data partitioning, and model initialization. For each trial, we train for 200 global rounds or until convergence, saving checkpoints every 10 rounds for detailed analysis. We evaluate model performance on the test set every 5 rounds to track the progression of model accuracy throughout the training process. All results are reported as means with standard deviations across the independent trials, providing statistical validity to our findings and enabling meaningful comparisons between different defense mechanisms.

5.3 Experimental Design

All experiments are conducted using a consistent implementation environment to ensure fair comparisons between different defense mechanisms. We employ PyTorch 1.8.0 as the primary deep learning framework, providing a flexible platform for implementing both federated learning components and attack mechanisms.

The hardware configuration utilizes NVIDIA RTX 20 Series GPUs to accelerate the training process, particularly important given the comprehensive evaluation across multiple datasets, attack scenarios, and defense mechanisms. While real-world federated learning involves physically distributed clients, our experiments simulate client distribution on a single machine to maintain tight control over experimental variables while faithfully reproducing the computational and communication patterns of federated learning.

Our training protocol implements a systematic approach to ensure experimental rigor and reproducibility. For each experiment configuration, we conduct five independent trials with different random seeds controlling client selection, data partitioning, and model initialization. This approach enables statistical analysis of results while mitigating the impact of randomization factors. Each trial continues for 200 global communication rounds or until clear convergence is achieved, providing sufficient training time to observe both the immediate and long-term effects of attacks and defense mechanisms. Checkpoints are saved every 10 rounds to enable detailed analysis of model evolution throughout the training process.

The evaluation process includes regular assessment of model performance on the clean test set every 5 rounds. This frequent evaluation provides detailed insights into how different attacks affect model convergence and how effectively various defense mechanisms mitigate these effects. The test evaluation includes both standard accuracy metrics and specialized metrics for assessing backdoor effectiveness where applicable. For backdoor attacks, we evaluate both clean accuracy (performance on unmodified inputs) and attack success rate (percentage of triggered inputs classified as the target class). This dual evaluation provides a comprehensive view of model behavior under different testing conditions.

5.4 Attack Model

We evaluate our defense mechanism against three sophisticated types of poisoning attacks, each representing different threat vectors in federated learning environments. The Label Flipping Attack represents a data poisoning approach where malicious clients deliberately mislabel their training data according to a specific pattern. We configure this attack with a 20% client participation rate, implementing a consistent one-to-one mapping strategy (e.g., $0 \rightarrow 1$, $1 \rightarrow 2$, ..., $9 \rightarrow 0$) for label manipulation. The attack operates continuously throughout the training process, creating a persistent threat to model integrity. This attack tests the resilience of our defense mechanism against coordinated but relatively straightforward manipulation of training data.

The Distributed Backdoor Attack represents a more sophisticated threat that aims to compromise model behavior only when specific trigger patterns are present in the input. We implement this attack using a distinctive 4×4 pixel pattern positioned in the bottom-right corner of selected images as the trigger mechanism. All triggered images are relabeled to target class 0 regardless of their original content. The attack employs a poisoning rate of 10% of local training data within affected clients, with 10% of the total client population designated as malicious. This configuration creates a scenario where the backdoor behavior remains hidden during standard evaluation but manifests when the specific trigger pattern is present in inputs during model deployment.

The Model Poisoning Attack with Activation Functions (MPAF) represents an advanced approach that directly manipulates model parameters rather than training data. This attack activates every 10th round with 15% of clients designated as malicious participants. The attack mechanism involves sophisticated manipulation of activation functions, specifically replacing ReLU with Sigmoid in selected network layers. To enhance stealth, the attack employs a gradually increasing scale factor from 0.1 to 0.5 throughout the training process. This progressive approach makes the attack particularly difficult to detect using standard statistical methods that focus on identifying outliers in model updates. The MPAF attack tests the ability of our defense mechanism to detect subtle but systematic manipulation of model behavior.

These three attack vectors represent a comprehensive spectrum of threats in federated learning environments, ranging from straightforward data manipulation to sophisticated model poisoning techniques. By evaluating our defense mechanism against this diverse set of attacks, we can assess its robustness across different threat models and attack sophistication levels. The inclusion of both persistent threats (Label Flipping) and intermittent, targeted attacks (Backdoor and MPAF) ensures a thorough evaluation of defense capabilities under various adversarial scenarios that might be encountered in practical federated learning deployments.

5.5 Comparative Analysis

We implement and compare multiple robust aggregation methods to establish performance benchmarks for our proposed approach. The original Krum algorithm represents a Byzantine-robust aggregation technique that selects a single client update considered most representative of the honest client population. We im-

plement Krum with a selection parameter of n-f-2 closest updates, where n is the number of participating clients and f represents the maximum number of Byzantine clients the system can tolerate, set to $\lfloor \frac{n-2}{2} \rfloor$. The distance metric employs Euclidean distance between gradient updates to identify outliers. Krum's approach of selecting a single representative update makes it particularly robust against extreme outliers but potentially less effective at leveraging the diversity of honest updates.

