Robust Peer-to-Peer Machine Learning Against Poisoning Attacks

Myria Bouhaddi and Kamel Adi
Computer Security Research Laboratory, University of Quebec in Outaouais, Gatineau, Quebec, Canada

Keywords: Peer-to-Peer Machine Learning, Poisoning Attacks, Adversarial Machine Learning, Robust Aggregation, Decentralized AI.

Abstract: Peer-to-Peer Machine Learning (P2P ML) offers a decentralized alternative to Federated Learning (FL), removing the need for a central server and enhancing scalability and privacy. However, the lack of centralized oversight exposes P2P ML to model poisoning attacks, where malicious peers inject corrupted updates. A major threat comes from adversarial coalitions, groups of peers that collaborate to reinforce poisoned updates and bypass local trust mechanisms. In this work, we investigate the impact of such coalitions and propose a defense framework that combines variance-based trust evaluation, Byzantine-inspired thresholding, and a feedback-driven self-healing mechanism. Extensive simulations in various attack scenarios demonstrate that our approach significantly improves robustness, ensuring high accuracy, detection by attackers, and model stability under adversarial conditions.

1 INTRODUCTION

Machine learning (ML) has transformed domains such as autonomous systems, medical diagnostics, financial fraud detection, and cybersecurity. These advances have mainly relied on centralized architectures, where large volumes of data are aggregated on a central server for model training. Although this facilitates optimization, it raises concerns about data privacy, security vulnerabilities, and scalability, particularly with sensitive or geographically distributed data.

Decentralized learning paradigms have emerged to address these issues. Federated Learning (FL) enables clients to collaboratively train a model without sharing raw data, but it still relies on a central server for aggregation, creating a single point of failure and a potential adversarial target.

To eliminate central coordination, *Peer-to-Peer Machine Learning* (P2P ML) offers a fully decentralized alternative. Each node maintains and trains its local model, periodically exchanging parameters with neighbors. This structure promotes scalability, preserves data locality, and suits privacy-sensitive or infrastructure-constrained environments.

In scenarios where centralized coordination is infeasible due to connectivity constraints, dynamic topologies, or lack of infrastructure, P2P ML becomes a natural fit. Unlike Federated Learning, which still depends on a central server for aggregation, P2P ML enables nodes to exchange and integrate updates locally through a fully decentralized protocol. This architecture is particularly suited for environments such as the Internet of Things (IoT), Mobile Ad Hoc Networks (MANETs), Vehicular Ad Hoc Networks (VANETs), and decentralized blockchain ecosystems.

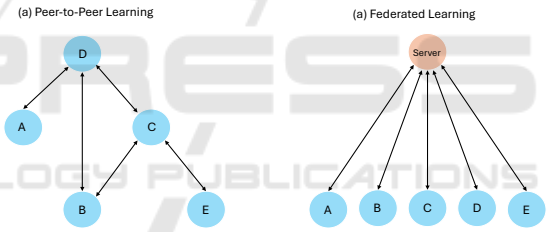

Figure 1: Comparison between Federated Learning and Peer-to-Peer Machine Learning. In FL, a central server aggregates updates from clients, while in P2P ML, peers exchange updates directly.

In these domains, devices must operate autonomously under energy constraints, intermittent connectivity, or rapid topology changes. For example, IoT nodes often lack continuous communication; VANETs require model updates on the move; MANETs form spontaneously without fixed infrastructure; and blockchain applications align naturally with decentralized learning models, especially for collaborative fraud or anomaly detection.

By adapting to these conditions, P2P ML broadens the scope of collaborative learning. However,

Bouhaddi, M. and Adi, K.
Robust Peer-to-Peer Machine Learning Against Poisoning Attacks.
DOI: 10.5220/0013640600003979
In *Proceedings of the 22nd International Conference on Security and Cryptography (SECRYPT 2025)*, pages 539-546
ISBN: 978-989-758-760-3; ISSN: 2184-7711
Copyright © 2025 by Paper published under CC license (CC BY-NC-ND 4.0)

the absence of central authority introduces vulnerabilities, notably model poisoning attacks, where malicious peers inject corrupted updates to degrade performance or implant stealthy backdoors (Bouhaddi and Adi, 2024). Unlike FL, which allows for centralized filtering, ML P2P's localized, asynchronous exchanges make detecting poisoning attacks significantly harder. Without a global reference, the nodes must rely solely on limited neighborhood views, complicating the distinction between benign and malicious behavior. The decentralized and dynamic nature of P2P networks further exacerbates this risk, allowing adversaries to exploit topology weaknesses, coordinate across multiple nodes, and propagate poisoned updates across several hops before detection.

