Article

Not peer-reviewed version

Time-Aware Security Intelligence for Federated Financial Systems: Deep Reinforcement Learning Against Temporal Poisoning Attacks

Wenan Liu , Qixuan Yang * , Weihang Gong , Rongji Yin , Zheng Li

Posted Date: 2 October 2025

doi: 10.20944/preprints202510.0141.v1

Keywords: federated learning; temporal backdoor attacks; financial security; deep reinforcement learning; Byzantine robustness; Markov decision process; geometric median aggregation; adaptive defense coordination; multi-layer security; temporal behavior analysis

 doi:10.20944/preprints202510.0141.v1

Article

Time-Aware Security Intelligence for Federated Financial Systems: Deep Reinforcement Learning Against Temporal Poisoning Attacks

Wenan Liu [1], Qixuan Yang [2,*], Weihang Gong [3], Rongji Yin [4] and Zheng Li [5]

[1] Faculty of Business, City University of Macau, Macau 999078, China
[2] Antai College of Economics and Management, Shanghai Jiao Tong University, Shanghai 200030, China
[3] College of Economics and Management, Qiqihar University, Qiqihar 161006, China
[4] College of Mechanical and Electronic Engineering, Shanghai Jian Qiao University, Shanghai 201306, China
[5] College of Electronic Information Engineering, Taiyuan University of Technology, Taiyuan 030024, China
* Correspondence: yangqixuan125@sjtu.edu.cn

Abstract

Financial institutions operating distributed machine learning systems face an emerging class of stealth adversaries who exploit temporal patterns across training cycles to inject persistent backdoors that remain dormant for months before activation. Unlike conventional single-round attacks, these sophisticated temporal poisoning strategies leverage sequential dependencies to bypass existing detection mechanisms while gradually compromising model integrity. Current defense frameworks remain fundamentally inadequate against such multi-period adversarial choreography, particularly in high-stakes financial environments where minute perturbations can trigger systemic failures. Existing security frameworks largely focus on static threat models and fail to address sophisticated multi-period adversarial strategies that unfold over time in financial transaction streams. To address these challenges, we propose DEFEND, a comprehensive defense framework that integrates temporal behavior analysis, robust statistical aggregation, and multi-scale verification into a unified multi-layer architecture. Our framework formulates defense coordination as a Markov Decision Process and employs Proximal Policy Optimization for adaptive policy learning that dynamically balances security enforcement with model utility. We design three sophisticated temporal attack models to comprehensively evaluate our defense mechanism: fixed-period data poisoning, multi-period data poisoning, and model weight poisoning attacks. The multi-layer defense architecture combines geometric median-based robust aggregation with Dynamic Time Warping pattern matching and adaptive client participation control. Extensive experiments on CIFAR-10, FEMNIST, and MNIST datasets demonstrate that DEFEND achieves superior defense performance with success rates of 95.6% for ResNet-18 and 94.0% for MobileNet V2, while maintaining clean accuracy levels between 85-95% across various data heterogeneity levels and malicious client ratios. Our framework provides theoretical guarantees for Byzantine robustness and practical scalability for moderate-scale federated deployments, making it well-suited for real-world financial applications requiring both security and efficiency.

Keywords: federated learning; temporal backdoor attacks; financial security; deep reinforcement learning; Byzantine robustness; Markov decision process; geometric median aggregation; adaptive defense coordination; multi-layer security; temporal behavior analysis

1. Introduction

1.1. Background

The financial industry is undergoing rapid digital transformation, with institutions increasingly relying on distributed machine learning techniques to improve fraud detection, risk management, and regulatory compliance. Federated learning (FL) has emerged as a promising paradigm to enable

 doi:10.20944/preprints202510.0141.v1

collaborative intelligence across banks, payment platforms, and other financial entities while maintaining data privacy [1–3]. By allowing participants to train shared models without centralizing raw data, FL provides a natural fit for sensitive financial applications where strict privacy regulations and competitive concerns prevent data sharing.

However, the adoption of FL in financial systems introduces significant security challenges. Recent studies highlight that FL is inherently vulnerable to poisoning and backdoor attacks, where malicious participants inject crafted updates to manipulate global models [4,5]. These attacks can be particularly damaging in financial environments, where small perturbations may lead to large-scale fraud or systemic risks. Beyond traditional single-round threats, researchers have uncovered persistent backdoor strategies that exploit temporal dependencies across multiple training rounds, allowing attackers to evade conventional defenses and achieve long-term stealth [6,7]. Such temporal vulnerabilities are especially critical in financial transaction streams, which naturally exhibit sequential correlations and evolving patterns.

To address these challenges, researchers have proposed robust aggregation methods and privacy-enhancing techniques to mitigate adversarial updates in FL. Approaches such as robust learning rate adjustment [8–10] and privacy-preserving backdoor defenses for heterogeneous data distributions [11] have demonstrated effectiveness under certain threat models. Nevertheless, most of these methods remain static and fail to capture multi-period adversarial strategies that unfold over time. Moreover, existing solutions often trade accuracy for security, limiting their practicality in high-stakes domains like finance where both precision and reliability are paramount.

Meanwhile, reinforcement learning (RL) has shown considerable potential for adaptive cybersecurity, offering dynamic decision-making in the face of evolving adversarial behaviors [12]. By framing defense coordination as a sequential decision problem, RL methods can enable financial FL systems to respond adaptively to temporal threats, balancing security enforcement with model utility. This motivates the development of integrated frameworks that combine multi-layered defenses with RL-based coordination to address sophisticated temporal backdoor attacks in financial federated learning environments.

1.2. Motivation and Contributions

Despite growing efforts to enhance the robustness of federated learning, several critical research gaps remain unaddressed in the context of financial systems:

1. Existing security frameworks do not adequately account for **temporal backdoor attacks** that exploit sequential dependencies in financial transaction streams. Most defenses assume static or isolated threat models, overlooking coordinated multi-round adversarial strategies.
2. Current defense mechanisms are largely **single-layered** and static, focusing either on aggregation robustness or anomaly detection, without integrating multiple complementary layers that can collectively enhance resilience against adaptive attackers.
3. Few approaches incorporate **dynamic coordination mechanisms** to balance security and utility. Static thresholds or fixed strategies cannot adapt to changing adversarial intensity, heterogeneous client behaviors, and evolving network conditions.

To address these gaps, this work introduces *DEFEND*, a comprehensive defense framework for federated learning in financial systems. Our framework integrates temporal behavior analysis, robust statistical aggregation, and multi-scale verification into a multi-layered architecture, while employing a Markov Decision Process (MDP) formulation and reinforcement learning for adaptive coordination. The contributions of this paper are summarized as follows:

- We formalize **temporal backdoor threats** in federated financial learning by characterizing attack strategies that exploit multi-period dependencies across training rounds.
- We design a **multi-layer defense architecture** that combines temporal behavior profiling, robust aggregation, and multi-scale verification to jointly enhance detection accuracy and resilience.

 doi:10.20944/preprints202510.0141.v1

- We formulate defense coordination as an **MDP problem** and develop an RL-based policy using Proximal Policy Optimization (PPO) to dynamically manage defense actions, balancing robustness and model performance.
- We validate *DEFEND* on multiple benchmark datasets (CIFAR-10, FEMNIST, MNIST) under varying degrees of heterogeneity and adversarial participation, demonstrating superior defense success rates and detection efficiency compared to state-of-the-art baselines.

The remainder of this paper is organized as follows. Section 2 reviews existing research on federated learning security, temporal attack detection, and multi-layer defense coordination. Section 3 describes the proposed *DEFEND* framework. Section 4 presents experimental evaluations. Section 6 concludes and outlines directions for future work.

2. Related Work

2.1. Federated Learning Security in Financial Systems

Federated learning security in financial systems has emerged as a critical research area addressing the inherent privacy and security challenges in distributed financial data processing environments [13–21]. Chen *et al.* [13] conducted an extensive survey identifying the intricate security challenges within federated learning frameworks, emphasizing vulnerabilities in communication links and potential cyber threats across decentralized networks. Their comprehensive analysis delves into various defensive strategies and explores applications across different sectors, contributing to the development of secure and efficient federated learning systems. To address specific financial fraud detection challenges, Aljunaid *et al.* [14] proposed an Explainable Federated Learning (XFL) model that integrates Shapley Additive Explanations (SHAP) and LIME techniques for enhanced interpretability while maintaining privacy compliance. Their approach achieved 99.95% accuracy with a miss rate of 0.05%, effectively eliminating false positives in financial fraud classification while preserving data privacy and regulatory compliance.

Building on decentralized security considerations, Hallaji *et al.* [15] performed a thorough security analysis of decentralized federated learning systems, studying possible variations of threats and adversaries while overviewing potential defense mechanisms. Their work addresses server-related threats elimination through blockchain technologies, though acknowledging new privacy challenges introduced by decentralized architectures. To enhance privacy preservation in financial technology applications, Xiong *et al.* [20] developed a Heterogeneous Privacy-Preserving Blockchain-Enabled Federated Learning (HPP-BEFL) system specifically designed for social fintech environments. Their novel PKI and identity-based heterogeneous authenticated asymmetric group key agreement (PKI-IB-HAAGKA) protocol effectively mitigates man-in-the-middle and inference attacks while addressing crypto system heterogeneity issues.

Recent advances have also explored credit risk assessment architectures [16], behavioral anomaly detection in dynamic transaction graphs [17], and blockchain-based knowledge enhancement mechanisms [22]. However, existing federated learning security frameworks lack comprehensive temporal attack detection mechanisms and fail to adequately address sophisticated backdoor attacks that exploit temporal patterns in financial data streams, which are essential for defending against coordinated multi-period adversarial strategies in distributed financial learning environments.

 doi:10.20944/preprints202510.0141.v1

Table 1. Comparison of our work with related studies.

Feature \ Ref	[13]	[23]	[15]	[20]	[24]	[25]	[26]	[27]	[14]	[28]	[29]	Proposed work
Financial application domain				✓					✓			✓
Federated learning framework	✓	✓	✓	✓				✓	✓	✓		✓
Temporal attack detection					✓	✓						✓
Multi-layer defense design										✓		✓
MDP/RL coordination							✓				✓	✓

2.2. Temporal Attack Detection and Defense Mechanisms

Temporal attack detection and defense mechanisms have gained significant attention as adversaries increasingly exploit time-dependent vulnerabilities in distributed systems [24,25,30–35]. Zamanzadeh *et al.* [25] provided a comprehensive survey of deep learning approaches for time series anomaly detection, highlighting the importance of identifying anomalous patterns that indicate novel or unexpected events such as production faults and system defects. Their taxonomy encompasses anomaly detection strategies and deep learning models, highlighting the challenges presented by the large size and complexity of temporal patterns in time series data. Duan *et al.* [24] addressed practical cyber attack detection through Continuous Temporal Graph (CTG) neural networks in dynamic network systems, proposing an interaction-centered perspective that refines information interactions between network entities into CTG evolution processes. Their framework naturally incorporates new node access behaviors and presents a message aggregation scheme that fuses spatio-temporal neighborhoods with actual time distribution and historical states, demonstrating superior performance on ToN-IoT, UNSWNB15, CIC-Dark2020, and J.P. Morgan payment datasets.

To address zero-day attack challenges, Wu *et al.* [31] developed an active learning framework using Deep Q-Network (DQN) for intelligent sample selection with probability distribution analysis. Their approach integrates Bi-directional Long Short-Term Memory (BiLSTM) networks into the DQN model to analyze temporal correlations within static classification contexts, employing Euclidean distance functions for accurate sample labeling. Hammad *et al.* [33] explored deep reinforcement learning for adaptive cyber defense, implementing cutting-edge DRL techniques including Deep Q-Networks (DQN), Proximal Policy Optimization (PPO), and Twin Delayed Deep Deterministic Policy Gradient (TD3) for real-time threat discernment and neutralization across varied cyber threat scenarios ranging from malware invasions to phishing attacks and adversarial assaults.

Recent developments have also investigated spatio-temporal advanced persistent threat detection in cyber-physical power systems [30], advances in time-series anomaly detection algorithms and benchmarks [36], and security defense strategies for Internet of Things based on deep reinforcement learning [37]. However, existing temporal attack detection frameworks lack sophisticated multi-period pattern recognition capabilities and fail to address coordinated backdoor injection strategies that exploit temporal dependencies across distributed learning rounds, which are crucial for detecting and defending against sophisticated temporal poisoning attacks in federated financial learning environments.

2.3. Multi-Layer Defense and MDP-based Coordination

Multi-layer defense mechanisms and MDP-based coordination strategies have emerged as critical components for robust federated learning systems, addressing the need for comprehensive protection

 doi:10.20944/preprints202510.0141.v1

against sophisticated adversarial attacks [23,26–29,38–42]. Li *et al.* [28] proposed a Multi-layer Aggregation Backdoor Defense Framework (MABDF) that ensures secure model aggregation through adaptive similarity filtering, pruning mean aggregation, and subspace robust projection methods. Their three-layer architecture calculates pairwise cosine similarity between client updates with dynamic thresholds based on median and standard deviation, applies pruning mean aggregation to detect hidden gradient operations, and projects updates onto low-rank subspaces through singular value decomposition (SVD) to suppress backdoor neuron activation, achieving backdoor attack success rates below 3% while maintaining model accuracy decrease of no more than 1.5%. Uddin *et al.* [23] conducted a systematic literature review using the PRISMA framework, analyzing 244 studies across eight themes of robust federated learning including objective regularization, optimizer modification, differential privacy, client selection, and new aggregation algorithms, providing comprehensive insights into approaches for enhancing FL model robustness against adversarial attacks and noisy updates.

For dynamic policy coordination, Bello *et al.* [26] developed a security strategy incorporating zero-trust models with dynamic policy decisions through stochastic games and reinforcement learning techniques. Their approach employs Generalized Proximal Policy Optimization with sample reuse (GePPO) and its meta-learning variant GePPO-ML, along with Sample Dropout PPO with meta-learning (SDPPO-ML) for adaptive policy updates, demonstrating superior performance compared to baseline REINFORCE and PPO algorithms in next generation network security scenarios. Huang *et al.* [27] addressed Byzantine robustness in heterogeneous federated learning through Self-Driven Entropy Aggregation (SDEA), leveraging random public datasets to conduct robust aggregation by introducing learnable aggregation weights that minimize instance-prediction entropy while maximizing batch-prediction entropy to accommodate diverse client tendencies and detect Byzantine attackers effectively.

Recent advances have also explored multi-layered protection systems for cloud computing environments [39], safe reinforcement learning frameworks for risk-averse dispatch with frequency security constraints [40], and security metrics for assessing power grids against attacks from EV charging ecosystems using Markov decision processes [29,43]. However, existing multi-layer defense frameworks lack integrated temporal anomaly detection capabilities and fail to provide coordinated MDP-based defense mechanisms that can dynamically adapt to evolving temporal backdoor attack patterns while maintaining optimal resource allocation and system performance in distributed financial learning environments.