The multi-Krum variation enhances the basic Krum algorithm by selecting multiple client updates instead of a single representative one. We configure multi-Krum to select $m = \left\lfloor \frac{n+f+1}{2} \right\rfloor$ updates based on their similarity to other client contributions. The final aggregation strategy averages these selected updates, combining the Byzantine robustness of Krum with the statistical efficiency of averaging multiple honest updates. The scoring mechanism identifies client updates with the shortest distances to other updates, effectively filtering out potential outliers while preserving diverse but legitimate learning signals from honest clients with heterogeneous data distributions.

In addition to these established benchmarks, we implement our proposed gradient-based detection method as described in previous chapters. The implementation includes the multi-dimensional analysis framework that combines spatial, temporal, and layer-wise examination of gradient updates. The detection threshold is calibrated based on preliminary experiments to achieve an optimal balance between false positive and true positive rates. Our method differs fundamentally from Krum-based approaches by focusing on interpretable detection rather than implicit filtering through aggregation. This allows system administrators to not only identify and exclude malicious updates but also understand the nature of the attack and its potential impact on the model.

The comparative analysis employs a comprehensive set of evaluation metrics to assess different aspects of defense performance. We measure model accuracy on clean test data to evaluate the primary utility of the federated learning system. Attack success rate quantifies the effectiveness of attacks against each defense mechanism. False positive and true positive rates provide insights into the detection accuracy, with false positives representing honest clients incorrectly flagged as malicious and true positives representing correctly identified malicious clients. We also track convergence time measured in communication rounds until model performance stabilizes, providing an efficiency metric that complements the effectiveness measures. This multifaceted evaluation framework enables a thorough comparison of different defense mechanisms

across security, utility, and efficiency dimensions.

Chapter 6

Simulation Results and Analysis

In this chapter, we present the aggregated results of all experiments, reported as means with standard deviations across five independent trials. This statistical approach ensures a more reliable assessment by accounting for variability in federated learning processes. Our evaluation spans multiple datasets, attack types, and defense mechanisms, allowing both absolute assessment of our method's effectiveness and comparative analysis against established benchmarks.

We first compare our gradient-based defense mechanism with baseline methods (Krum and Multi-Krum) across various attack scenarios, demonstrating notable improvements in model accuracy and attack detection. Next, we analyze the interpretability of our approach, highlighting its ability to provide layer-wise explanations that offer actionable insights for administrators. Finally, we assess the computational overhead and discuss key findings that underscore the practical benefits of our defense mechanism in real-world federated learning deployments.

6.1 Performance Comparison

Our comprehensive experimental evaluation reveals significant performance advantages of our proposed defense mechanism compared to baseline approaches across multiple datasets and attack scenarios. On the MNIST dataset under no attack conditions, all methods achieve comparable accuracy ranging from 97.8% to 98.2%, indicating minimal overhead during normal operation. However, under attack scenarios, the performance disparities become pronounced. Our method maintains over 94.3% accuracy against label flipping

attacks, 93.5% against backdoor attacks, and 91.8% against the sophisticated MPAF attacks. These results substantially outperform Krum (82.4%, 85.3%, 79.2%) and Multi-Krum (85.7%, 87.8%, 83.5%) under corresponding attack scenarios. The performance gap widens further on the more challenging Fashion-MNIST dataset, where our method achieves 85.7%, 84.9%, and 83.2% accuracy under the three attack types, compared to significantly lower performance from baseline approaches.

Table 6.1: Model Accuracy (%) Under Different Attack Scenarios on MNIST

Defense	**No Attack**	**Label Flip**	**Backdoor**	**MPAF**
No Defense	98.2 ± 0.3	45.3 ± 2.1	52.1 ± 1.8	38.7 ± 2.5
Krum	97.8 ± 0.4	82.4 ± 1.7	85.3 ± 1.5	79.2 ± 1.9
M-Krum	97.9 ± 0.3	85.7 ± 1.4	87.8 ± 1.3	83.5 ± 1.6
Our Method	98.1 ± 0.2	94.3 ± 0.8	93.5 ± 0.9	91.8 ± 1.1