In this work, we investigate critical vulnerabilities in peer-to-peer machine learning under poisoning attacks, focusing particularly on adversarial coalitions. Without centralized coordination, each node must independently assess the trustworthiness of its neighbors based solely on local information, creating an asymmetry that coordinated adversaries can exploit to subvert learning outcomes while avoiding detection.

To address this challenge, we propose a defense framework tailored for fully decentralized environments. It combines three complementary mechanisms: (1) a local variance-based reputation update to detect anomalies, (2) a coalition-based detection method that integrates structural, reputational and performance-based signals, and (3) a self-healing protocol that enables compromised nodes to gradually recover by adjusting their dependence on local updates. Together, these layers allow nodes to adapt dynamically, suppress poisoned information, and maintain robust model performance.

The remainder of this paper is organized as follows. Section 2 reviews related work on adversarial threats in decentralized learning; Section 3 introduces the threat model and attack strategies; Section 4 presents our defense mechanism; Section 5 provides experimental validation and analysis; and Section 6 concludes with insights and future directions.

2 STATE OF THE ART

Numerous studies have highlighted the impact of poisoning attacks in Federated Learning, where adversaries exploit collaborative updates by corrupting training data or directly manipulating model updates. Since peer-to-peer machine learning shares architectural similarities with FL but lacks a central aggregator, understanding poisoning attacks in FL provides a strong basis for analyzing threats in decentralized settings.

Poisoning attacks can be categorized by methodology and attacker intent. The two main types are data poisoning and model poisoning. In data poisoning, adversaries manipulate training data, either through clean label attacks, where adversarial samples resemble legitimate data, or dirty label attacks, where labels are flipped to mislead learning (Shejwalkar et al., 2022; Sun et al., 2022; Shafahi et al., 2018; Rong et al., 2022). Model poisoning involves directly altering updates, optimizing them to avoid detection while significantly influencing the global model (Bhagoji et al., 2019; Sun et al., 2019; Bagdasaryan et al., 2020). Attackers may also inject backdoors, hidden triggers causing targeted misclassifications (Xie et al., 2019; Zhou et al., 2021).

Another key dimension is the target of the attacker. Targeted poisoning alters predictions for specific inputs while maintaining overall accuracy, often through backdoors (Sun et al., 2022; Tolpegin et al., 2020). Semi-target attacks degrade the performance of certain classes, and untargeted attacks disrupt the overall convergence of the model (Cao and Gong, 2022). Centralized aggregation in FL can mitigate some effects, but in P2P ML, adversarial coalitions reinforce malicious updates, making semi-targeted and untargeted attacks especially dangerous.

Despite extensive research, many defenses remain insufficient, particularly in decentralized environments. Byzantine-resilient aggregation (BRA) filters extreme gradients but often assumes a majority of honest clients, an assumption invalidated under collusion (Sun et al., 2019; Panda et al., 2022). Privacy-preserving techniques such as Secure Aggregation (SA) and differential privacy protect confidentiality but do not prevent adversarial updates, and can even be exploited via structured noise injection (Sun et al., 2019; Hossain et al., 2021; Naseri et al., 2020). Moreover, most defenses assume static attack strategies, whereas adaptive adversaries that use reinforcement learning can continuously evade detection (Li et al., 2022).

The transition from FL to P2P ML introduces new challenges. In FL, centralized servers enforce trust, but in P2P ML, trust must be decentralized, exposing the system to Sybil attacks and collusion. Blockchain-based trust models, leveraging cryptographic proofs, offer a promising solution. In addition, topology-aware aggregation can detect poisoned updates by detecting anomalies in local neighborhoods. Given that adversarial influence spreads faster in P2P ML, defenses must integrate metalearning and autoencoder-based anomaly detection to identify subtle deviations without accessing raw data.

In sum, poisoning attacks represent a critical challenge for both FL and P2P ML. Although FL defenses are well studied, their effectiveness in fully decentralized settings remains uncertain. P2P ML demands trust-based, distributed, and topology-aware defenses capable of mitigating adversarial influence while preserving collaborative learning efficiency. Future research should focus on distributed anomaly detection, adaptive poisoning mitigation, and resilient peer-to-peer aggregation to secure decentralized learning systems.