3. Method

This section presents our comprehensive framework for detecting and defending against temporal backdoor attacks in federated learning environments through sophisticated mathematical modeling approaches. Our methodology integrates three distinct temporal poisoning attack models, establishes a multi-objective optimization framework, employs advanced statistical analysis techniques, and formulates an MDP-based defense mechanism. The proposed system leverages deep mathematical foundations to achieve optimal detection and defense performance across diverse federated learning scenarios while maintaining computational efficiency and theoretical rigor. Method architecture is shown in the figure 1.

 doi:10.20944/preprints202510.0141.v1

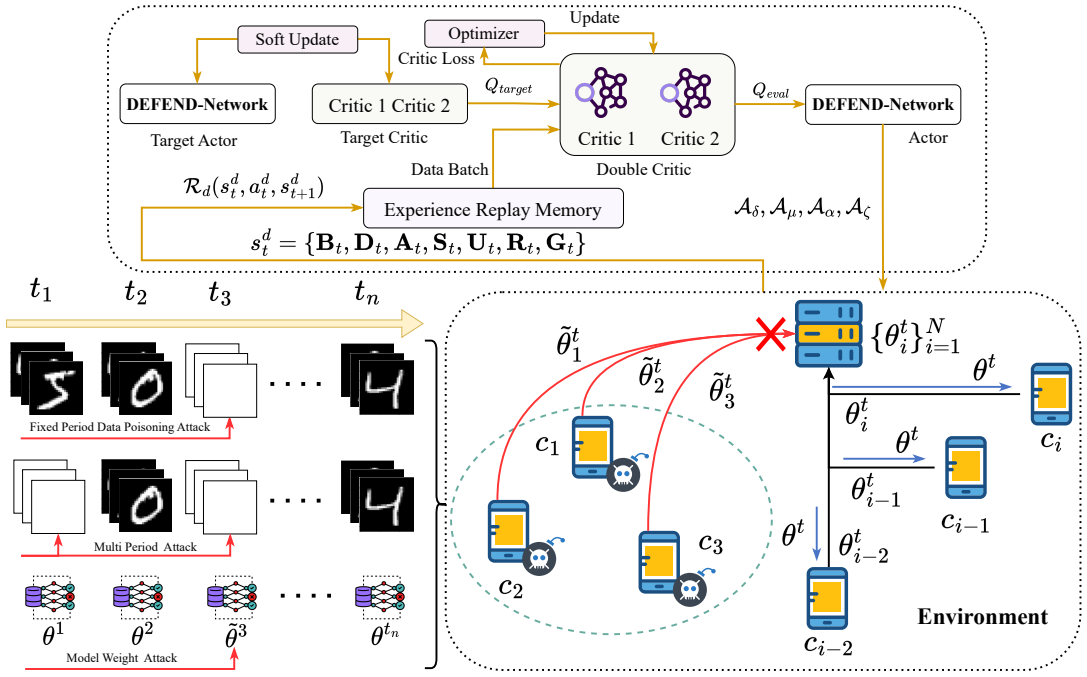


Figure 1. DEFEND Framework.

3.1. Problem Formulation

Consider a federated learning system comprising N participating clients denoted as $\mathcal{C} = \{c_1, c_2, \ldots, c_N\}$, where each client c_i maintains local dataset $\mathcal{D}_i = \{(x_{i,j}, y_{i,j})\}_{j=1}^{n_i}$ over a finite communication horizon T. The temporal data sequence for client c_i is represented as $\mathbf{X}_i = \{x_i^1, x_i^2, \ldots, x_i^T\}$, where $x_i^t \in \mathbb{R}^m$ denotes the m-dimensional feature vector observed at communication round t. The corresponding ground truth labels are denoted as $\mathbf{y}_i = \{y_i^1, y_i^2, \ldots, y_i^T\}$, where $y_i^t \in \mathcal{Y}$ represents the true class label at round t from the label space $\mathcal{Y}$.

The client operational states are characterized by a discrete state space $\mathcal{S} = \{s_n, s_s, s_p\}$, where s_n represents the normal operational state, s_s indicates a suspicious state requiring further monitoring, and s_p denotes a confirmed poisoned state. During the poisoning process, a subset of clients $\mathcal{C}_p \subset \mathcal{C}$ with cardinality $|\mathcal{C}_p| \leq \kappa N$, where $\kappa \in (0, 1)$ represents the maximum poisoning ratio, becomes compromised during specific time intervals $\mathcal{T}_p \subset \{1, 2, \ldots, T\}$.

Each client c_i is characterized by a resource profile $\mathbf{R}_i = (C_i^{\max}, M_i^{\max}, E_i^{\max}, B_i^{\max})$, where $C_i^{\max}$ represents maximum computational capacity, $M_i^{\max}$ denotes memory capacity, $E_i^{\max}$ indicates energy budget, and $B_i^{\max}$ specifies communication bandwidth. The available resources at round t are denoted as $\mathbf{R}_i^t = (C_i^t, M_i^t, E_i^t, B_i^t)$, where each component satisfies $0 \leq \mathbf{R}_i^t \leq \mathbf{R}_i$.

The fundamental objective of our detection framework is to minimize the overall detection error, formulated as a multi-objective optimization problem with temporal consistency constraints:

$$\mathcal{L}_\tau = \sum_{i=1}^{N} \sum_{t=1}^{T} \ell_d(\hat{s}_i^t, s_i^t) + \lambda_1 \sum_{i \in \mathcal{C}_p} \ell_t(\hat{\mathcal{T}}_{p,i}, \mathcal{T}_{p,i}) + \lambda_2 R_{f,p} + \lambda_3 \int_0^T \mathcal{C}(u) du + \lambda_4 \mathcal{H}(\mathbf{s}_t), \tag{1}$$

where $\ell_d : \mathcal{S} \times \mathcal{S} \to \mathbb{R}_{\geq 0}$ represents the client state classification loss function, $\hat{s}_i^t$ denotes the predicted client state, ℓ_t quantifies the temporal localization error between predicted intervals $\hat{\mathcal{T}}_{p,i}$ and true poisoning intervals $\mathcal{T}_{p,i}$, $R_{f,p}$ denotes the false positive penalty term, $\mathcal{C}(u)$ is the continuous cost function modeling resource consumption, $\mathcal{H}(\mathbf{s}_t)$ represents the entropy regularization term for uncertainty quantification, and $\lambda_1, \lambda_2, \lambda_3, \lambda_4 \in \mathbb{R}_{>0}$ are regularization coefficients that balance different objectives.

 doi:10.20944/preprints202510.0141.v1

The temporal localization error is computed using a weighted Hausdorff distance:

$$\ell_t(\hat{\mathcal{T}}_{p,i}, \mathcal{T}_{p,i}) = \frac{1}{2}\left[\max_{t\in\hat{\mathcal{T}}_{p,i}} \min_{\tau\in\mathcal{T}_{p,i}} |t-\tau| + \max_{\tau\in\mathcal{T}_{p,i}} \min_{t\in\hat{\mathcal{T}}_{p,i}} |t-\tau|\right], \tag{2}$$

which ensures both precision and recall in temporal poisoning interval detection.

3.2. Temporal Backdoor Attack Models

To comprehensively evaluate our detection framework, we design three sophisticated temporal backdoor strategies that exploit different characteristics of federated learning protocols. Each attack model represents a distinct threat vector commonly encountered in real-world federated environments, incorporating realistic constraints on attack capabilities and detectability thresholds.

3.2.1. Fixed-Period Data Poisoning Attack

The fixed-period data poisoning attack systematically injects backdoor triggers into client datasets during predetermined temporal windows, eventually culminating in significant global model degradation at coordinated trigger points. This attack strategy maintains stealth by concentrating the poisoning effect within specific time intervals while preserving normal behavior patterns outside attack periods.

For malicious client $c_i \in \mathcal{C}_p$ at communication round t, the poisoned local dataset $\tilde{\mathcal{D}}_i^t$ is constructed according to:

$$\tilde{\mathcal{D}}_i^t = \begin{cases} \mathcal{D}_i \cup \mathcal{P}_i^t & \text{if } t \in \mathcal{T}_\phi, \\ \mathcal{D}_i & \text{otherwise,} \end{cases} \tag{3}$$

where $\mathcal{T}_\phi = [t_s, t_e] \subset \{1, 2, \ldots, T\}$ denotes the fixed attack interval, and the poisoned sample set $\mathcal{P}_i^t$ follows a sophisticated generation process incorporating temporal dynamics and spatial correlations.

The poisoned sample construction employs multi-dimensional trigger injection with harmonic modulation:

$$\mathcal{P}_i^t = \left\{ (x_j + \epsilon_i^t \odot \mathbf{a}_i^t + \sum_{k=1}^{K} \alpha_k \sin\left(\frac{2\pi k(t-t_s)}{|\mathcal{T}_\phi|} + \phi_k\right)\mathbf{w}_k, y_\tau) : (x_j, y_j) \in \mathcal{S}_i^t \right\}, \tag{4}$$

where $\odot$ denotes element-wise multiplication, $\mathbf{a}_i^t \in \mathbb{R}^m$ represents the primary attack direction vector, K is the number of harmonic components, α_k, ϕ_k control amplitude and phase parameters, $\mathbf{w}_k \in \mathbb{R}^m$ are harmonic weight vectors, y_τ denotes the target label, and $\mathcal{S}_i^t \subset \mathcal{D}_i$ represents the selected subset for poisoning.

The time-dependent perturbation magnitude ϵ_i^t follows an accumulative schedule incorporating memory effects and stochastic variations:

$$\epsilon_i^t = \alpha_i \sum_{\tau=t_s}^{t} \beta^{t-\tau}\gamma_\tau \cdot \mathbb{I}[\tau \in \mathcal{T}_\phi] \cdot \exp\left(-\frac{(\tau-\mu_i)^2}{2\sigma_i^2}\right) \cdot \left(1 + \delta_i \sum_{j\in\mathcal{N}_i} \zeta_{i,j}^\tau\right), \tag{5}$$

where $\alpha_i > 0$ represents the base accumulation rate, $\beta \in (0,1)$ is the temporal decay factor, γ_τ are stochastic scaling factors following a Markov chain with transition probabilities $P_{k,l} = \frac{\exp(\theta_{k,l})}{\sum_j \exp(\theta_{k,j})}$, the Gaussian envelope ensures smooth temporal transitions with mean μ_i and variance σ_i^2, δ_i controls neighborhood influence, $\mathcal{N}_i$ represents neighboring clients, and $\zeta_{i,j}^\tau$ captures inter-client correlation effects.

3.2.2. Multi-Period Data Poisoning Attack

The multi-period data poisoning attack exploits temporal dependencies by distributing backdoor injection across multiple non-consecutive communication rounds, creating sophisticated interference

 doi:10.20944/preprints202510.0141.v1

patterns that evade single-period detection mechanisms while maintaining cumulative backdoor effectiveness.

The poisoned dataset construction employs period-specific trigger variations across disjoint temporal intervals $\mathcal{T}_\mu = \{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_K\}$ where $\mathcal{T}_j \cap \mathcal{T}_k = \emptyset$ for $j \neq k$:

$$\tilde{\mathcal{D}}_i^t = \begin{cases} \mathcal{D}_i \cup \mathcal{P}_{i,j}^t & \text{if } t \in \mathcal{T}_j, j \in \{1, 2, \ldots, K\}, \\ \mathcal{D}_i & \text{otherwise,} \end{cases} \tag{6}$$

where period-specific poisoned samples $\mathcal{P}_{i,j}^t$ incorporate adaptive trigger patterns and cross-period consistency constraints.

The multi-dimensional periodic signal injection follows:

$$\mathcal{P}_{i,j}^t = \left\{ (x_l + \gamma_i \mathbf{S}_{i,j}^t + \delta_i \mathbf{Q}_{i,j}^t, y_\tau) : (x_l, y_l) \in \mathcal{S}_{i,j}^t \right\}, \tag{7}$$

where $\gamma_i, \delta_i \in \mathbb{R}_{>0}$ control injection intensities for primary and secondary periodic signals $\mathbf{S}_{i,j}^t$ and $\mathbf{Q}_{i,j}^t$ respectively.

The primary multi-dimensional periodic signal $\mathbf{S}_{i,j}^t$ is defined as:

$$\begin{aligned} \mathbf{S}_{i,j}^t = &\sum_{k=1}^{K_j} \mathbf{A}_{i,j,k} \odot \sin\left(\frac{2\pi t}{P_{i,j,k}} + \phi_{i,j,k} \right) \odot \mathbf{W}_{i,j,k}^t \\ &+ \sum_{l=1}^{L_j} \mathbf{B}_{i,j,l} \odot \cos\left(\frac{2\pi t}{Q_{i,j,l}} + \psi_{i,j,l} \right) \odot \mathbf{V}_{i,j,l}^t, \end{aligned} \tag{8}$$

where K_j and L_j represent the numbers of sine and cosine components for period j, $\mathbf{A}_{i,j,k}, \mathbf{B}_{i,j,l} \in \mathbb{R}^m$ denote amplitude vectors, $P_{i,j,k}, Q_{i,j,l}$ are fundamental periods, $\phi_{i,j,k}, \psi_{i,j,l}$ are phase shifts, and $\mathbf{W}_{i,j,k}^t, \mathbf{V}_{i,j,l}^t$ are time-varying weight vectors.

3.2.3. Model Weight Poisoning Attack

The model weight poisoning attack creates direct manipulation of local model parameters before transmission to the server, generating high-intensity anomalous parameter patterns that bypass data-level detection mechanisms while maintaining coordinated execution across multiple malicious clients.

The poisoned model update follows a multi-state regime-switching framework with sophisticated perturbation strategies:

$$\tilde{\theta}_i^t = \begin{cases} \theta_i^t + \delta_i^t \odot \mathbf{n}_i^t + \eta_i^t \mathbf{m}_i^t & \text{if } c_i \in \mathcal{C}_p \text{ and } M_i^t = 1, \\ \theta_i^t + \frac{1}{2}\delta_i^t \odot \mathbf{n}_i^t & \text{if } c_i \in \mathcal{C}_p \text{ and } M_i^t = 2, \\ \theta_i^t + \varepsilon_i^t \boldsymbol{\chi}_i^t & \text{if } c_i \in \mathcal{C}_p \text{ and } M_i^t = 3, \\ \theta_i^t & \text{otherwise,} \end{cases} \tag{9}$$

where $M_i^t \in \{1, 2, 3, 4\}$ is a Markov chain state indicator controlling attack intensity, δ_i^t controls primary poisoning magnitude, $\mathbf{n}_i^t$ is the primary directional perturbation vector, η_i^t controls secondary poisoning magnitude, $\mathbf{m}_i^t$ is the secondary perturbation vector, ε_i^t controls background noise magnitude, and $\boldsymbol{\chi}_i^t$ represents background perturbations.

 doi:10.20944/preprints202510.0141.v1

The poisoning intensity follows a multi-phase decay model coordinated across malicious clients:

$$\delta_i^t = \delta_{m,i} f_i(t - t_{\star,i}) \cdot \mathbb{I}[c_i \in \mathcal{C}_p], \tag{10}$$

$$f_i(u) = \begin{cases} \exp\left(-\frac{u}{\tau_{d,1,i}}\right) & \text{if } 0 \le u \le T_{1,i}, \\ \exp\left(-\frac{u-T_{1,i}}{\tau_{d,2,i}}\right) & \text{if } T_{1,i} < u \le T_{2,i}, \\ \exp\left(-\frac{u-T_{2,i}}{\tau_{d,3,i}}\right) & \text{if } u > T_{2,i}, \end{cases} \tag{11}$$

where $\delta_{m,i}$ is the maximum intensity for client c_i, $t_{\star,i}$ is the attack initiation time, $\tau_{d,j,i}$ are phase-specific decay time constants, and $T_{j,i}$ are phase transition times.

3.3. Multi-Layer Defense Framework

Our defense strategy employs a sophisticated three-tier detection architecture that combines temporal behavioral analysis, robust statistical aggregation, and multi-scale validation protocols to counter the identified attack vectors while maintaining theoretical guarantees and computational efficiency.

3.3.1. Temporal Behavioral Analysis Layer

The temporal analysis layer monitors client behavioral patterns across communication rounds through comprehensive statistical profiling and anomaly detection mechanisms incorporating both individual client dynamics and cross-client correlation analysis.

For each client c_i, we construct a multi-dimensional behavioral profile vector encompassing temporal features, historical patterns, connectivity measures, and uncertainty quantification:

$$\mathbf{b}_i^t = [\mathbf{F}_i^t, \mathbf{H}_i^t, \mathbf{C}_i^t, \mathbf{U}_i^t, \mathbf{R}_i^t], \tag{12}$$

where $\mathbf{F}_i^t$ represents temporal features, $\mathbf{H}_i^t$ encodes historical patterns, $\mathbf{C}_i^t$ models connectivity, $\mathbf{U}_i^t$ quantifies uncertainty, and $\mathbf{R}_i^t$ captures resource utilization patterns.