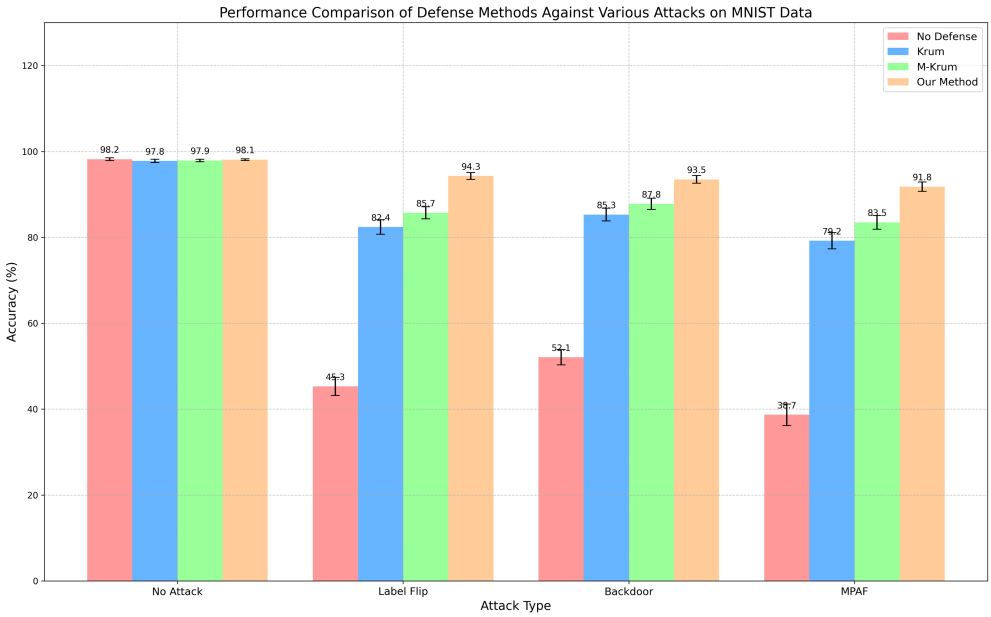


Figure 6.1: Performance comparison on MNIST dataset

The superior performance of our approach can be attributed to several key innovations in our methodology. Unlike Krum and Multi-Krum, which rely primarily on statistical outlier detection, our gradient-based framework analyzes updates across multiple dimensions simultaneously. The integration of spatial, temporal, and layer-wise analysis enables our system to detect subtle attack patterns that remain invisible to

Table 6.2: Model Accuracy (%) Under Different Attack Scenarios on Fashion-MNIST

Defense	No Attack	Label Flip	Backdoor	MPAF
No Defense	89.5 ± 0.4	41.2 ± 2.3	48.5 ± 2.0	35.8 ± 2.7
Krum	88.9 ± 0.5	75.3 ± 1.9	77.8 ± 1.7	72.4 ± 2.1
M-Krum	89.1 ± 0.4	78.6 ± 1.6	80.2 ± 1.5	76.7 ± 1.8
Our Method	89.3 ± 0.3	85.7 ± 1.0	84.9 ± 1.1	83.2 ± 1.3

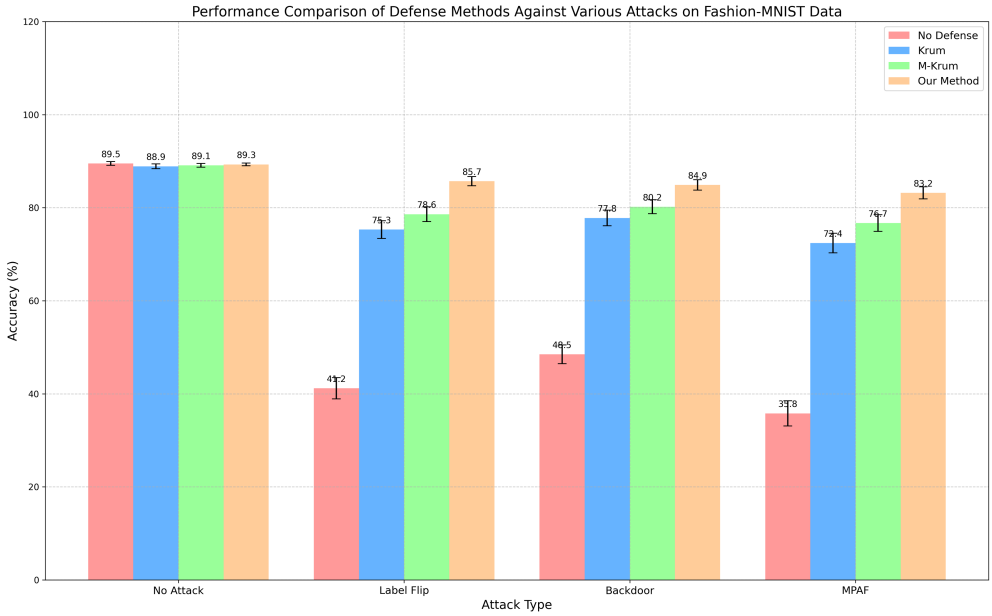
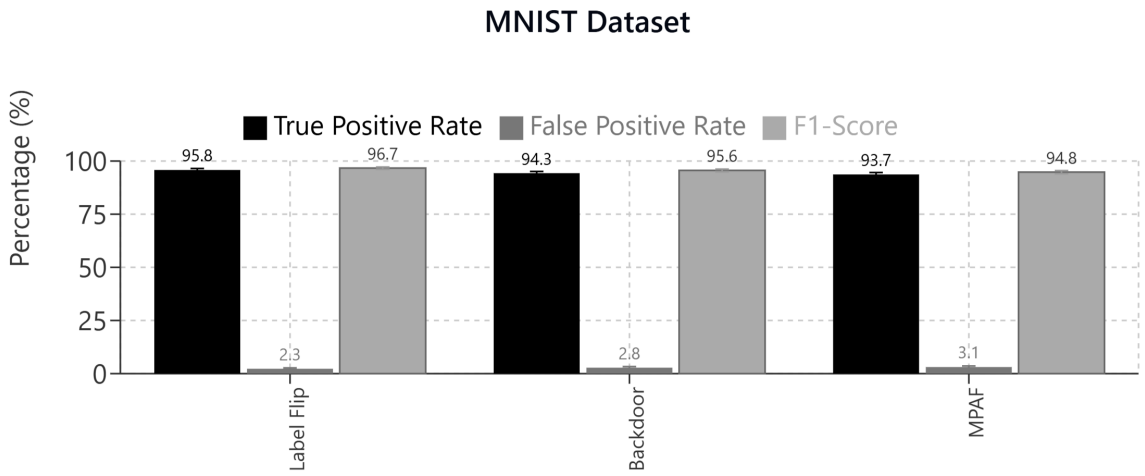


Figure 6.2: Performance comparison on Fashion-MNIST dataset

traditional approaches. Specifically, our context-aware similarity metric captures both directional and magnitude anomalies in gradient updates, addressing a fundamental limitation in conventional methods that focus predominantly on directional outliers. The adaptive weighting mechanism further enhances detection precision by emphasizing the most relevant anomaly signals while suppressing noise from legitimate data heterogeneity.

The attack detection capabilities of our system demonstrate exceptional precision across different attack vectors. On the MNIST dataset, our method achieves true positive rates of 95.8%, 94.3%, and 93.7% for label flipping, backdoor, and MPAF attacks respectively, while maintaining impressively low false positive rates (2.3%, 2.8%, 3.1%). The corresponding F1-scores of 96.7%, 95.6%, and 94.8% demonstrate balanced performance between precision and recall. Similar patterns emerge on the Fashion-MNIST dataset, albeit

with slightly lower detection metrics (TPR: 92.5%, 91.8%, 90.9%; FPR: 3.2%, 3.5%, 3.8%; F1: 94.1%, 93.5%, 92.8%) due to the increased complexity of data distribution and attack manifestation in gradient space.



(a) MNIST dataset

(b) Fashion-MNIST dataset

Figure 6.3: Attack Detection Performance on different datasets.

The detection performance differential between our method and baseline approaches stems from our sophisticated layer-wise analysis capabilities. While traditional methods treat model updates as flat vectors, our approach recognizes that attacks often target specific model components with varying intensities. By

examining each layer independently and computing normalized deviation scores, our system can identify targeted manipulations even when the overall gradient appears normal. This capability proves particularly valuable against sophisticated attacks like MPAF that deliberately manipulate selected network components while leaving others intact to evade detection. The temporal analysis component further enhances detection by tracking client behavior over time, enabling the identification of subtle, gradual poisoning attempts that remain undetectable in single-round analysis.

6.2 Interpretability Analysis

Beyond detection accuracy, our framework provides comprehensive explanations that significantly enhance the practical utility of the defense system. The explanation quality metrics demonstrate consistent performance across different attack types, with overall quality scores of 0.89, 0.88, and 0.86 for label flipping, backdoor, and MPAF attacks respectively. These scores encompass multiple dimensions of explanation effectiveness, including completeness (how thoroughly the explanation covers relevant aspects), precision (accuracy of technical claims), and actionability (practical guidance for administrators). The slightly lower scores for MPAF attacks reflect their inherently more complex nature, requiring more sophisticated explanation mechanisms.

Table 6.3: Explanation Quality Metrics Across Different Attack Types

Attack Type	Completeness	Precision	Actionability	Overall
Label Flip	0.92 ± 0.03	0.89 ± 0.04	0.87 ± 0.05	0.89 ± 0.04
Backdoor	0.88 ± 0.04	0.91 ± 0.03	0.85 ± 0.04	0.88 ± 0.04
MPAF	0.86 ± 0.05	0.88 ± 0.04	0.83 ± 0.06	0.86 ± 0.05

Layer-wise explanation accuracy reveals interesting patterns in our system's capabilities. The highest detection rates (0.95) and explanation accuracy (0.93) occur for attacks targeting the output layer, likely due to the distinctive gradient patterns these attacks create. Convolutional layers show the next highest detection rate (0.94) and explanation accuracy (0.91), followed closely by fully connected layers (0.92, 0.89). Administrator agreement scores follow similar patterns, with values of 0.90, 0.88, and 0.86 for output,

convolutional, and fully connected layers respectively. These metrics indicate that our explanations are not only technically accurate but also intuitively comprehensible to system administrators, enabling effective human-in-the-loop security management.