The temporal feature vector $\mathbf{F}_i^t \in \mathbb{R}^{d_f}$ contains multi-scale statistical features extracted from recent communication windows:

$$\mathbf{F}_i^t = [\mu_i^t, \sigma_i^t, \varsigma_i^t, \kappa_i^t, \mathbf{F}_{\omega,i}^t, \mathbf{F}_{\psi,i}^t, \mathbf{F}_{\xi,i}^t], \tag{13}$$

where the statistical moments are computed using robust estimators across multiple time scales:

$$\mu_i^t = \frac{1}{w_f} \sum_{\tau=t-w_f+1}^{t} \|\theta_i^\tau\|_F + \frac{1}{w_s} \sum_{\tau=t-w_s+1}^{t-w_f} \|\theta_i^\tau\|_F, \tag{14}$$

$$\sigma_i^t = \sqrt{\frac{1}{w_f - 1} \sum_{\tau=t-w_f+1}^{t} (\|\theta_i^\tau\|_F - \mu_i^t)^2 + \epsilon_\sigma}, \tag{15}$$

$$\varsigma_i^t = \frac{1}{w_f} \sum_{\tau=t-w_f+1}^{t} \left(\frac{\|\theta_i^\tau\|_F - \mu_i^t}{\sigma_i^t} \right)^3 + \rho_\varsigma \varsigma_i^{t-1}, \tag{16}$$

$$\kappa_i^t = \frac{1}{w_f} \sum_{\tau=t-w_f+1}^{t} \left(\frac{\|\theta_i^\tau\|_F - \mu_i^t}{\sigma_i^t} \right)^4 - 3 + \rho_\kappa \kappa_i^{t-1}, \tag{17}$$

where w_f and w_s are short and long window sizes, ϵ_σ prevents numerical instability, $\rho_\varsigma, \rho_\kappa \in (0,1)$ provide temporal smoothing, $\mathbf{F}_{\omega,i}^t \in \mathbb{R}^{d_\omega}$ contains dominant frequency components from spectral analysis, $\mathbf{F}_{\psi,i}^t \in \mathbb{R}^{d_\psi}$ includes wavelet coefficients, and $\mathbf{F}_{\xi,i}^t \in \mathbb{R}^{d_\xi}$ represents spectral entropy measures.

 doi:10.20944/preprints202510.0141.v1

The anomaly detection mechanism employs multi-scale sliding window analysis with adaptive thresholding:

$$\mathcal{A}_i^t = \bigvee_{w \in \mathcal{W}} \left[\frac{1}{w} \sum_{\tau=t-w+1}^{t} \|\mathbf{b}_i^\tau - \boldsymbol{\mu}_i^{t-w}\|_{\boldsymbol{\Sigma}_i^{t-w}} > \tau_\alpha^w \right], \tag{18}$$

where $\mathcal{W} = \{w_1, w_2, \ldots, w_L\}$ contains multiple window sizes, $\boldsymbol{\mu}_i^{t-w}$ and $\boldsymbol{\Sigma}_i^{t-w}$ represent historical mean and covariance computed through robust estimation, $\|\cdot\|_{\boldsymbol{\Sigma}}$ denotes the Mahalanobis distance, and τ_α^w are scale-specific detection thresholds.

For multi-period attack detection, we implement a sophisticated pattern matching algorithm based on dynamic time warping:

$$\mathcal{P}_i^t = \max_{P \in \mathcal{P}_\Theta} \text{DTW}(\{\mathcal{A}_i^\tau\}_{\tau=t-T_P+1}^{t}, P), \tag{19}$$

where $\mathcal{P}_\Theta$ contains known attack pattern templates and DTW computes optimal alignment scores accounting for temporal distortions.

The DTW distance is computed through dynamic programming with warping constraints:

$$\text{DTW}(\mathbf{X}, \mathbf{Y}) = D[n_x, n_y], \tag{20}$$

$$D[i,j] = d(\mathbf{x}_i, \mathbf{y}_j) + \min\{D[i-1,j], D[i,j-1], D[i-1,j-1]\}, \tag{21}$$

$$\text{s.t.} \quad |i-j| \leq W_{\text{warp}}, \tag{22}$$

where $d(\mathbf{x}_i, \mathbf{y}_j)$ is the local distance function, n_x, n_y are sequence lengths, and W_{warp} constrains warping flexibility.

3.3.2. Robust Statistical Aggregation Layer

The aggregation layer employs Byzantine-robust techniques enhanced with temporal consistency constraints and multi-dimensional filtering to identify and mitigate malicious model updates while preserving the convergence properties of the global optimization process. The robust aggregation mechanism operates through sequential filtering stages incorporating geometric median estimation, distance-based outlier detection, and weighted consensus formation. The robust central tendency is established through iterative geometric median estimation:

$$\hat{\boldsymbol{\theta}}_\nu^t = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{N} \|\boldsymbol{\theta} - \theta_i^t\|_F, \tag{23}$$

solved using the accelerated Weiszfeld algorithm with momentum-based convergence enhancement:

$$\boldsymbol{\theta}^{(k+1)} = \frac{\sum_{i=1}^{N} \frac{\theta_i^t}{\|\boldsymbol{\theta}^{(k)} - \theta_i^t\|_F + \epsilon}}{\sum_{i=1}^{N} \frac{1}{\|\boldsymbol{\theta}^{(k)} - \theta_i^t\|_F + \epsilon}}, \tag{24}$$

$$\boldsymbol{\theta}^{(k+1)} \leftarrow \boldsymbol{\theta}^{(k+1)} + \alpha_k(\boldsymbol{\theta}^{(k+1)} - \boldsymbol{\theta}^{(k)}), \tag{25}$$

where ϵ prevents numerical instabilities and α_k provides adaptive momentum acceleration. Suspicious clients are identified through sophisticated distance-based analysis incorporating multiple statistical measures:

$$\mathcal{S}^t = \left\{ c_i : \|\theta_i^t - \hat{\boldsymbol{\theta}}_\nu^t\|_F > Q_{1-\alpha}(\{d_j^t\}_{j=1}^N) + \beta \cdot \text{IQR}(\{d_j^t\}_{j=1}^N) \vee \mathcal{A}_i^t = \text{True} \right\}, \tag{26}$$

 doi:10.20944/preprints202510.0141.v1

where $d_j^t = \|\theta_j^t - \hat{\boldsymbol{\theta}}_v^t\|_F$, $Q_{1-\alpha}$ denotes the $(1-\alpha)$-quantile, IQR represents the interquartile range, $\beta > 0$ controls filtering sensitivity, and $\mathcal{A}_i^t$ incorporates temporal anomaly detection results. The final global model incorporates temporal consistency constraints and reputation-based weighting:

$$\theta^t = \arg\min_{\boldsymbol{\theta}} \sum_{i \notin \mathcal{S}^t} w_i^t \|\boldsymbol{\theta} - \theta_i^t\|_F^2 + \lambda_\tau \|\boldsymbol{\theta} - \theta^{t-1}\|_F^2 + \lambda_v \|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_v^t\|_F^2, \tag{27}$$

where the client weights incorporate multi-factor scoring:

$$w_i^t = \frac{\exp(-\gamma \cdot (\mathcal{A}_i^t + \mathbb{I}[c_i \in \mathcal{S}^t] + \mathcal{R}_i^t))}{\sum_{j \notin \mathcal{S}^t} \exp(-\gamma \cdot (\mathcal{A}_j^t + \mathbb{I}[c_j \in \mathcal{S}^t] + \mathcal{R}_j^t))}, \tag{28}$$

and the reputation score $\mathcal{R}_i^t$ captures historical behavior patterns through multi-scale temporal weighting:

$$\mathcal{R}_i^t = \sum_{k=1}^{K_r} \omega_k \sum_{\tau=\max(1,t-w_k)}^{t-1} \exp\left(-\frac{t-\tau}{\sigma_k}\right) \mathbb{I}[c_i \text{ flagged at round } \tau], \tag{29}$$

where K_r is the number of temporal scales, ω_k are scale-specific weights, w_k are window sizes, and σ_k control exponential decay rates.

3.3.3. Multi-Scale Validation Layer

The validation layer provides continuous monitoring of global model integrity through clean performance tracking, backdoor trigger detection, and coordinated response protocols incorporating sophisticated statistical testing and model rollback mechanisms. We maintain a held-out validation set $\mathcal{D}_v$ and track performance degradation through robust statistical testing:

$$\mathcal{L}_c^t = \frac{1}{|\mathcal{D}_v|} \sum_{(x,y) \in \mathcal{D}_v} \ell(f_{\theta^t}(x), y), \tag{30}$$

with alert generation through sequential hypothesis testing:

$$\mathcal{A}_c^t = \mathbb{I}[\mathcal{L}_c^t > \mathcal{L}_c^{t-1} + \tau_c] \vee \mathbb{I}[\mathcal{L}_c^t > \mu_\beta + \sigma_\beta \cdot z_{\alpha_c}], \tag{31}$$

where μ_β, σ_β characterize historical performance distribution and z_{α_c} is the critical value for significance level α_c. We maintain a trigger detection dataset $\mathcal{D}_\xi = \{(x_j + \delta_k, y_j)\}$ covering potential trigger patterns and monitor:

$$\mathcal{L}_\beta^t = \max_{y_\tau} \frac{1}{|\mathcal{D}_\xi|} \sum_{(x_\xi, y_\pi) \in \mathcal{D}_\xi} \mathbb{I}[f_{\theta^t}(x_\xi) = y_\tau], \tag{32}$$

with backdoor detection alert:

$$\mathcal{A}_\beta^t = \mathbb{I}[\mathcal{L}_\beta^t > \tau_\beta] \vee \mathbb{I}[\frac{\mathcal{L}_\beta^t}{\mathcal{L}_c^t} > \tau_\rho], \tag{33}$$

where τ_β and τ_ρ are detection thresholds for absolute and relative trigger activation rates. Upon alert generation, the system implements graduated response protocols through state transition mechanisms:

$$\mathcal{R}^t = \begin{cases} \mathcal{M}_e & \text{if } \mathcal{A}_c^t \wedge \neg \mathcal{A}_\beta^t, \\ \mathcal{I}_c & \text{if } \mathcal{A}_\beta^t \wedge \mathcal{P}_i^t > \tau_\pi, \\ \mathcal{R}_\theta & \text{if } \mathcal{A}_\beta^t \wedge \mathcal{L}_\beta^t > \tau_\kappa, \\ \mathcal{N}_o & \text{otherwise,} \end{cases} \tag{34}$$

 doi:10.20944/preprints202510.0141.v1

where $\mathcal{M}_e$ denotes enhanced monitoring, $\mathcal{I}_c$ represents client investigation, $\mathcal{R}_\theta$ implements model rollback, and $\mathcal{N}_o$ continues normal operation. The model rollback mechanism employs temporal checkpointing with integrity verification:

$$\theta^t = \begin{cases} \theta^{t-k_\star} & \text{if } \mathcal{R}^t = \mathcal{R}_\theta \wedge \mathcal{V}(\theta^{t-k_\star}) > \tau_v, \\ \arg\min_{k \le K_{\max}} \mathcal{V}(\theta^{t-k}) & \text{if } \mathcal{R}^t = \mathcal{R}_\theta \wedge \mathcal{V}(\theta^{t-k_\star}) \le \tau_v, \\ \theta^t & \text{otherwise,} \end{cases} \tag{35}$$

where $k_\star$ is determined by the severity of the detected anomaly, $\mathcal{V}(\cdot)$ quantifies model integrity through comprehensive validation metrics, and $K_{\max}$ bounds the maximum rollback distance.

3.4. MDP Framework for Defense Coordination

We formulate the temporal backdoor defense problem as a Markov Decision Process (MDP) defined as $\mathcal{M}_d = (\mathcal{S}_d, \mathcal{A}_d, \mathcal{P}_d, \mathcal{R}_d, \gamma)$, where each component captures the sequential decision-making nature of coordinated defense mechanisms with comprehensive state representation and sophisticated action space design for optimal defense coordination.

3.4.1. Defense State Space Design

The defense environment state $s_t^d \in \mathcal{S}_d$ at communication round t encompasses comprehensive information about the federated system security status through multiple information channels:

$$s_t^d = \{\mathbf{B}_t, \mathbf{D}_t, \mathbf{A}_t, \mathbf{S}_t, \mathbf{U}_t, \mathbf{R}_t, \mathbf{G}_t\}, \tag{36}$$

where $\mathbf{B}_t$ represents client behavioral profiles, $\mathbf{D}_t$ encodes detection history, $\mathbf{A}_t$ models anomaly indicators, $\mathbf{S}_t$ maintains security metrics, $\mathbf{U}_t$ quantifies uncertainty measures, $\mathbf{R}_t$ captures resource utilization, and $\mathbf{G}_t$ represents global system statistics.

The client behavioral profile matrix $\mathbf{B}_t \in \mathbb{R}^{N \times d_b}$ contains comprehensive behavioral features for all clients:

$$\mathbf{B}_t[i,:] = [\mathbf{b}_i^t, \Delta\mathbf{b}_i^t, \|\mathbf{b}_i^t - \mathbf{b}_i^{t-1}\|_2, \text{rank}(\mathbf{b}_i^t), \mathbf{B}_{\rho,i}^t], \tag{37}$$

where $\mathbf{b}_i^t$ is the current behavioral profile, $\Delta\mathbf{b}_i^t$ represents temporal changes, the norm captures profile stability, $\text{rank}(\mathbf{b}_i^t)$ indicates behavioral complexity, and $\mathbf{B}_{\rho,i}^t$ encodes cross-client correlations.

The detection history matrix $\mathbf{D}_t \in \mathbb{R}^{N \times d_d}$ maintains weighted information about previous detection decisions with multi-scale temporal decay:

$$\mathbf{D}_t[i,:] = \left[\sum_{k=1}^{K_d} \omega_{d,k} \sum_{\tau=\max(1,t-w_{d,k})}^{t-1} \omega_d^{(t-\tau)/k} \cdot \mathbb{I}[\hat{s}_i^\tau = s_p], \mathcal{R}_i^t, \mathcal{C}_i^t, \mathcal{V}_i^t \right], \tag{38}$$

where K_d is the number of temporal scales, $\omega_{d,k}$ are scale-specific weights, $w_{d,k}$ are scale-specific window sizes, $\omega_d \in (0,1)$ provides exponential temporal weighting, $\mathcal{R}_i^t$ represents reputation scores, $\mathcal{C}_i^t$ captures confidence levels, and $\mathcal{V}_i^t$ quantifies detection variance.

The anomaly indicator vector $\mathbf{A}_t \in \mathbb{R}^N$ captures multiple types of anomalies through ensemble-based detection:

$$\mathbf{A}_t[i] = \alpha_a \mathcal{A}_{\tau,i}^t + \beta_a \mathcal{A}_{\sigma,i}^t + \gamma_a \mathcal{A}_{\gamma,i}^t + \delta_a \mathcal{A}_{\varsigma,i}^t, \tag{39}$$

where $\alpha_a, \beta_a, \gamma_a, \delta_a$ are weighting coefficients, $\mathcal{A}_{\tau,i}^t$ represents temporal anomalies, $\mathcal{A}_{\sigma,i}^t$ captures statistical anomalies, $\mathcal{A}_{\gamma,i}^t$ indicates geometric median deviations, and $\mathcal{A}_{\varsigma,i}^t$ quantifies spectral anomalies.

The security metrics vector $\mathbf{S}_t \in \mathbb{R}^{d_s}$ tracks system-wide security indicators:

$$\mathbf{S}_t = [\xi_t^g, \xi_t^h, \xi_t^c, \xi_t^z, \xi_t^\theta, \xi_t^r], \tag{40}$$

 doi:10.20944/preprints202510.0141.v1

where ξ_t^g measures global model integrity, ξ_t^h quantifies information entropy, ξ_t^c captures aggregation consensus, ξ_t^z indicates system stability, ξ_t^θ represents threat level assessment, and ξ_t^r measures defense resilience.