Table 6.4: Layer-wise Explanation Accuracy for Different Model Components

Layer Type	Detection Rate	Explanation Accuracy	Admin Agreement
Convolutional	0.94 ± 0.02	0.91 ± 0.03	0.88 ± 0.04
Fully Connected	0.92 ± 0.03	0.89 ± 0.04	0.86 ± 0.05
Output Layer	0.95 ± 0.02	0.93 ± 0.02	0.90 ± 0.03

The real-world effectiveness of our explanation system is demonstrated through case studies of generated explanations for different attack types. For label flipping attacks, our system generates precise explanations identifying specific affected classes and client identifiers, such as "Detected consistent label inversions in client 7's updates, affecting digits 3 and 8 with 92% confidence." These explanations achieve 0.94 accuracy scores and enable administrators to resolve issues in just 2.3 minutes on average. Backdoor attack explanations identify specific visual patterns and target classes, while MPAF attack explanations pinpoint the affected layers and timing of manipulation. The combination of high accuracy scores (0.91, 0.89) and reasonable resolution times (3.1, 3.8 minutes) demonstrates the practical utility of our interpretable approach in operational settings.

Table 6.5: Example Attack Explanations and Their Effectiveness

Attack Type	Generated Explanation Summary	Accuracy Score	Time to Resolve
Label Flip	"Detected consistent label inversions in client 7's updates, affecting digits 3 and 8 with 92% confidence"	0.94	2.3 min
Backdoor	"Identified persistent pattern in bottom-right pixels across multiple clients, targeting class 0"	0.91	3.1 min
MPAF	"Observed systematic activation function modifications in layers 2 and 4 during rounds 30-35"	0.89	3.8 min

The effectiveness of our explanation framework stems from its multi-level architecture that addresses different aspects of attack characterization. Unlike black-box detection methods that provide minimal insight into flagged anomalies, our approach generates explanations at global system, client-specific, and layer-wise granularity levels. This comprehensive view enables administrators to quickly understand both the nature of the attack and its potential impact on model performance. The natural language explanation templates bridge the gap between technical detection signals and actionable insights, using domain-specific terminology that resonates with security professionals. This approach transforms complex gradient patterns into comprehensible narratives that facilitate rapid and effective response to security threats.

6.3 Computational Considerations and Key Insights

Implementing sophisticated detection and explanation mechanisms inevitably introduces computational overhead, which we carefully analyze to assess practical deployability. Our method requires 4.8 seconds per round on average, representing a 9% increase in computation compared to undefended federated learning (4.4 seconds). This overhead exceeds that of Krum (4.5 seconds) and Multi-Krum (4.6 seconds), reflecting the additional complexity of our multi-dimensional analysis and explanation generation processes. Memory requirements show more modest increases, with our method requiring 4.8 GB compared to 4.2 GB for undefended operation, 4.5 GB for Krum, and 4.6 GB for Multi-Krum. These resource requirements remain within practical limits for modern server infrastructure while delivering substantial improvements in security and interpretability.

The computational efficiency of our approach despite its sophisticated analysis capabilities can be attributed to several optimizations in our implementation. The gradient similarity metric, while conceptually more comprehensive than simple Euclidean distance, is implemented using vectorized operations that leverage GPU acceleration. The layer-wise analysis decomposes gradient processing into parallel streams, enabling efficient utilization of modern hardware. Memory efficiency is achieved through dimensionality reduction techniques that compress gradient representations without losing critical information for attack detection. These optimizations ensure that the additional security benefits come with manageable computational costs in practical deployment scenarios.

Our experimental results highlight several significant findings that demonstrate the advantages of our