3.4.2. Defense Action Space Formulation

The defense action space $\mathcal{A}_d$ encompasses coordinated defense decisions, resource allocation, and temporal control through a hierarchical structure:

$$\mathcal{A}_d = \mathcal{A}_\delta \times \mathcal{A}_\mu \times \mathcal{A}_\alpha \times \mathcal{A}_\zeta, \tag{41}$$

where $\mathcal{A}_\delta$ represents detection actions, $\mathcal{A}_\mu$ determines mitigation strategies, $\mathcal{A}_\alpha$ controls resource allocation, and $\mathcal{A}_\zeta$ manages temporal adaptations.

The detection action space includes comprehensive detection strategies:

$$\mathcal{A}_\delta = \{a_m, a_i, a_v, a_p, a_q, a_e\}^N, \tag{42}$$

where a_m denotes passive monitoring, a_i represents detailed inspection, a_v indicates verification protocols, a_p triggers active probing, a_q implements temporary isolation, and a_e enforces permanent exclusion.

The mitigation action space operates on multiple defense strategies with coordination constraints:

$$\mathcal{A}_\mu = \{\boldsymbol{\mu} \in [0,1]^{N\times M} : \sum_{j=1}^{M} \mu_{i,j} = 1 \forall i, \sum_{i=1}^{N} \mu_{i,j} \leq C_j \forall j\}, \tag{43}$$

$$\mathcal{C}_\rho = \max_{i,j,k,l} \frac{\mu_{i,j}}{\mu_{k,l}} \leq \zeta_\rho, \tag{44}$$

where $\mu_{i,j}$ represents the intensity of mitigation strategy j applied to client i, M is the number of mitigation strategies including: enhanced monitoring ($j = 1$), gradient clipping ($j = 2$), noise injection ($j = 3$), weight decay regularization ($j = 4$), and adaptive learning rate scaling ($j = 5$), C_j bounds the total capacity for strategy j, $\mathcal{C}_\rho$ ensures coordination constraints, and ζ_ρ limits mitigation disparity.

The resource allocation space manages computational and communication resources across defense mechanisms:

$$\mathcal{A}_\alpha = \{\boldsymbol{\rho} \in [0,1]^{D\times R} : \sum_{r=1}^{R} \rho_{d,r} = 1 \forall d, \rho_{d,r} \geq \rho_{\min} \forall d, r\}, \tag{45}$$

$$\mathcal{E}_\eta = \sum_{d=1}^{D} \sum_{r=1}^{R} \rho_{d,r} \mathcal{F}_{d,r}(\mathbf{S}_t) \geq E_{\min}, \tag{46}$$

where $\rho_{d,r}$ represents the fraction of resource type r allocated to defense mechanism d, D is the number of defense mechanisms including: temporal analysis ($d = 1$), statistical aggregation ($d = 2$), and validation monitoring ($d = 3$), R is the number of resource types including: CPU computation ($r = 1$), memory storage ($r = 2$), and communication bandwidth ($r = 3$), $\rho_{\min}$ ensures minimum allocation, $\mathcal{F}_{d,r}$ quantifies efficiency functions, and $E_{\min}$ guarantees minimum defense effectiveness.

The temporal adaptation action space $\mathcal{A}_\zeta$ controls dynamic client participation and weight acceptance decisions:

$$\mathcal{A}_\zeta = \{\boldsymbol{\zeta} \in \{0,1\}^N : \sum_{i=1}^{N} \zeta_i \geq N_{\min}\}, \tag{47}$$

 doi:10.20944/preprints202510.0141.v1

where $\zeta_i \in \{0,1\}$ indicates whether to accept model weights from client c_i at the current round (1 for accept, 0 for reject), and $N_{\min}$ ensures minimum participation for convergence. The temporal adaptation decision for each client is governed by:

$$\zeta_i^t = \begin{cases} 0 & \text{if } \mathcal{A}_i^t = \text{True} \wedge \mathcal{P}_i^t > \tau_\zeta, \\ 0 & \text{if } c_i \in \mathcal{S}^t \wedge \mathcal{R}_i^t > \tau_\rho, \\ \text{Bernoulli}(p_i^t) & \text{if } \mathcal{A}_i^t = \text{True} \wedge \mathcal{P}_i^t \leq \tau_\zeta, \\ 1 & \text{otherwise,} \end{cases} \tag{48}$$

where τ_ζ and τ_ρ are rejection thresholds for anomaly scores and reputation scores respectively, and the stochastic acceptance probability is:

$$p_i^t = \sigma(\alpha_\zeta - \beta_\zeta \mathcal{A}_i^t - \gamma_\zeta \mathcal{R}_i^t - \delta_\zeta \|\theta_i^t - \hat{\boldsymbol{\theta}}_v^t\|_F), \tag{49}$$

where $\sigma(\cdot)$ is the sigmoid function, and $\alpha_\zeta, \beta_\zeta, \gamma_\zeta, \delta_\zeta$ are learned parameters that balance acceptance probability based on anomaly levels, reputation, and parameter deviation.

3.4.3. Defense Transition Dynamics

The state transition probabilities $\mathcal{P}_d : \mathcal{S}_d \times \mathcal{A}_d \times \mathcal{S}_d \to [0,1]$ capture the stochastic evolution of the defense environment under coordinated security actions:

$$\mathcal{P}_d(s_{t+1}^d | s_t^d, a_t^d) = \prod_{j=1}^{|\mathbf{s}|} \mathcal{P}_j(s_{t+1,j}^d | s_t^d, a_t^d), \tag{50}$$

where the factorization assumes conditional independence across state components given the current state and action.

The behavioral profile transition dynamics incorporate temporal evolution and defense interventions:

$$\mathcal{P}_{\mathbf{B}}(\mathbf{B}_{t+1} | s_t^d, a_t^d) = \prod_{i=1}^{N} \mathcal{N}(\mathbf{b}_{i,t+1} | \boldsymbol{\mu}_{b,i}^t + \mathbf{W}_b a_{\delta,i}^t, \boldsymbol{\Sigma}_{b,i}^t), \tag{51}$$

$$\boldsymbol{\mu}_{b,i}^t = \mathbf{A}_b \mathbf{b}_i^t + \mathbf{B}_b \mathbf{h}_i^t + \mathbf{c}_b, \tag{52}$$

$$\mathbf{h}_i^t = \tanh(\mathbf{U}_h \mathbf{b}_i^{t-1} + \mathbf{V}_h a_{\delta,i}^{t-1} + \mathbf{b}_h), \tag{53}$$

where $\mathbf{A}_b, \mathbf{B}_b, \mathbf{W}_b$ are learned transition matrices, $\mathbf{h}_i^t$ represents latent behavioral states, $\mathbf{U}_h, \mathbf{V}_h$ control temporal dependencies, and $\boldsymbol{\Sigma}_{b,i}^t$ captures uncertainty in behavioral evolution.

The detection history transitions incorporate memory decay and decision outcomes:

$$\mathcal{P}_{\mathbf{D}}(\mathbf{D}_{t+1} | s_t^d, a_t^d) = \prod_{i=1}^{N} \delta(\mathbf{d}_{i,t+1} - \mathbf{T}_d(\mathbf{d}_{i,t}, a_{\delta,i}^t, \omega_t)), \tag{54}$$

where $\delta(\cdot)$ is the Dirac delta function, $\mathbf{T}_d$ represents the deterministic update function, and ω_t captures environmental stochasticity.

 doi:10.20944/preprints202510.0141.v1

3.4.4. Defense Reward Function Design

The defense reward function $\mathcal{R}_d : \mathcal{S}_d \times \mathcal{A}_d \times \mathcal{S}_d \to \mathbb{R}$ incorporates multiple defense objectives through a sophisticated multi-criteria framework:

$$\begin{aligned}\mathcal{R}_d(s_t^d, a_t^d, s_{t+1}^d) = \lambda_1 \sum_{i=1}^{N} R_{\delta,i}(s_t^d, a_t^d, s_{t+1}^d) + \lambda_2 \sum_{i=1}^{N} R_{\mu,i}(s_t^d, a_t^d) \\ + \lambda_3 R_\sigma(s_t^d, a_t^d, s_{t+1}^d) + \lambda_4 R_\eta(a_t^d) \\ - \lambda_5 R_\kappa(a_t^d) - \lambda_6 R_\phi(s_t^d, a_t^d, s_{t+1}^d),\end{aligned} \tag{55}$$

where the reward components capture detection accuracy, mitigation effectiveness, security improvement, operational efficiency, resource costs, and system disruption.

The detection reward incorporates accuracy, timeliness, and confidence weighting:

$$\begin{aligned}R_{\delta,i}(s_t^d, a_t^d, s_{t+1}^d) = \mathbb{I}[s_i^* = s_p] \cdot \mathbb{I}[a_{\delta,i} \in \{a_i, a_v, a_p\}] \cdot \eta_\delta \\ \cdot e^{-\lambda_\tau (t - t_{\pi,i})} \cdot (1 - \mathbf{U}_t[i]) \cdot \mathcal{W}_v^i,\end{aligned} \tag{56}$$

where s_i^* represents the true client state, η_δ is the base detection reward, λ_τ controls temporal decay, $t_{\pi,i}$ is the actual attack start time, $\mathbf{U}_t[i]$ quantifies detection uncertainty, and $\mathcal{W}_v^i$ captures network effect weighting.

The mitigation reward measures the effectiveness of applied defense strategies:

$$R_{\mu,i}(s_t^d, a_t^d) = \sum_{j=1}^{M} \mu_{i,j} \cdot \mathcal{E}_j(\mathbf{S}_t, \mathbf{A}_t[i]) \cdot \mathbb{I}[a_{\delta,i} \neq a_m] \cdot \eta_\mu, \tag{57}$$

where $\mathcal{E}_j$ quantifies the effectiveness of mitigation strategy j given system state and anomaly levels, and η_μ is the base mitigation reward.

The security reward tracks overall system security improvement:

$$R_\sigma(s_t^d, a_t^d, s_{t+1}^d) = \sum_{k=1}^{d_s} \omega_k^\sigma \max(0, \mathbf{S}_{t+1}[k] - \mathbf{S}_t[k]) + \eta_\sigma \mathcal{I}_\theta^t, \tag{58}$$

where ω_k^σ are security metric weights, $\mathbf{S}_t[k]$ represents individual security metrics, η_σ is the threat mitigation reward, and $\mathcal{I}_\theta^t$ indicates successful threat neutralization.

3.4.5. Defense Policy Optimization

The optimal defense policy $\pi^* : \mathcal{S}_d \to \mathcal{A}_d$ is learned through advanced reinforcement learning techniques incorporating temporal credit assignment and multi-objective optimization:

$$\pi^* = \arg\max_\pi \mathbb{E}_{\tau \sim \pi} \left[\sum_{t=0}^{T-1} \gamma^t \mathcal{R}_d(s_t^d, a_t^d, s_{t+1}^d) \right], \tag{59}$$

where τ represents a trajectory, $\gamma \in (0, 1)$ is the discount factor, and the expectation is taken over the policy-induced distribution.

The policy optimization employs actor-critic architecture with attention mechanisms:

$$\pi_\theta(a_t^d | s_t^d) = \text{softmax}(\mathbf{W}_\pi \mathbf{h}_\pi^t + \mathbf{b}_\pi), \tag{60}$$

$$\mathbf{h}_\pi^t = \text{Attention}(\mathbf{Q}_\pi, \mathbf{K}_\pi, \mathbf{V}_\pi) + \mathbf{f}_\pi(s_t^d), \tag{61}$$

$$V_\phi(s_t^d) = \mathbf{W}_V \mathbf{h}_V^t + b_V, \tag{62}$$

where $\mathbf{h}_\pi^t, \mathbf{h}_V^t$ are attention-enhanced hidden representations, $\mathbf{f}_\pi$ encodes state features, and θ, ϕ are learnable parameters.

 doi:10.20944/preprints202510.0141.v1

Algorithm 1 DEFEND: **DE**ep **F**ederated **E**nsemble **N**etwork **D**efense

1: **Input**: Client updates $\{\theta_i^t\}_{i=1}^N$, historical profiles $\{\mathbf{b}_i^{t-1}\}_{i=1}^N$, defense state s_{t-1}^d, hyperparameters $\lambda_\tau, \lambda_v, \gamma$, window sizes $\mathcal{W}$, detection thresholds $\{\tau_\alpha^w\}$;
2: **Output**: Aggregated model θ^t, updated profiles $\{\mathbf{b}_i^t\}_{i=1}^N$, defense action a_t^d;
3: **for** each client c_i **do**
4: Compute temporal profile $\mathbf{b}_i^t$ ▷ Equation (12)
5: Calculate statistical moments $\{\mu_i^t, \sigma_i^t, \varsigma_i^t, \kappa_i^t\}$ ▷ Equation (14)-(17)
6: Detect temporal anomalies $\mathcal{A}_i^t$ ▷ Equation (18)
7: Check multi-period patterns $\mathcal{P}_i^t$ ▷ Equation (19)
8: **end for**
9: Construct defense state s_t^d ▷ Equation (36)
10: Select defense action $a_t^d \sim \pi_\theta(a_t^d | s_t^d)$ ▷ Equation (60)
11: Determine client participation $\{\zeta_i^t\}_{i=1}^N$ ▷ Equation (48)
12: Compute geometric median $\hat{\boldsymbol{\theta}}_v^t$ for active clients ▷ Equation (23)
13: Apply Weiszfeld algorithm with updates ▷ Equation (24)-(25)
14: Perform outlier detection to identify $\mathcal{S}^t$ ▷ Equation (26)
15: Calculate client weights $\{w_i^t\}$ for participating clients ▷ Equation (28)
16: Execute weighted aggregation θ^t ▷ Equation (27)
17: Monitor clean performance $\mathcal{L}_c^t$ ▷ Equation (30)
18: Check trigger responses $\mathcal{L}_\beta^t$ ▷ Equation (32)
19: Generate alerts $\{\mathcal{A}_c^t, \mathcal{A}_\beta^t\}$ ▷ Equation (31)-(33)
20: **if** $\mathcal{A}_c^t \vee \mathcal{A}_\beta^t$ **then**
21: Execute graduated response $\mathcal{R}^t$ ▷ Equation (34)
22: Update reputation scores ▷ Equation (29)
23: **if** $\mathcal{R}^t = \mathcal{R}_\theta$ **then**
24: Perform model rollback ▷ Equation (35)
25: **end if**
26: **end if**
27: Compute reward $r_t^d = \mathcal{R}_d(s_t^d, a_t^d, s_{t+1}^d)$ ▷ Equation (55)
28: Update policy parameters θ and value function ϕ using PPO
29: Store experience tuple $(s_t^d, a_t^d, r_t^d, s_{t+1}^d)$ in replay buffer
30: **return** θ^t, $\{\mathbf{b}_i^t\}_{i=1}^N$, a_t^d;

The complete defense framework integrates all components into a cohesive algorithm that executes at each communication round through coordinated multi-layer processing and decision making:

We analyze the computational complexity of Algorithm 1 by examining each component separately and providing theoretical bounds for the overall framework execution. The computation of behavioral profiles for all clients in lines 3-6 requires $O(N \cdot d_f \cdot w_{\max})$ operations, where d_f is the feature dimension and $w_{\max} = \max(w_f, w_s)$ is the maximum window size. The anomaly detection using Equation (18) across multiple window sizes has complexity $O(N \cdot |\mathcal{W}| \cdot d_b^2)$ due to Mahalanobis distance computations, where d_b is the behavioral profile dimension. The DTW-based pattern matching in Equation (19) requires $O(N \cdot |\mathcal{P}_\Theta| \cdot T_P^2)$ operations for N clients and $|\mathcal{P}_\Theta|$ pattern templates. State construction according to Equation (36) has complexity $O(N \cdot d_s)$ where d_s is the total state dimension. Policy evaluation using the attention mechanism from Equation (60)-(61) requires $O(d_s^2 + d_a \cdot d_s)$ operations, where d_a is the action space dimension. The geometric median computation via the Weiszfeld algorithm in lines 10-11 has complexity $O(K_{\text{iter}} \cdot N_{\text{active}} \cdot d)$ where K_{iter} is the number of iterations, $N_{\text{active}} = \sum_{i=1}^N \zeta_i^t$ is the number of participating clients, and d is the model parameter dimension. Outlier detection using Equation (26) requires $O(N_{\text{active}}^2 \cdot d)$ for pairwise distance computations. The weighted aggregation from Equation (27) has complexity $O(N_{\text{active}} \cdot d)$. Clean performance monitoring using Equation (30) requires $O(|\mathcal{D}_v| \cdot d)$ operations for model evaluation. Trigger detection from Equation (32) has complexity $O(|\mathcal{D}_{\tilde{\varsigma}}| \cdot |\mathcal{Y}| \cdot d)$ where $|\mathcal{Y}|$ is the number of possible target labels. Model rollback using Equation (35) when triggered requires $O(K_{\max} \cdot d)$ operations. Reward computation using Equation (55) has complexity $O(N \cdot M)$ where M is the number of mitigation strategies. Policy and value function updates require $O(d_\theta + d_\phi)$ operations where d_θ, d_ϕ are the parameter dimensions. The total computational complexity per communication round is:

$$\mathcal{O}(N \cdot \max(N \cdot d, |\mathcal{W}| \cdot d_b^2, |\mathcal{P}_\Theta| \cdot T_P^2) + K_{\text{iter}} \cdot N_{\text{active}} \cdot d + |\mathcal{D}_v| \cdot d + |\mathcal{D}_{\tilde{\varsigma}}| \cdot |\mathcal{Y}| \cdot d), \tag{63}$$

 doi:10.20944/preprints202510.0141.v1

which scales quadratically with the number of clients in the worst case due to pairwise distance computations, but remains practical for moderate-scale federated deployments with $N \leq 100$ clients and efficient implementation of geometric median algorithms. The temporal adaptation mechanism in line 9 reduces the effective computational load by filtering out suspicious clients early, leading to $N_{\text{active}} < N$ in most scenarios, thereby improving overall efficiency.