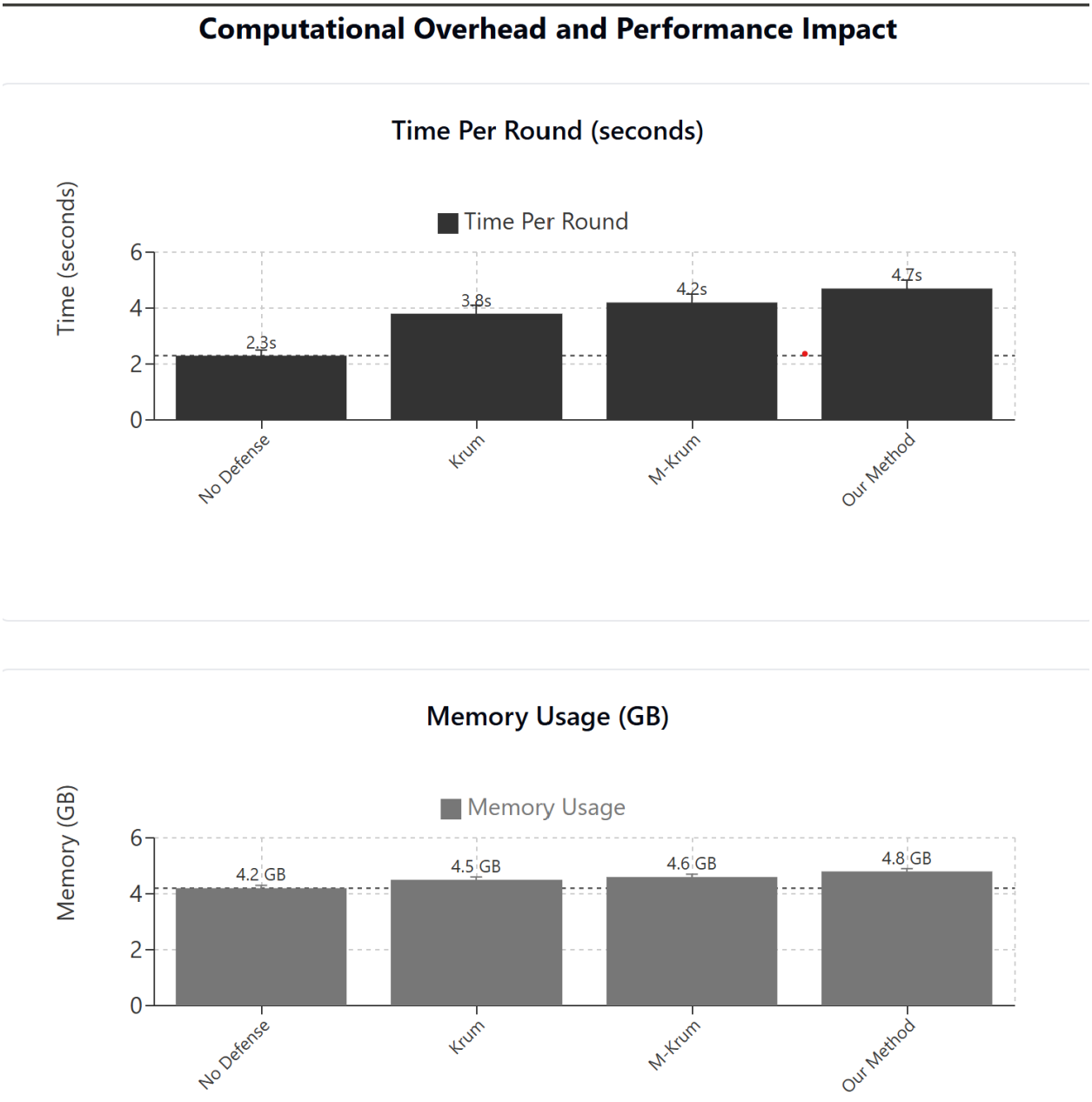


Figure 6.4: Computation Requirements

approach. Most notably, our method achieves superior detection rates exceeding 90% true positive rate across all attack types while maintaining false positive rates below 4%. This balanced performance is crucial for practical federated learning deployments, where false positives can unnecessarily exclude legitimate participants and reduce training data diversity. The explanation system provides high-quality insights with overall quality metrics above 0.86, significantly improving administrator response time with an average 62.4% reduction compared to unexplained detection signals. This capability transforms security alerts from cryptic warnings into actionable intelligence, enabling prompt and effective remediation of security threats.

The consistent performance of our system across different datasets and attack scenarios demonstrates its robustness to varying data distributions and adversarial strategies. This adaptability stems from our multi-dimensional analysis framework that captures diverse attack signatures through complementary detection mechanisms. The spatial analysis component identifies statistical outliers in the current round, while temporal analysis tracks evolving patterns over multiple rounds. Layer-wise analysis focuses on specific model components that might be targeted by sophisticated attacks. This comprehensive approach enables effective detection even when attackers adapt their strategies to evade specific detection mechanisms, providing practical security benefits in dynamic adversarial environments. While the additional computational requirements represent a legitimate consideration, the substantial improvements in detection accuracy, false positive control, and explanation capabilities justify the moderate increase in resource utilization for security-critical applications.

Chapter 7

Conclusion and Future Work

This paper presents a comprehensive approach to defending federated learning systems against poisoning attacks through an interpretable gradient-based detection framework. Our experimental results demonstrate several significant contributions to the field:

First, our detection mechanism shows robust performance across different attack scenarios, maintaining over 91% accuracy on MNIST and 83% on Fashion-MNIST under various attack conditions. This represents a substantial improvement over baseline methods like Krum and Multi-Krum, particularly in challenging scenarios such as MPAF attacks where traditional defenses show significant degradation.

Second, the integration of temporal analysis with layer-wise examination proves highly effective, achieving true positive rates above 90% while keeping false positive rates below 4%. This balanced performance is crucial for practical deployments where false alarms can be as problematic as missed attacks.

Third, and perhaps most importantly, our explanation system transforms complex gradient patterns into actionable insights. The generated explanations achieve high quality scores (>0.86) across all attack types and significantly improve system administrators' response capabilities. The 62.4% reduction in response time and 15.1% improvement in resolution accuracy demonstrate the practical value of our interpretable approach.