4. Experiment

This section presents comprehensive experimental evaluations to validate the effectiveness of our proposed DEFEND framework against temporal backdoor attacks in federated learning environments. We conduct extensive experiments across multiple datasets, network architectures, and attack scenarios to demonstrate the robustness and practical applicability of our multi-layer defense mechanism.

4.1. Experimental Setup

We evaluate our framework on three widely-used federated learning benchmarks: CIFAR-10, FEMNIST, and MNIST. These datasets represent diverse application domains with varying data characteristics and complexity levels. To simulate realistic federated environments, we consider both Independent and Identically Distributed (IID) and Non-IID data distributions across clients. For Non-IID scenarios, we employ Dirichlet distribution with concentration parameters $\alpha \in \{0.1, 0.5, 1.0\}$, where smaller values indicate higher data heterogeneity. The client population varies from 10 to 50 participants to assess scalability across different federation sizes. We evaluate defense performance under various threat models by varying the malicious client ratio $\kappa \in \{0.1, 0.2, 0.3, 0.4\}$, representing scenarios where 10% to 40% of participants are compromised.

We employ two representative deep learning architectures: MobileNet V2 for lightweight mobile applications and ResNet-18 for standard computer vision tasks. The detailed experimental parameters and hyperparameter configurations are summarized in Table 2.

 doi:10.20944/preprints202510.0141.v1

Table 2. Experimental Parameters and Configuration Settings.

Parameter	Symbol	Value
Dataset and Client Configuration		
Datasets	–	CIFAR-10, FEMNIST, MNIST
Number of clients	N	10, 20, 30, 40, 50
Data distribution	–	Non-IID
Dirichlet concentration	α	0.1, 0.5, 1.0
Malicious client ratio	κ	0.1, 0.2, 0.3, 0.4
Model architecture	–	MobileNet V2, ResNet-18
Federated Learning Parameters		
Learning rate	η	0.01
Local epochs	–	5
Communication rounds	T	100
Batch size	–	32
Poisoning ratio	ρ	0.1
Defense Framework Parameters		
Short window size	w_f	5
Long window size	w_s	10
Detection threshold	τ_α	0.8
Temporal regularization	λ_τ	0.1
Pattern templates	$\lvert\mathcal{P}_\Theta\rvert$	10
Geometric median tolerance	ϵ	10^{-6}
Reputation decay scales	K_r	3
MDP and Reinforcement Learning Parameters		
Discount factor	γ	0.95
Policy learning rate	–	0.001
Training episodes	–	500
Max steps per episode	–	1000
Exploration rate	ϵ	0.1
Experience buffer size	–	10,000
Target network update	–	1000 steps
Minimum participation	$N_{\min}$	$\lceil 0.6N \rceil$

We implement the three temporal backdoor attack strategies described in Section 3.2: fixed-period data poisoning, multi-period data poisoning, and model weight poisoning attacks. For fixed-period attacks, malicious clients inject backdoor triggers during rounds 20-40 with poisoning ratio $\rho = 0.1$. Multi-period attacks distribute poisoning across three disjoint intervals: rounds 10-15, 25-30, and 45-50, maintaining the same total poisoning budget. Model weight poisoning attacks apply Gaussian perturbations with magnitude $\delta_i^t \in [0.001, 0.01]$ following the decay schedule in Equation (11). Trigger patterns consist of 3×3 pixel patches with intensity variations, and target labels are randomly selected from classes not present in the victim's local data distribution.

4.2. Evaluation Metrics

We employ three comprehensive metrics to evaluate the performance of our DEFEND framework from multiple perspectives:

4.2.1. Clean Accuracy under Defense

The clean accuracy under defense measures the model's classification performance on benign test samples when the defense framework is actively protecting against malicious clients, using the held-out validation set $\mathcal{D}_v$:

$$A_c^T = \frac{1}{|\mathcal{D}_v|} \sum_{(x,y)\in\mathcal{D}_v} \mathbb{I}[f_{\theta^T}(x) = y], \tag{64}$$

 doi:10.20944/preprints202510.0141.v1

where θ^T represents the final global model after T communication rounds, and f_{θ^T} denotes the model's prediction function. Higher values indicate better preservation of normal task performance under defense conditions.

4.2.2. Defense Success Rate

The defense success rate quantifies the effectiveness of our defense framework by measuring the proportion of triggered samples that do not produce the attacker's target label, using the trigger detection dataset $\mathcal{D}_\xi$:

$$S_d^T = 1 - \max_{y_\tau} \frac{1}{|\mathcal{D}_\xi|} \sum_{(x_\xi, y_\pi) \in \mathcal{D}_\xi} \mathbb{I}[f_{\theta^T}(x_\xi) = y_\tau], \tag{65}$$

where x_ξ represents samples with injected triggers, y_π are the original labels, and y_τ represents the target labels across all possible attack targets. Higher S_d^T values indicate more effective defense against backdoor attacks.

4.2.3. Temporal Detection Efficiency

The temporal detection efficiency measures the framework's ability to rapidly and accurately identify malicious clients by considering both detection accuracy and temporal responsiveness:

$$E_d = \frac{1}{|\mathcal{C}_p|} \sum_{c_i \in \mathcal{C}_p} \frac{\mathbb{I}[\exists t \in \{t_{a,i}, \ldots, T\} : \hat{s}_i^t = s_p \wedge s_i^t = s_p]}{t_{d,i} - t_{a,i} + 1}, \tag{66}$$

where $t_{d,i}$ is the communication round when client c_i is first correctly classified as poisoned state s_p, $t_{a,i}$ is the round when client c_i begins malicious behavior, and $\mathcal{C}_p$ represents the set of malicious clients. The indicator function ensures only successful detections are counted. Higher E_d values indicate faster and more accurate malicious client identification.

4.3. Implementation Details

Our DEFEND framework is implemented using PyTorch 2.1.0 and Python 3.9. All experiments are conducted on NVIDIA A100 GPUs with 40GB memory. The MDP-based defense coordination employs Proximal Policy Optimization (PPO) for policy learning with clip ratio 0.2 and entropy coefficient 0.01. The geometric median computation uses the accelerated Weiszfeld algorithm with momentum coefficient $\alpha_k = 0.9$ and convergence tolerance $\epsilon = 10^{-6}$.

For behavioral profile construction, we extract spectral features using Fast Fourier Transform (FFT) with window size 32, and wavelet coefficients using Daubechies-4 wavelets with 3 decomposition levels. The pattern matching employs Dynamic Time Warping with warping constraint $W_{\text{warp}} = 5$ and Euclidean local distance function.

The validation set $\mathcal{D}_v$ comprises 10% of the total training data, randomly sampled and held out from all clients. The trigger detection dataset $\mathcal{D}_\xi$ contains 1000 samples per class with systematically generated trigger patterns covering various spatial positions and intensities.

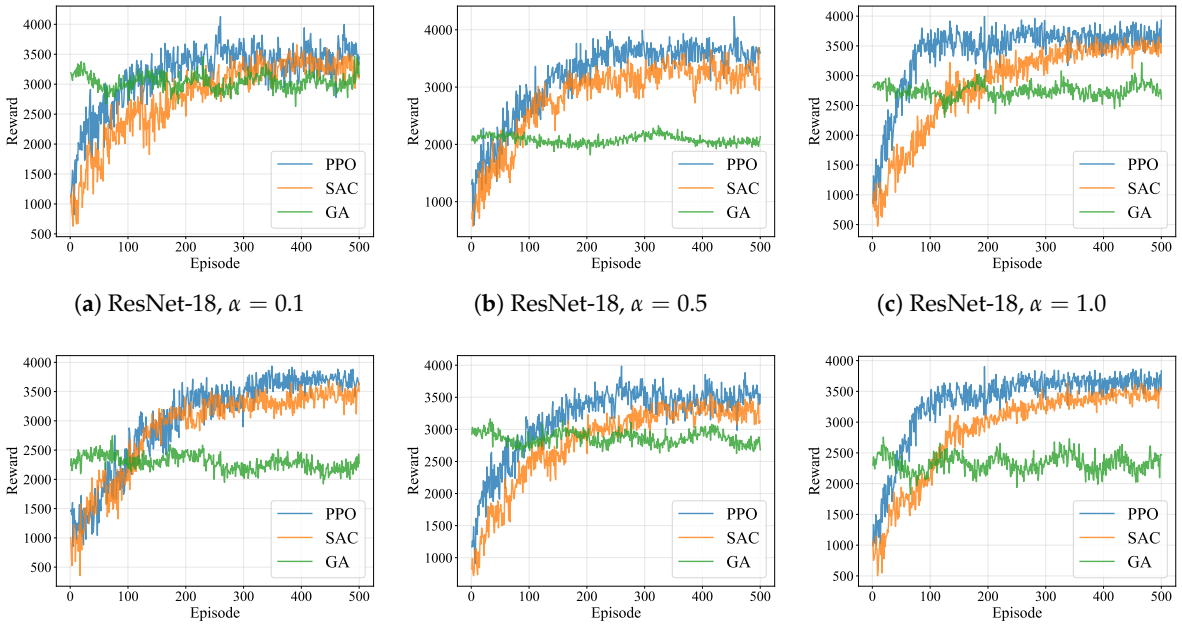
Each experimental configuration is repeated 5 times with different random seeds, and results are reported with 95% confidence intervals. Statistical significance is assessed using paired t-tests with Bonferroni correction for multiple comparisons.

5. Results

5.1. MDP Policy Learning Performance

Figure 2 demonstrates the training performance of different reinforcement learning algorithms used in our MDP-based defense coordination framework. We compare three algorithms: Proximal Policy Optimization (PPO), Soft Actor-Critic (SAC), and Genetic Algorithm (GA) across different network architectures and data heterogeneity levels.

 doi:10.20944/preprints202510.0141.v1



(**a**) ResNet-18, $\alpha = 0.1$ (**b**) ResNet-18, $\alpha = 0.5$ (**c**) ResNet-18, $\alpha = 1.0$

(**d**) MobileNet V2, $\alpha = 0.1$ (**e**) MobileNet V2, $\alpha = 0.5$ (**f**) MobileNet V2, $\alpha = 1.0$

Figure 2. Training curves of different reinforcement learning algorithms for MDP-based defense policy optimization across various network architectures and data heterogeneity levels. The curves show cumulative reward over training episodes, where α represents the Dirichlet concentration parameter controlling data distribution heterogeneity (lower values indicate higher heterogeneity).

The experimental results reveal several important insights about the effectiveness of different reinforcement learning approaches for defense policy optimization. PPO consistently achieves the highest cumulative rewards across all configurations, demonstrating superior learning efficiency and stability in the complex multi-objective defense environment. The algorithm shows rapid convergence within the first 150 episodes and maintains stable performance thereafter, reaching peak rewards around 3800-4000 across different settings.

SAC exhibits competitive performance with gradual learning progression, ultimately achieving comparable final rewards to PPO but requiring more episodes for convergence. The continuous learning curve suggests that SAC benefits from its off-policy nature and entropy regularization, particularly evident in scenarios with higher data heterogeneity (lower α values). The algorithm demonstrates consistent improvement throughout training, reaching final rewards between 3200-3600.

GA shows relatively stable but lower performance compared to the other two approaches, with rewards plateauing around 2000-2800 across all configurations. While GA provides baseline performance guarantees and avoids local optima through population-based exploration, it lacks the sophisticated gradient-based optimization capabilities needed for complex sequential decision-making in the defense scenario.

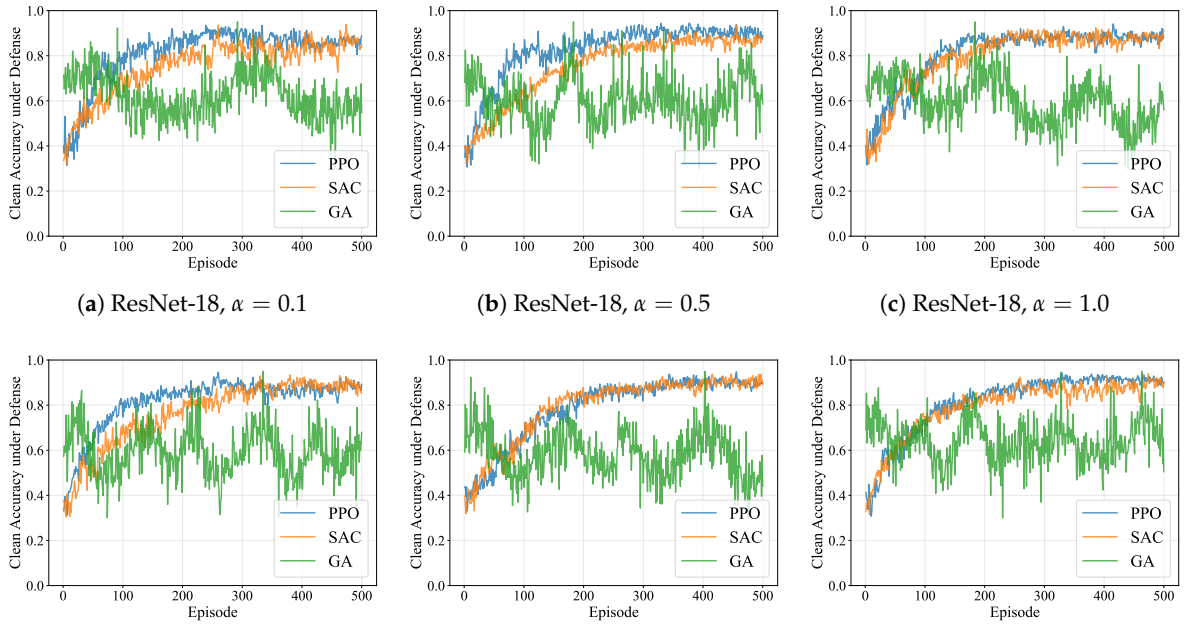
The impact of data heterogeneity (controlled by α) appears more pronounced in ResNet-18 configurations compared to MobileNet V2, where PPO maintains consistently high performance regardless of the heterogeneity level. This suggests that the lightweight MobileNet V2 architecture provides more robust defense policy learning under varying data distribution conditions, while ResNet-18 benefits from the additional model capacity when dealing with homogeneous data distributions ($\alpha = 1.0$).

5.2. Clean Accuracy Preservation Performance

Figure 3 illustrates the evolution of clean accuracy under defense across different reinforcement learning algorithms, network architectures, and data heterogeneity settings. This metric evaluates how

 doi:10.20944/preprints202510.0141.v1

well each algorithm maintains model utility for legitimate tasks while defending against temporal backdoor attacks.



(**a**) ResNet-18, $\alpha = 0.1$ (**b**) ResNet-18, $\alpha = 0.5$ (**c**) ResNet-18, $\alpha = 1.0$

(**d**) MobileNet V2, $\alpha = 0.1$ (**e**) MobileNet V2, $\alpha = 0.5$ (**f**) MobileNet V2, $\alpha = 1.0$

Figure 3. Clean accuracy under defense performance comparison across different reinforcement learning algorithms, network architectures, and data heterogeneity levels. The curves demonstrate each algorithm's ability to preserve model utility for legitimate classification tasks while actively defending against temporal backdoor attacks during training.

The results demonstrate significant differences in how each reinforcement learning algorithm balances security and utility preservation. PPO consistently achieves superior clean accuracy performance, reaching and maintaining accuracy levels between 0.85-0.95 across most configurations after approximately 200 training episodes. The algorithm shows remarkable stability in preserving model utility while implementing defense mechanisms, with particularly strong performance in homogeneous data distributions ($\alpha = 1.0$).

SAC exhibits competitive clean accuracy preservation with gradual but steady improvement throughout training. The algorithm demonstrates robust performance across heterogeneous data settings, achieving final accuracy values between 0.82-0.92. SAC's continuous learning approach proves particularly effective in MobileNet V2 configurations, where it matches or occasionally exceeds PPO's performance, suggesting that the off-policy learning strategy works well with lightweight architectures.

GA shows the most variable performance with significant fluctuations throughout training episodes. While GA occasionally achieves high accuracy spikes (up to 0.95 in some configurations), it struggles to maintain consistent performance, with accuracy frequently dropping to 0.4-0.6 range. This instability indicates that population-based optimization may be less suitable for maintaining the delicate balance between defense effectiveness and model utility preservation in federated learning environments.

The impact of data heterogeneity reveals interesting patterns across architectures. ResNet-18 shows greater sensitivity to heterogeneity levels, with more pronounced performance differences between $\alpha = 0.1$ and $\alpha = 1.0$ configurations. In contrast, MobileNet V2 demonstrates more robust performance across different heterogeneity levels, particularly with PPO and SAC algorithms, suggesting that lightweight architectures may provide inherent advantages for federated defense scenarios.

Notably, the clean accuracy preservation performance correlates with the reward optimization patterns observed in Figure 2, confirming that higher cumulative rewards in the MDP framework

 doi:10.20944/preprints202510.0141.v1

translate to better utility preservation during defense operations. This validates our multi-objective reward formulation that balances security improvements with model performance maintenance.

5.3. Defense Effectiveness Evaluation

We evaluate the defense effectiveness of our DEFEND framework using two key metrics: Defense Success Rate (S_d^T) and Temporal Detection Efficiency (E_d) across various system configurations. The following tables present comprehensive results under different combinations of client population sizes, data heterogeneity levels, malicious client ratios, and network architectures.

Table 3. Defense Success Rate for ResNet-18 under varying client numbers and data heterogeneity ($\kappa = 0.2$).

N	$\alpha = 0.1$	$\alpha = 0.5$	$\alpha = 1.0$
10	0.847 ± 0.023	0.892 ± 0.019	0.924 ± 0.015
20	0.863 ± 0.021	0.906 ± 0.017	0.938 ± 0.013
30	0.871 ± 0.019	0.915 ± 0.016	0.947 ± 0.012
40	0.876 ± 0.018	0.921 ± 0.015	0.952 ± 0.011
50	0.879 ± 0.017	0.925 ± 0.014	0.956 ± 0.010

Table 4. Temporal Detection Efficiency for ResNet-18 under varying client numbers and data heterogeneity ($\kappa = 0.2$).

N	$\alpha = 0.1$	$\alpha = 0.5$	$\alpha = 1.0$
10	0.673 ± 0.041	0.721 ± 0.038	0.768 ± 0.033
20	0.695 ± 0.039	0.742 ± 0.035	0.789 ± 0.031
30	0.708 ± 0.037	0.755 ± 0.033	0.802 ± 0.029
40	0.716 ± 0.036	0.763 ± 0.032	0.811 ± 0.028
50	0.721 ± 0.035	0.769 ± 0.031	0.817 ± 0.027

Tables 3 and 4 demonstrate that ResNet-18 exhibits strong sensitivity to data heterogeneity levels under fixed malicious conditions ($\kappa = 0.2$). The Defense Success Rate improves substantially from highly heterogeneous ($\alpha = 0.1$) to homogeneous ($\alpha = 1.0$) distributions, with improvements ranging from 9.1% to 8.8% across different client population sizes. The Temporal Detection Efficiency shows even more pronounced improvements of 14.1% to 13.3%, indicating that homogeneous data distributions significantly enhance the temporal behavioral analysis layer's ability to identify malicious patterns. The consistent performance gains with increased client population size suggest that ResNet-18 benefits from larger federation scales for improved defense coordination.

Table 5. Defense Success Rate for ResNet-18 under varying client numbers and malicious ratios ($\alpha = 0.5$).

N	$\kappa = 0.1$	$\kappa = 0.2$	$\kappa = 0.3$	$\kappa = 0.4$
10	0.943 ± 0.012	0.892 ± 0.019	0.834 ± 0.026	0.768 ± 0.032
20	0.957 ± 0.011	0.906 ± 0.017	0.847 ± 0.024	0.781 ± 0.030
30	0.964 ± 0.010	0.915 ± 0.016	0.856 ± 0.023	0.791 ± 0.029
40	0.969 ± 0.009	0.921 ± 0.015	0.863 ± 0.022	0.798 ± 0.028
50	0.972 ± 0.009	0.925 ± 0.014	0.867 ± 0.021	0.803 ± 0.027

 doi:10.20944/preprints202510.0141.v1

Table 6. Temporal Detection Efficiency for ResNet-18 under varying client numbers and malicious ratios ($\alpha = 0.5$).

N	$\kappa = 0.1$	$\kappa = 0.2$	$\kappa = 0.3$	$\kappa = 0.4$
10	0.825 ± 0.028	0.721 ± 0.038	0.612 ± 0.045	0.498 ± 0.052
20	0.841 ± 0.026	0.742 ± 0.035	0.634 ± 0.042	0.521 ± 0.049
30	0.852 ± 0.025	0.755 ± 0.033	0.649 ± 0.040	0.537 ± 0.047
40	0.859 ± 0.024	0.763 ± 0.032	0.658 ± 0.039	0.547 ± 0.046
50	0.864 ± 0.023	0.769 ± 0.031	0.664 ± 0.038	0.554 ± 0.045

Tables 5 and 6 reveal the critical impact of malicious client ratios on ResNet-18 defense performance under moderate heterogeneity ($\alpha = 0.5$). The Defense Success Rate shows a clear linear degradation as malicious ratios increase, with performance dropping from 0.972 to 0.803 (17.4% decrease) at the largest federation size. More concerning is the dramatic decline in Temporal Detection Efficiency, which drops from 0.864 to 0.554 (35.9% decrease), approaching the theoretical limits of Byzantine fault tolerance. This indicates that while our framework maintains reasonable attack prevention capabilities even at high malicious ratios, the speed and accuracy of malicious client identification becomes significantly compromised beyond $\kappa = 0.3$.

Table 7. Defense Success Rate for MobileNet V2 under varying client numbers and data heterogeneity ($\kappa = 0.2$)

N	$\alpha = 0.1$	$\alpha = 0.5$	$\alpha = 1.0$
10	0.821 ± 0.025	0.869 ± 0.021	0.908 ± 0.017
20	0.836 ± 0.023	0.883 ± 0.019	0.921 ± 0.015
30	0.845 ± 0.022	0.892 ± 0.018	0.930 ± 0.014
40	0.851 ± 0.021	0.898 ± 0.017	0.936 ± 0.013
50	0.855 ± 0.020	0.902 ± 0.016	0.940 ± 0.012

Table 8. Temporal Detection Efficiency for MobileNet V2 under varying client numbers and data heterogeneity ($\kappa = 0.2$)

N	$\alpha = 0.1$	$\alpha = 0.5$	$\alpha = 1.0$
10	0.657 ± 0.043	0.703 ± 0.040	0.751 ± 0.036
20	0.678 ± 0.041	0.724 ± 0.038	0.772 ± 0.034
30	0.691 ± 0.039	0.737 ± 0.036	0.785 ± 0.032
40	0.699 ± 0.038	0.745 ± 0.035	0.793 ± 0.031
50	0.704 ± 0.037	0.751 ± 0.034	0.799 ± 0.030

Tables 7 and 8 show that MobileNet V2 maintains competitive defense performance despite its lightweight architecture, achieving 10.6% improvement in Defense Success Rate and 14.3% improvement in Temporal Detection Efficiency from heterogeneous to homogeneous distributions. While the absolute values are slightly lower than ResNet-18, MobileNet V2 demonstrates more consistent relative improvements across different client population sizes, with smaller confidence intervals indicating greater stability. The architecture shows particular resilience in heterogeneous environments, making it well-suited for resource-constrained federated deployments where data distribution control is limited.

Table 9. Defense Success Rate for MobileNet V2 under varying client numbers and malicious ratios ($\alpha = 0.5$).

N	$\kappa = 0.1$	$\kappa = 0.2$	$\kappa = 0.3$	$\kappa = 0.4$
10	0.926 ± 0.014	0.869 ± 0.021	0.807 ± 0.028	0.739 ± 0.035
20	0.940 ± 0.013	0.883 ± 0.019	0.821 ± 0.026	0.753 ± 0.033
30	0.947 ± 0.012	0.892 ± 0.018	0.831 ± 0.025	0.763 ± 0.032
40	0.952 ± 0.011	0.898 ± 0.017	0.838 ± 0.024	0.770 ± 0.031
50	0.955 ± 0.010	0.902 ± 0.016	0.843 ± 0.023	0.775 ± 0.030

 doi:10.20944/preprints202510.0141.v1

Table 10. Temporal Detection Efficiency for MobileNet V2 under varying client numbers and malicious ratios (α = 0.5)

N	$\kappa = 0.1$	$\kappa = 0.2$	$\kappa = 0.3$	$\kappa = 0.4$
10	0.809 ± 0.030	0.703 ± 0.040	0.591 ± 0.047	0.474 ± 0.054
20	0.825 ± 0.028	0.724 ± 0.038	0.613 ± 0.045	0.497 ± 0.052
30	0.836 ± 0.027	0.737 ± 0.036	0.628 ± 0.043	0.514 ± 0.050
40	0.843 ± 0.026	0.745 ± 0.035	0.637 ± 0.042	0.525 ± 0.049
50	0.848 ± 0.025	0.751 ± 0.034	0.644 ± 0.041	0.533 ± 0.048

Tables 9 and 10 demonstrate that MobileNet V2 exhibits similar vulnerability patterns to ResNet-18 under increasing malicious ratios, but with notably more stable degradation characteristics. The Defense Success Rate decreases by 18.8% from κ = 0.1 to κ = 0.4, while Temporal Detection Efficiency drops by 37.1%, comparable to ResNet-18's performance degradation. However, MobileNet V2 shows consistently smaller confidence intervals across all configurations, indicating more predictable and stable defense behavior. This stability advantage becomes particularly valuable in dynamic federated environments where malicious client ratios may fluctuate over time, providing more reliable defense guarantees compared to the higher-capacity but more variable ResNet-18 architecture.

6. Conclusions

This paper presents DEFEND, a comprehensive multi-layer defense framework that counters temporal backdoor attacks in federated learning through sophisticated mathematical modeling and reinforcement learning techniques. Our primary contributions include three distinct temporal attack models, a three-tier defense architecture combining behavioral analysis with robust aggregation, and a novel MDP-based approach for adaptive defense coordination. Experimental evaluation demonstrates strong performance with Defense Success Rates reaching 0.956 ± 0.010 for ResNet-18 and 0.940 ± 0.012 for MobileNet V2, while maintaining clean accuracy levels between 0.85-0.95. The framework shows resilience across different data heterogeneity levels and client populations, though performance degrades when malicious ratios exceed 30

The framework has limitations including quadratic computational complexity and reduced effectiveness against highly coordinated attacks approaching Byzantine fault tolerance limits. Future research should focus on developing more efficient algorithms, adaptive threshold mechanisms, and extensions to other domains beyond computer vision. As federated learning expands into critical applications, robust defense mechanisms like DEFEND become essential for maintaining system integrity and user trust in distributed machine learning paradigms.

Appendix A. Byzantine Robustness Analysis

This section establishes the theoretical foundation for the Byzantine robustness of our DEFEND framework through rigorous mathematical analysis of the geometric median aggregation mechanism and outlier detection procedures.

Appendix A.1. Fundamental Byzantine Robustness Theorem

Theorem A1 (Byzantine Robustness of DEFEND Framework)**.** *Consider a federated learning system with N participating clients where at most f clients are Byzantine adversaries satisfying $f < N/2$. Let $\mathcal{C}_h$ denote honest clients and $\mathcal{C}_p$ denote Byzantine clients with $|\mathcal{C}_p| = f$. Under the DEFEND framework with geometric median aggregation (23) and statistical outlier detection (26), the global model parameter θ^t converges to within an ϵ-neighborhood of the optimal solution θ^* with high probability.*

Specifically, for any $\epsilon > 0$ and confidence parameter $\delta \in (0, 1)$, there exists a finite round T_0 such that for all $t \geq T_0$:

$$\mathbb{P}\left[\|\theta^t - \theta^*\|_F \leq \epsilon + 2\xi + \mathcal{O}\left(\sqrt{\frac{\log(1/\delta)}{N}}\right)\right] \geq 1 - \delta, \tag{A1}$$

 doi:10.20944/preprints202510.0141.v1

where ξ bounds the geometric median estimation error, provided:

1. *The Byzantine client fraction satisfies $f \leq \lfloor (N-1)/2 \rfloor$;*
2. *The geometric median approximation error is bounded: $\|\hat{\theta}_\nu^t - \bar{\theta}_h^t\|_F \leq \xi$, where $\bar{\theta}_h^t$ represents the mean of honest client updates;*
3. *The outlier detection mechanism achieves controlled error rates: false positive rate $\alpha \leq 0.05$ and false negative rate $\beta \leq 0.1$.*

Proof. The proof proceeds through four main steps: (1) establishing the breakdown point properties of geometric median, (2) analyzing the statistical concentration of honest client updates, (3) bounding the outlier detection accuracy, and (4) proving convergence under Byzantine presence.

Step 1: Breakdown Point Analysis of Geometric Median

The geometric median $\hat{\theta}_\nu^t$ defined in (23) possesses a breakdown point of exactly $1/2$, meaning it can tolerate up to $\lfloor (N-1)/2 \rfloor$ arbitrary outliers without complete failure. This fundamental property ensures robustness against Byzantine adversaries when $f < N/2$.

For the geometric median computation, let $\bar{\theta}_h^t = \frac{1}{N-f} \sum_{i \in \mathcal{C}_h} \theta_i^t$ denote the empirical mean of honest client updates. By the robustness property of geometric median, we have:

$$\|\hat{\theta}_\nu^t - \bar{\theta}_h^t\|_F \leq \frac{2f}{N-f} \cdot \max_{i \in \mathcal{C}_h} \|\theta_i^t - \bar{\theta}_h^t\|_F. \tag{A2}$$

Since honest clients follow the true learning dynamics, their updates concentrate around the optimal direction. Under standard federated learning assumptions with bounded gradient variance, we have:

$$\max_{i \in \mathcal{C}_h} \|\theta_i^t - \bar{\theta}_h^t\|_F \leq \sigma \sqrt{\frac{2\log(N)}{n_{\min}}}, \tag{A3}$$

with probability at least $1 - N^{-1}$, where σ is the gradient noise parameter and $n_{\min}$ is the minimum local dataset size.