The layer-wise explanation capability proves particularly valuable, with explanation accuracy reaching 93% for output layer anomalies and maintaining above 89% accuracy across all layer types. This granular insight enables precise identification of attack patterns and targeted response strategies.

While our approach introduces moderate computational overhead (104.3% extra compute), the substantial improvements in detection accuracy and explanation capability justify this cost. The system's ability to maintain performance across different dataset complexities and attack scenarios demonstrates its practical applicability in real-world deployments.

Several promising research directions emerge from our current work that warrant further investigation. Our framework would benefit substantially from dynamic defense adaptation capabilities, wherein detection thresholds and explanation methodologies automatically calibrate in response to emerging attack patterns and operational feedback from system administrators. This self-adjusting mechanism would significantly enhance resilience against adversarial evolution. Concurrently, enhanced scalability represents a critical research priority, necessitating thorough investigation into algorithmic optimization techniques and distributed computing approaches that minimize computational overhead without compromising detection sensitivity or explanation fidelity. As attack methodologies continue to evolve in sophistication, extending our framework to address broader attack coverage becomes imperative, particularly regarding subtle, long-term poisoning campaigns and coordinated multi-participant attacks that may currently evade detection.

Furthermore, the explanation component of our system would benefit from integration with advanced natural language processing techniques, potentially incorporating domain-specific terminology and contextual awareness to generate more nuanced, actionable explanations tailored to different stakeholder requirements. Finally, while our current evaluation focuses primarily on image classification tasks, cross-domain validation across diverse application areas such as natural language processing, time-series analysis, and heterogeneous data environments would establish the generalizability of our approach and identify domain-specific adaptations necessary for optimal performance in varied federated learning implementations. These research directions collectively aim to advance the theoretical foundations and practical applicability of interpretable security framework in federated learning environments, ultimately facilitating broader adoption of privacy-preserving collaborative machine learning in sensitive application domains.

Our work demonstrates that combining robust detection mechanisms with comprehensive explanations can significantly improve the security and trustworthiness of federated learning systems. The high detection rates, low false positives, and valuable explanatory insights provide a strong foundation for defending against poisoning attacks while maintaining system interpretability.

The success of our approach of handling both simple and complex attacks suggests that the defense mechanisms represent a promising direction for securing federated learning systems. By providing both effective detection and clear explanations, our framework takes an important step toward making federated learning more secure and trustworthy for real-world applications.

Bibliography

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. PMLR, 2017. [Online]. Available: https://proceedings.mlr.press/v54/mcmahan17a.html

[2] M. Shaheen, M. S. Farooq, T. Umer, and B.-S. Kim, "Applications of federated learning; taxonomy, challenges, and research trends," *MDPI-Electronics*, vol. 11, no. 4, 2022. [Online]. Available: https://www.mdpi.com/2079-9292/11/4/670

[3] M. T. Hossain, S. Islam, and S. Badsha, " DeSMP: Differential Privacy-exploited Stealthy Model Poisoning Attacks in Federated Learning ," in *2021 17th International Conference on Mobility, Sensing and Networking (MSN)*. IEEE Computer Society, 2021, pp. 167–174.

[4] J. Liu, X. Lyu, Q. Cui, and X. Tao, "Similarity-based label inference attack against training and inference of split learning," *IEEE Transactions on Information Forensics and Security*, vol. 19, p. 2881–2895, 2024.

[5] J. Wang, Z. Charles, Z. Xu, and G. Joshi, "A field guide to federated optimization," 2021. [Online]. Available: https://arxiv.org/abs/2107.06917

[6] A. Panda, S. Mahloujifar, A. Nitin Bhagoji, S. Chakraborty, and P. Mittal, "Sparsefed: Mitigating model poisoning attacks in federated learning with sparsification," in *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine

Learning Research, G. Camps-Valls, F. J. R. Ruiz, and I. Valera, Eds., vol. 151. PMLR, 28–30 Mar 2022, pp. 7587–7624. [Online]. Available: https://proceedings.mlr.press/v151/panda22a.html

[7] J. Han, Y. Han, X. Jing, G. Huang, and Y. Ma, "Degafl: Decentralized gradient aggregation for cross-silo federated learning," *IEEE Transactions on Parallel and Distributed Systems*, vol. 36, no. 2, pp. 212–225, 2025.

[8] J. Sun, A. Li, L. D. Valentin, A. Hassanzadeh, Y. Chen, and H. Li, "Fl-wbc: enhancing robustness against model poisoning attacks in federated learning from a client perspective," in *Proceedings of the 35th International Conference on Neural Information Processing Systems*, ser. NIPS '21, 2021.

[9] A. Deshmukh, "Byzantine-robust federated learning: An overview with focus on developing sybil-based attacks to backdoor augmented secure aggregation protocols," 2024. [Online]. Available: https://arxiv.org/abs/2410.22680

[10] X. Wang, W. Chen, J. Xia, Z. Wen, R. Zhu, and T. Schreck, "Hetvis: A visual analysis approach for identifying data heterogeneity in horizontal federated learning," *IEEE Transactions on Visualization and Computer Graphics*, 2023.