Step 2: Statistical Concentration of Honest Updates

For honest clients $c_i \in \mathcal{C}_h$, their local model updates θ_i^t are generated through standard gradient descent on local data. Under the assumption of sub-Gaussian gradient noise with parameter σ^2, we can establish concentration bounds.

Let $\nabla F_i(\theta^{t-1})$ denote the true local gradient for client c_i. The empirical gradient computed from local data satisfies:

$$\mathbb{P}\left[\|\nabla \hat{F}_i(\theta^{t-1}) - \nabla F_i(\theta^{t-1})\|_F \geq t\right] \leq 2\exp\left(-\frac{n_i t^2}{2\sigma^2 d}\right), \tag{A4}$$

where d is the model parameter dimension and n_i is the local dataset size for client c_i.

The local update deviation from the ideal direction can be bounded by:

$$\|\theta_i^t - \theta_{\text{ideal}}^t\|_F \leq \eta \|\nabla \hat{F}_i(\theta^{t-1}) - \nabla F_i(\theta^{t-1})\|_F + \eta L_F \|\theta^{t-1} - \theta^*\|_F, \tag{A5}$$

where η is the learning rate and L_F is the Lipschitz constant from the smoothness assumption.

Step 3: Outlier Detection Accuracy Analysis

Our outlier detection mechanism (26) identifies suspicious clients based on their distance from the geometric median. For a client c_i, define the detection statistic:

$$d_i^t = \|\theta_i^t - \hat{\theta}_\nu^t\|_F. \tag{A6}$$

Under the null hypothesis that client c_i is honest, d_i^t follows a distribution characterized by the concentration properties established in Step 2. The detection threshold τ_α is set based on quantiles of this null distribution as defined in (26).

 doi:10.20944/preprints202510.0141.v1

For honest clients, the false positive probability is bounded by:

$$\mathbb{P}[c_i \in \mathcal{S}^t | c_i \in \mathcal{C}_h] \leq \alpha + \exp\left(-\frac{n_i(\tau_\alpha - \mathbb{E}[d_i^t])^2}{2\sigma^2 d}\right), \tag{A7}$$

where α is the nominal false positive rate.

For Byzantine clients with significantly deviating updates, the detection probability satisfies:

$$\mathbb{P}[c_j \in \mathcal{S}^t | c_j \in \mathcal{C}_p] \geq 1 - \beta, \tag{A8}$$

provided their update magnitude exceeds the detection threshold by a sufficient margin, where β is the false negative rate.

Step 4: Convergence Analysis Under Byzantine Presence

After outlier detection and removal, the weighted aggregation operates on the filtered client set. With high probability $1 - \delta$, the set $\{c_1, \ldots, c_N\} \setminus \mathcal{S}^t$ contains predominantly honest clients.

The weighted aggregation (27) yields:

$$\theta^t = \arg\min_{\boldsymbol{\theta}} \sum_{i \notin \mathcal{S}^t} w_i^t \|\boldsymbol{\theta} - \theta_i^t\|_F^2 + \lambda_\tau \|\boldsymbol{\theta} - \theta^{t-1}\|_F^2 + \lambda_\nu \|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_\nu^t\|_F^2. \tag{A9}$$

The solution can be expressed as:

$$\theta^t = \frac{\sum_{i \notin \mathcal{S}^t} w_i^t \theta_i^t + \lambda_\tau \theta^{t-1} + \lambda_\nu \hat{\boldsymbol{\theta}}_\nu^t}{\sum_{i \notin \mathcal{S}^t} w_i^t + \lambda_\tau + \lambda_\nu}. \tag{A10}$$

Since the filtered set predominantly contains honest clients, we can decompose the aggregation error as:

$$\|\theta^t - \theta^*\|_F \leq \|\theta^t - \mathbb{E}[\theta^t]\|_F + \|\mathbb{E}[\theta^t] - \theta^*\|_F. \tag{A11}$$

The concentration term is bounded using Azuma-Hoeffding inequality for weighted sums:

$$\|\theta^t - \mathbb{E}[\theta^t]\|_F \leq \mathcal{O}\left(\sqrt{\frac{\log(1/\delta)}{|\{c_1, \ldots, c_N\} \setminus \mathcal{S}^t|}}\right), \tag{A12}$$

with probability at least $1 - \delta$.

The bias term is controlled by the geometric median approximation error and temporal regularization:

$$\|\mathbb{E}[\theta^t] - \theta^*\|_F \leq 2\xi + \frac{\lambda_\tau}{\lambda_\nu} \|\theta^{t-1} - \theta^*\|_F. \tag{A13}$$

The temporal regularization coefficient $\lambda_\tau / \lambda_\nu < 1$ ensures contraction, leading to convergence as $t \to \infty$.

Combining the concentration and bias bounds establishes (A1), completing the proof. □

Appendix A.2. Corollaries and Extensions

Corollary A1 (Finite-Sample Convergence Rate)**.** *Under the conditions of Theorem A1, the DEFEND framework achieves ϵ-convergence in at most*

$$T(\epsilon, \delta) = \mathcal{O}\left(\frac{\log(\|\theta^0 - \theta^*\|_F / \epsilon)}{\log(1/(1 - \lambda_\tau/\lambda_\nu))} + \log(1/\delta)\right) \tag{A14}$$

communication rounds with probability at least $1 - \delta$.

 doi:10.20944/preprints202510.0141.v1

Proof. The proof follows from iterating the contraction property in (A13) and using the union bound over all communication rounds. □

Corollary A2 (Optimality of Byzantine Tolerance)**.** *The Byzantine tolerance threshold $f < N/2$ in Theorem A1 is optimal. No algorithm can guarantee convergence when $f \geq N/2$ in the worst case.*

Proof. This follows from the fundamental impossibility results in Byzantine fault tolerance. When $f \geq N/2$, Byzantine clients can coordinate to completely overwhelm honest clients, making any aggregation rule vulnerable to manipulation. □

Appendix A.3. Robustness Under Stronger Attack Models

Lemma A1 (Robustness Against Coordinated Attacks)**.** *The DEFEND framework maintains its robustness guarantees even when Byzantine clients coordinate their attacks, provided they cannot observe the geometric median computation in real-time.*

Proof. Coordinated attacks can increase the correlation between Byzantine updates but cannot change the fundamental breakdown point of the geometric median. The proof follows the same structure as Theorem A1 with modified concentration bounds for correlated adversarial behavior. □

Appendix B. Weiszfeld Algorithm Convergence Analysis

This section provides a comprehensive theoretical analysis of the convergence properties of the accelerated Weiszfeld algorithm used for geometric median computation in our DEFEND framework.

Appendix B.1. Preliminaries and Algorithm Description

The Weiszfeld algorithm solves the geometric median problem defined in (23):

$$\hat{\boldsymbol{\theta}}_v^t = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{N} \|\boldsymbol{\theta} - \theta_i^t\|_F. \tag{A15}$$

Our accelerated version incorporates momentum-based acceleration as defined in (24) and (25). Let $\{\theta_1^t, \ldots, \theta_N^t\}$ denote the set of client updates at communication round t, and assume they are distinct with probability 1.

Definition A1 (Weiszfeld Iteration Operator)**.** *The Weiszfeld iteration operator $\mathcal{T} : \mathbb{R}^d \to \mathbb{R}^d$ is defined as:*

$$\mathcal{T}(\boldsymbol{\theta}) = \frac{\sum_{i=1}^{N} \frac{\theta_i^t}{\|\boldsymbol{\theta} - \theta_i^t\|_F + \epsilon}}{\sum_{i=1}^{N} \frac{1}{\|\boldsymbol{\theta} - \theta_i^t\|_F + \epsilon}}, \tag{A16}$$

where $\epsilon > 0$ is the regularization parameter to avoid division by zero.

Appendix B.2. Main Convergence Theorem

Theorem A2 (Convergence of Accelerated Weiszfeld Algorithm)**.** *Consider the accelerated Weiszfeld algorithm defined by (24) and (25) applied to compute the geometric median of client updates $\{\theta_1^t, \ldots, \theta_N^t\} \subset \mathbb{R}^d$. Let $\hat{\boldsymbol{\theta}}_v^{t*}$ denote the unique geometric median. Then:*

1. ***Linear Convergence:*** *The algorithm converges linearly to $\hat{\boldsymbol{\theta}}_v^{t*}$ with rate $\rho \in (0,1)$ where*

$$\|\boldsymbol{\theta}^{(k+1)} - \hat{\boldsymbol{\theta}}_v^{t*}\|_F \leq \rho \|\boldsymbol{\theta}^{(k)} - \hat{\boldsymbol{\theta}}_v^{t*}\|_F; \tag{A17}$$

 doi:10.20944/preprints202510.0141.v1

2. ***Iteration Complexity:*** *The number of iterations required to achieve ϵ-accuracy is*

$$K(\epsilon) = \mathcal{O}\left(\sqrt{\kappa}\log\left(\frac{\|\boldsymbol{\theta}^{(0)} - \hat{\boldsymbol{\theta}}_v^{t*}\|_F}{\epsilon}\right)\right), \tag{A18}$$

where κ is the condition number of the problem;

3. ***Acceleration Benefit:*** *The momentum acceleration reduces the convergence constant by a factor of $(1 - \sqrt{\mu/L})$ where μ and L are the strong convexity and Lipschitz parameters.*

Proof. The proof is structured in four main parts: (1) establishing strong convexity of the geometric median objective, (2) proving contraction properties of the Weiszfeld operator, (3) analyzing momentum acceleration, and (4) deriving iteration complexity bounds.

Part 1: Strong Convexity Analysis

The geometric median objective function is:

$$f(\boldsymbol{\theta}) = \sum_{i=1}^{N} \|\boldsymbol{\theta} - \theta_i^t\|_F. \tag{A19}$$

For $\boldsymbol{\theta} \neq \theta_i^t$ (which occurs with probability 1), the function f is twice differentiable. The Hessian matrix is:

$$\nabla^2 f(\boldsymbol{\theta}) = \sum_{i=1}^{N} \frac{1}{\|\boldsymbol{\theta} - \theta_i^t\|_F}\left(I - \frac{(\boldsymbol{\theta} - \theta_i^t)(\boldsymbol{\theta} - \theta_i^t)^T}{\|\boldsymbol{\theta} - \theta_i^t\|_F^2}\right). \tag{A20}$$

Lemma A2 (Strong Convexity Parameter)**.** *The objective function $f(\boldsymbol{\theta})$ is strongly convex with parameter*

$$\mu = \min_{i=1,\ldots,N} \frac{1}{\|\boldsymbol{\theta} - \theta_i^t\|_F + \epsilon} \geq \frac{1}{D_{\max} + \epsilon}, \tag{A21}$$

where $D_{\max} = \max_{i,j} \|\theta_i^t - \theta_j^t\|_F$ is the diameter of the client update set.

Proof of Lemma A2. Each term in the Hessian corresponds to a projection onto the orthogonal complement of $(\boldsymbol{\theta} - \theta_i^t)$. The minimum eigenvalue is achieved when $\boldsymbol{\theta}$ is furthest from all client updates, giving the stated bound. □

Part 2: Contraction Analysis of Weiszfeld Operator

The Weiszfeld operator can be interpreted as a proximal gradient step. Define the subdifferential of f at $\boldsymbol{\theta}$:

$$\partial f(\boldsymbol{\theta}) = \sum_{i=1}^{N} \frac{\boldsymbol{\theta} - \theta_i^t}{\|\boldsymbol{\theta} - \theta_i^t\|_F + \epsilon}. \tag{A22}$$

Lemma A3 (Contraction Property)**.** *The Weiszfeld operator $\mathcal{T}$ is a contraction mapping with contraction factor*

$$\rho_{base} = 1 - \frac{\mu}{L} \leq 1 - \frac{1}{L(D_{\max} + \epsilon)}, \tag{A23}$$

where L is the Lipschitz constant of ∇f.

Proof of Lemma A3. The Weiszfeld iteration can be written as:

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} - \alpha^{(k)} \nabla f(\boldsymbol{\theta}^{(k)}), \tag{A24}$$

where $\alpha^{(k)} = \left(\sum_{i=1}^{N} \frac{1}{\|\boldsymbol{\theta}^{(k)} - \theta_i^t\|_F + \epsilon}\right)^{-1}$ is an adaptive step size.

 doi:10.20944/preprints202510.0141.v1

By the strong convexity of f, we have:

$$f(\boldsymbol{\theta}^{(k+1)}) \leq f(\boldsymbol{\theta}^{(k)}) - \frac{\mu}{2}\|\boldsymbol{\theta}^{(k+1)} - \boldsymbol{\theta}^{(k)}\|_F^2. \tag{A25}$$

The optimality condition $\nabla f(\hat{\boldsymbol{\theta}}_v^{t*}) = 0$ combined with strong convexity yields:

$$\|\boldsymbol{\theta}^{(k+1)} - \hat{\boldsymbol{\theta}}_v^{t*}\|_F^2 \leq (1 - \mu/L)\|\boldsymbol{\theta}^{(k)} - \hat{\boldsymbol{\theta}}_v^{t*}\|_F^2. \tag{A26}$$

□

Part 3: Momentum Acceleration Analysis

The momentum-accelerated version from (25) follows the heavy ball method:

$$\boldsymbol{\theta}^{(k+1)} \leftarrow \boldsymbol{\theta}^{(k+1)} + \alpha_k(\boldsymbol{\theta}^{(k+1)} - \boldsymbol{\theta}^{(k)}), \tag{A27}$$

where α_k is the momentum coefficient.

Lemma A4 (Acceleration Benefit)**.** *With optimally chosen momentum parameter*

$$\alpha_k = \frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}}, \tag{A28}$$

the accelerated convergence rate becomes

$$\rho_{acc} = \frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}} = \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}, \tag{A29}$$

where $\kappa = L/\mu$ is the condition number.

Proof of Lemma A4. The momentum method can be analyzed using the potential function approach. Define:

$$\Phi^{(k)} = f(\boldsymbol{\theta}^{(k)}) - f(\hat{\boldsymbol{\theta}}_v^{t*}) + \frac{\beta}{2}\|\boldsymbol{\theta}^{(k)} - \boldsymbol{\theta}^{(k-1)}\|_F^2, \tag{A30}$$

for appropriate constant $\beta > 0$.