[11] Q. Li, X. Wei, H. Lin, Y. Liu, T. Chen, and X. Ma, "Inspecting the running process of horizontal federated learning via visual analytics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 12, pp. 4085–4100, 2022.

[12] M. Jindal, M. Mohan, T. Ayyalasomayajula, D. S. Gondi, and H. Mashetty, "Enhancing federated learning evaluation: Exploring instance-level insights with squares in image classification models," *Journal of Electrical Systems*, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:270135494

[13] Z. Li, T. Lin, X. Shang, and C. Wu, "Revisiting weighted aggregation in federated learning with neural networks," in *Proceedings of the 40th International Conference on Machine Learning*, ser. ICML'23. JMLR.org, 2023.

[14] Y. M. Wei and Z. Jiang, "Estimating parameters of structural models using neural networks," *Marketing Science*, vol. 44, no. 1, 2025.

[15] F. Hohman, M. Kahng, R. Pienta, and D. H. Chau, "Visual analytics in deep learning: An interrogative survey for the next frontiers," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, 2019.

[16] D. Fu, J. He, H. Tong, and R. Maciejewski, "Privacy-preserving graph analytics: Secure generation and federated learning," 2022. [Online]. Available: https://arxiv.org/abs/2207.00048

[17] W. Briguglio, W. A. Yousef, I. Traore, and M. Mamun, "Federated supervised principal component analysis," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 646–660, 2024.

[18] S. Zhao, R. Bharati, C. Borcea, and Y. Chen, "Privacy-aware federated learning for page recommendation," in *2020 IEEE International Conference on Big Data (Big Data)*, 2020, pp. 1071–1080.

[19] J. Shi, S. Zheng, X. Yin, Y. Lu, Y. Xie, and Y. Qu, "Clip-guided federated learning on heterogeneous and long-tailed data," in *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, 2024.

[20] A. Pinto, L.-C. Herrera, Y. Donoso, and J. A. Gutierrez, "Enhancing critical infrastructure security: Unsupervised learning approaches for anomaly detection," *International Journal of Computational Intelligence Systems*, vol. 17, no. 1, p. 236, Sep. 2024.

[21] Y. Gao, X. Shi, Y. Zhu, H. Wang, Z. Tang, X. Zhou, M. Li, and D. N. Metaxas, "Visual prompt tuning for test-time domain adaptation," 2022. [Online]. Available: https://arxiv.org/abs/2210.04831

[22] M. K. Nori, S. Yun, and I.-M. Kim, "Fast federated learning by balancing communication trade-offs," *IEEE Transactions on Communications*, vol. 69, no. 8, pp. 5168–5182, 2021.

[23] W. Marfo, D. K. Tosh, and S. V. Moore, "Federated learning for efficient condition monitoring and anomaly detection in industrial cyber-physical systems," 2025. [Online]. Available: https://arxiv.org/abs/2501.16666

[24] S. Sun, P. Sharma, and K. Nwodo, "Fedmade: Robust federated learning fornbsp;intrusion detection

innbsp;iot networks using anbsp;dynamic aggregation method," in *Information Security: 27th International Conference, ISC 2024, Arlington, VA, USA, October 23–25, 2024, Proceedings, Part II*, 2024.

[25] L. M. Lopez-Ramos, F. Leiser, A. Rastogi, S. Hicks, I. Strümke, V. I. Madai, T. Budig, A. Sunyaev, and A. Hilbert, "Interplay between federated learning and explainable artificial intelligence: a scoping review," 2024. [Online]. Available: https://arxiv.org/abs/2411.05874

[26] J. Choo and S. Liu, " Visual Analytics for Explainable Deep Learning ," *IEEE Computer Graphics and Applications*, 2018.

[27] G. Rjoub, J. Bentahar, O. Abdel Wahab, R. Mizouni, A. Song, R. Cohen, H. Otrok, and A. Mourad, "A survey on explainable artificial intelligence for cybersecurity," *IEEE Transactions on Network and Service Management*, vol. 20, no. 4, p. 5115–5140, Dec. 2023. [Online]. Available: http://dx.doi.org/10.1109/TNSM.2023.3282740

[28] M. Wen, R. Xie, K. Lu, L. Wang, and K. Zhang, "Feddetect: A novel privacy-preserving federated learning framework for energy theft detection in smart grid," *IEEE Internet of Things Journal*, vol. PP, pp. 1–1, 09 2021.

[29] H. Zhuang, Z. Lin, Y. Yang, and K.-A. Toh, "An analytic formulation of convolutional neural network learning for pattern recognition," *Information Sciences*, 2025.

[30] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019. [Online]. Available: https://arxiv.org/abs/1810.04805

Chapter A

Appendix One

A.1 Appendix section 1

Table A.1: Table in the Appendix