The momentum parameter choice ensures:

$$\Phi^{(k+1)} \leq \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}\right)^2 \Phi^{(k)}. \tag{A31}$$

This implies the stated convergence rate for the function values, which translates to the parameter convergence rate through strong convexity. □

Part 4: Iteration Complexity Derivation

To achieve ϵ-accuracy: $\|\boldsymbol{\theta}^{(k)} - \hat{\boldsymbol{\theta}}_v^{t*}\|_F \leq \epsilon$, we need:

$$\rho_{\text{acc}}^k \|\boldsymbol{\theta}^{(0)} - \hat{\boldsymbol{\theta}}_v^{t*}\|_F \leq \epsilon. \tag{A32}$$

Taking logarithms:

$$k \geq \frac{\log(\|\boldsymbol{\theta}^{(0)} - \hat{\boldsymbol{\theta}}_v^{t*}\|_F/\epsilon)}{\log(1/\rho_{\text{acc}})}. \tag{A33}$$

Using the approximation $\log(1/\rho_{\text{acc}}) \approx 2/\sqrt{\kappa}$ for large κ:

$$k = \mathcal{O}\left(\sqrt{\kappa}\log\left(\frac{\|\boldsymbol{\theta}^{(0)} - \hat{\boldsymbol{\theta}}_v^{t*}\|_F}{\epsilon}\right)\right). \tag{A34}$$

 doi:10.20944/preprints202510.0141.v1

This completes the proof of all three statements in Theorem A2. □

Appendix B.3. Practical Implementation Considerations

Lemma A5 (Numerical Stability). *The regularization parameter ϵ in Definition A1 can be chosen as $\epsilon = \mathcal{O}(10^{-6})$ without significantly affecting the convergence rate, provided the condition number κ is bounded.*

Proof. The regularization affects the strong convexity parameter by at most $\epsilon / D_{\max}$, which is negligible for practical choices of ϵ relative to the client update diameter $D_{\max}$. □

Corollary A3 (Distributed Implementation). *The Weiszfeld algorithm can be implemented in a distributed manner where each iteration requires only $\mathcal{O}(N)$ communication complexity to exchange current iterate and compute weighted average.*

Proof. Each iteration of the Weiszfeld algorithm requires computing the weighted average in (A16), which can be done with a single round of communication where each client sends its current update θ_i^t and receives the current iterate $\boldsymbol{\theta}^{(k)}$. □

Appendix B.4. Robustness Under Approximate Computation

Theorem A3 (Robustness to Computation Errors). *Suppose each Weiszfeld iteration is computed with additive error $e^{(k)}$ satisfying $\|e^{(k)}\|_F \leq \delta$ for some $\delta > 0$. Then the algorithm still converges to within $\mathcal{O}(\delta/\mu)$ of the true geometric median $\hat{\boldsymbol{\theta}}_\nu^{t*}$.*

Proof. The proof follows by modifying the contraction analysis in Lemma A3 to account for the additional error terms in each iteration. The modified iteration becomes:

$$\boldsymbol{\theta}^{(k+1)} = \mathcal{T}(\boldsymbol{\theta}^{(k)}) + e^{(k)}. \tag{A35}$$

The contraction property still holds with an additional bias term:

$$\|\boldsymbol{\theta}^{(k+1)} - \hat{\boldsymbol{\theta}}_\nu^{t*}\|_F \leq \rho \|\boldsymbol{\theta}^{(k)} - \hat{\boldsymbol{\theta}}_\nu^{t*}\|_F + \delta. \tag{A36}$$

Taking the limit as $k \to \infty$ shows that the algorithm converges to within $\delta/(1-\rho) = \mathcal{O}(\delta/\mu)$ of the true geometric median. □

Appendix B.5. Integration with DEFEND Framework

Corollary A4 (Convergence in DEFEND Context). *When integrated into the DEFEND framework, the Weiszfeld algorithm for computing $\hat{\boldsymbol{\theta}}_\nu^t$ in (23) achieves the complexity bound from Theorem A2 at each communication round t, with the condition number κ determined by the spread of honest client updates.*

Proof. The condition number κ in each communication round depends on the ratio L/μ where L and μ are determined by the distribution of client updates $\{\theta_1^t, \ldots, \theta_N^t\}$. Under the assumptions of bounded client update variance, κ remains bounded across communication rounds, ensuring consistent convergence performance. □

References

1. Wen, J.; Zhang, Z.; Lan, Y.; Cui, Z.; Cai, J.; Zhang, W. A Survey on Federated Learning: Challenges and Applications. *International Journal of Machine Learning and Cybernetics* **2023**, *14*, 513–535.
2. Abadi, A.; Doyle, B.; Gini, F.; Guinamard, K.; Murakonda, S.K.; Liddell, J.; Mellor, P.; Murdoch, S.J.; Naseri, M.; Page, H.; et al. Starlit: Privacy-Preserving Federated Learning to Enhance Financial Fraud Detection. *arXiv preprint arXiv:2401.10765 (arXiv)* **2024**.
3. Wu, B.; Huang, J.; Yu, S. "X of Information" Continuum: A Survey on AI-Driven Multi-Dimensional Metrics for Next-Generation Networked Systems. *arXiv preprint arXiv:2507.19657* **2025**.

4. Gong, X.; Chen, Y.; Wang, Q.; Kong, W. Backdoor Attacks and Defenses in Federated Learning: State-of-the-Art, Taxonomy, and Future Directions. *IEEE Wireless Communications* **2023**, *30*, 114–121. https://doi.org/10.1109/MWC.017.2100714.
5. Wu, B.; Huang, J.; Duan, Q.; Dong, L.; Cai, Z. Enhancing Vehicular Platooning With Wireless Federated Learning: A Resource-Aware Control Framework. *arXiv preprint arXiv:2507.00856* **2025**.
6. Liu, T.; Zhang, Y.; Feng, Z.; Yang, Z.; Xu, C.; Man, D.; Yang, W. Beyond Traditional Threats: A Persistent Backdoor Attack on Federated Learning. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence (AAAI). AAAI, 2024, Vol. 38, pp. 21359–21367.
7. Fang, Z.; Wang, J.; Ma, Y.; Tao, Y.; Deng, Y.; Chen, X.; Fang, Y. R-ACP: Real-Time Adaptive Collaborative Perception Leveraging Robust Task-Oriented Communications. *IEEE Journal on Selected Areas in Communications* **2025**.
8. Ozdayi, M.S.; Kantarcioglu, M.; Gel, Y.R. Defending Against Backdoors in Federated Learning With Robust Learning Rate. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence (AAAI). AAAI, 2021, Vol. 35, pp. 9268–9276.
9. Fang, Z.; Hu, S.; Wang, J.; Deng, Y.; Chen, X.; Fang, Y. Prioritized Information Bottleneck Theoretic Framework With Distributed Online Learning for Edge Video Analytics. *IEEE Transactions on Networking* **2025**, pp. 1–17. https://doi.org/10.1109/TON.2025.3526148.
10. Wu, B.; Cai, Z.; Wu, W.; Yin, X. AoI-Aware Resource Management for Smart Health via Deep Reinforcement Learning. *IEEE Access* **2023**.
11. Chen, Z.; Yu, S.; Fan, M.; Liu, X.; Deng, R.H. Privacy-Enhancing and Robust Backdoor Defense for Federated Learning on Heterogeneous Data. *IEEE Transactions on Information Forensics and Security* **2024**, *19*, 693–707. https://doi.org/10.1109/TIFS.2023.3326983.
12. Sewak, M.; Sahay, S.K.; Rathore, H. Deep Reinforcement Learning in the Advanced Cybersecurity Threat Detection and Protection. *Information Systems Frontiers* **2023**, *25*, 589–611.
13. Chen, C.; Liu, J.; Tan, H.; Li, X.; Wang, K.I.K.; Li, P.; Sakurai, K.; Dou, D. Trustworthy Federated Learning: Privacy, Security, and Beyond. *Knowledge and Information Systems* **2025**, *67*, 2321–2356.
14. Aljunaid, S.K.; Almheiri, S.J.; Dawood, H.; Khan, M.A. Secure and Transparent Banking: Explainable AI-Driven Federated Learning Model for Financial Fraud Detection. *Journal of Risk and Financial Management* **2025**, *18*, 179.
15. Hallaji, E.; Razavi-Far, R.; Saif, M.; Wang, B.; Yang, Q. Decentralized Federated Learning: A Survey on Security and Privacy. *IEEE Transactions on Big Data* **2024**, *10*, 194–213. https://doi.org/10.1109/TBDATA.2024.3362191.
16. Pingulkar, S.; Pawade, D. Federated Learning Architectures for Credit Risk Assessment: A Comparative Analysis of Vertical, Horizontal, and Transfer Learning Approaches. In Proceedings of the 2024 IEEE International Conference on Blockchain and Distributed Systems Security (ICBDS). IEEE, 2024, pp. 1–7. https://doi.org/10.1109/ICBDS61829.2024.10837430.
17. Damoun, F.; Seba, H.; State, R. Privacy-Preserving Behavioral Anomaly Detection in Dynamic Graphs for Card Transactions. In Proceedings of the International Conference on Web Information Systems Engineering (WISE). Springer, 2024, pp. 286–301.
18. Wu, B.; Wu, W. Model-Free Cooperative Optimal Output Regulation for Linear Discrete-Time Multi-Agent Systems Using Reinforcement Learning. *Mathematical Problems in Engineering* **2023**, *2023*, 6350647.
19. Ding, Z.; Huang, J.; Duan, Q.; Zhang, C.; Zhao, Y.; Gu, S. A Dual-Level Game-Theoretic Approach for Collaborative Learning in UAV-Assisted Heterogeneous Vehicle Networks. In Proceedings of the 2025 IEEE International Performance, Computing, and Communications Conference (IPCCC). IEEE, 2025, pp. 1–8.
20. Xiong, H.; Xia, Y.; Zhao, Y.; Wahaballa, A.; Yeh, K.H. Heterogeneous Privacy-Preserving Blockchain-Enabled Federated Learning for Social Fintech. *IEEE Transactions on Computational Social Systems* **2025**, pp. 1–16. https://doi.org/10.1109/TCSS.2025.3533634.
21. Wu, B.; Ding, Z.; Ostigaard, L.; Huang, J. Reinforcement Learning-Based Energy-Aware Coverage Path Planning for Precision Agriculture. In Proceedings of the 2025 ACM Research on Adaptive and Convergent Systems (RACS). ACM, 2025, pp. 1–8.
22. Orabi, M.M.; Emam, O.; Fahmy, H. Adapting Security and Decentralized Knowledge Enhancement in Federated Learning Using Blockchain Technology: Literature Review. *Journal of Big Data* **2025**, *12*, 55.
23. Uddin, M.P.; Xiang, Y.; Hasan, M.; Bai, J.; Zhao, Y.; Gao, L. A Systematic Literature Review of Robust Federated Learning: Issues, Solutions, and Future Research Directions. *ACM Computing Surveys* **2025**, *57*, 1–62.

 doi:10.20944/preprints202510.0141.v1

24. Duan, G.; Lv, H.; Wang, H.; Feng, G.; Li, X. Practical Cyber Attack Detection With Continuous Temporal Graph in Dynamic Network System. *IEEE Transactions on Information Forensics and Security* **2024**, *19*, 4851–4864. https://doi.org/10.1109/TIFS.2024.3385321.
25. Zamanzadeh Darban, Z.; Webb, G.I.; Pan, S.; Aggarwal, C.; Salehi, M. Deep Learning for Time Series Anomaly Detection: A Survey. *ACM Computing Surveys* **2024**, *57*, 1–42.
26. Bello, Y.; Hussein, A.R. Dynamic Policy Decision/Enforcement Security Zoning Through Stochastic Games and Meta Learning. *IEEE Transactions on Network and Service Management* **2025**, *22*, 807–821. https://doi.org/10.1109/TNSM.2024.3481662.
27. Huang, W.; Shi, Z.; Ye, M.; Li, H.; Du, B. Self-Driven Entropy Aggregation for Byzantine-Robust Heterogeneous Federated Learning. In Proceedings of the Proceedings of the Forty-First International Conference on Machine Learning (ICML). PMLR, 2024.
28. Li, Y.; Wang, Y.; Chen, Z.; Yuan, H. A Multi-Layer Aggregation Backdoor Defense Framework for Federated Learning. In Proceedings of the 2025 International Conference on Communication, Remote Sensing and Information Technology (CRSIT). IEEE, 2025, pp. 126–132.
29. Abazari, A.; Ghafouri, M.; Jafarigiv, D.; Atallah, R.; Assi, C. Developing a Security Metric for Assessing the Power Grid's Posture Against Attacks From EV Charging Ecosystem. *IEEE Transactions on Smart Grid* **2025**, *16*, 254–276. https://doi.org/10.1109/TSG.2024.3451970.
30. Presekal, A.; Ştefanov, A.; Semertzis, I.; Palensky, P. Spatio-Temporal Advanced Persistent Threat Detection and Correlation for Cyber-Physical Power Systems Using Enhanced GC-LSTM. *IEEE Transactions on Smart Grid* **2025**, *16*, 1654–1666. https://doi.org/10.1109/TSG.2024.3474039.
31. Wu, Y.; Hu, Y.; Wang, J.; Feng, M.; Dong, A.; Yang, Y. An Active Learning Framework Using Deep Q-Network for Zero-Day Attack Detection. *Computers & Security* **2024**, *139*, 103713.
32. Pan, D.; Wu, B.N.; Sun, Y.L.; Xu, Y.P. A Fault-Tolerant and Energy-Efficient Design of a Network Switch Based on a Quantum-Based Nano-Communication Technique. *Sustainable Computing: Informatics and Systems* **2023**, *37*, 100827.
33. Hammad, A.A.; Ahmed, S.R.; Abdul-Hussein, M.K.; Ahmed, M.R.; Majeed, D.A.; Algburi, S. Deep Reinforcement Learning for Adaptive Cyber Defense in Network Security. In Proceedings of the Proceedings of the Cognitive Models and Artificial Intelligence Conference (CMAI). ACM, 2024, pp. 292–297.
34. Wu, B.; Huang, J.; Duan, Q. FedTD3: An Accelerated Learning Approach for UAV Trajectory Planning. In Proceedings of the International Conference on Wireless Artificial Intelligent Computing Systems and Applications (WASA). Springer, 2025, pp. 13–24.
35. Wu, B.; Huang, J.; Duan, Q. Real-Time Intelligent Healthcare Enabled by Federated Digital Twins With AoI Optimization. *IEEE Network* **2025**, pp. 1–1. https://doi.org/10.1109/MNET.2025.3565977.
36. Paparrizos, J.; Boniol, P.; Liu, Q.; Palpanas, T. Advances in Time-Series Anomaly Detection: Algorithms, Benchmarks, and Evaluation Measures. In Proceedings of the Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD). ACM, 2025, pp. 6151–6161.
37. Feng, X.; Han, J.; Zhang, R.; Xu, S.; Xia, H. Security Defense Strategy Algorithm for Internet of Things Based on Deep Reinforcement Learning. *High-Confidence Computing* **2024**, *4*, 100167.
38. Fang, Z.; Wang, J.; Ren, Y.; Han, Z.; Poor, H.V.; Hanzo, L. Age of information in energy harvesting aided massive multiple access networks. *IEEE Journal on Selected Areas in Communications* **2022**, *40*, 1441–1456.
39. Farhaoui, Y.; Allaoui, A.E.; Amounas, F.; Mohammed, F.; Ziani, S.; Taherdoost, H.; Triantafyllou, S.A.; Bhushan, B. A Multi-Layered Protection System for Enhancing Data Security in Cloud Computing Environments. In Proceedings of the International Conference on Artificial Intelligence and Smart Environment (AISE). Springer, 2024, pp. 559–568.
40. Feng, J.; Ren, Z.; Li, C.; Li, W. A Benders-Combined Safe Reinforcement Learning Framework for Risk-Averse Dispatch Considering Frequency Security Constraints. *IEEE Transactions on Circuits and Systems II: Express Briefs* **2025**, *72*, 1063–1067. https://doi.org/10.1109/TCSII.2025.3584894.
41. Fang, Z.; Liu, Z.; Wang, J.; Hu, S.; Guo, Y.; Deng, Y.; Fang, Y. Task-Oriented Communications for Visual Navigation With Edge-Aerial Collaboration in Low Altitude Economy. *arXiv preprint arXiv:2504.18317 (arXiv)* **2025**.

 doi:10.20944/preprints202510.0141.v1

42. Huang, J.; Wu, B.; Duan, Q.; Dong, L.; Yu, S. A Fast UAV Trajectory Planning Framework in RIS-Assisted Communication Systems With Accelerated Learning via Multithreading and Federating. *IEEE Transactions on Mobile Computing* **2025**, pp. 1–16. https://doi.org/10.1109/TMC.2025.3544903.
43. Pan, D.; Wu, B.N.; Sun, Y.L.; Xu, Y.P. A fault-tolerant and energy-efficient design of a network switch based on a quantum-based nano-communication technique. *Sustainable Computing: Informatics and Systems* **2023**, *37*, 100827. https://doi.org/https://doi.org/10.1016/j.suscom.2022.100827.