3 THREAT MODEL

3.1 System Model

We consider a decentralized peer-to-peer machine learning framework where nodes collaboratively train a model without a central server. The system is represented as a graph $G = (V,E)$, with V denoting the set of nodes and $E \subseteq V \times V$ the bidirectional communication links. Each node $i \in V$ maintains a local model w_i and periodically exchanges parameters with its neighbors $\mathcal{N}_i$, aggregating updates based on trust, similarity, or statistical heuristics.

Without a global coordinator, the attack surface increases significantly. Unlike federated learning, where a central server can apply robust aggregation or anomaly detection, P2P ML systems rely entirely on peer interactions, making them more vulnerable to adversarial interference, especially under coordinated attacks.

3.2 Adversarial Capabilities

We assume a coalition of malicious nodes $V_A \subset V$ that aims to disrupt the learning process by injecting corrupted updates during aggregation. These nodes operate independently and have full control over their local training data and model updates. Before broadcasting, they can arbitrarily alter gradients or model weights, producing noisy, biased, or adversarial updates.

The proportion of adversarial nodes is denoted by $\alpha = \frac{|V_A|}{|V|}$, where $0 < \alpha < 1$. Malicious nodes participate in each training round, exchange messages with honest neighbors, and use various poisoning strategies, ranging from simple label flipping to sophisticated gradient manipulations designed to evade naive detection mechanisms.

3.3 Attack Objectives and Strategies

The objectives of adversarial nodes can be broadly classified into three categories. The first is global model degradation, where inconsistent or high-variance updates disrupt convergence, slowing training, or leading to unstable, poorly generalized models. The second is targeted model manipulation, where adversaries embed specific misclassifications or backdoors, subtly altering decision boundaries while maintaining overall accuracy to evade detection.

The third and more sophisticated strategy is coalition-based trust subversion. Here, malicious nodes coordinate to reinforce each other's poisoned updates, exploiting trust mechanisms based on similarity or consistency. This coordination gradually increases their influence, steering the learning process toward adversarial objectives while avoiding detection through mutual support.

3.4 Formalization of the Threat

Formally, each node $i \in V$ aggregates neighbor models using a local aggregation rule $\mathcal{A}_i$. An adversarial node $j \in V_A$ aims to produce an update w_j^A such that the aggregated model $\mathcal{A}_i(\{w_k\}_{k\in\mathcal{N}_i})$ deviates maximally from the expected global update w^*, while remaining stealthy to evade local detection. This defines a dual objective: to maximize impact while minimizing detectability.

The effectiveness of such attacks depends on factors such as the number and distribution of malicious nodes, the topology and dynamics of the communication graph, and the aggregation strategies used by the honest nodes. In highly connected graphs, malicious influence may be diluted, whereas in sparse or structured networks, even small coalitions can exert significant impact, especially by exploiting trust mechanisms or statistical shortcuts.

3.5 Adversarial Scenarios

To capture the range of adversarial behaviors and evaluate the robustness of our defense mechanisms, we define three representative attack scenarios.

Scenario 1 – Single Adversarial Neighbor. The node is surrounded by mostly honest neighbors, with only one adversarial peer injecting poisoned updates. This scenario evaluates local statistical techniques, such as variance-based reputation mechanisms, to identify and isolate outliers.

Scenario 2 – Critical Mass of Malicious Neighbors. As the number of adversarial neighbors in-

creases, their influence on aggregation increases. This scenario explores the threshold beyond which local defenses like statistical filtering or reputation adjustments fail, analogous to the Byzantine fault tolerance threshold.

Scenario 3 – Fully Poisoned Neighborhood. In this extreme case, all neighbors are malicious. The node, deprived of any honest reference, must rely on a feedback-based mechanism, using future or alternative neighbors to retroactively detect poisoning and engage in self-healing behavior.

These scenarios cover the spectrum of adversarial influence in decentralized learning environments, from isolated attacks to complete compromise. The next section introduces our defense mechanisms and shows how they address these scenarios under varying conditions.

4 DEFENSE MODEL AGAINST POISONING ATTACKS IN P2P MACHINE LEARNING

We introduce a defense model to counter poisoning attacks in peer-to-peer machine learning systems. Our approach combines a dynamically evolving trust mechanism based on update consistency, a reputation-aware aggregation strategy, and a feedback-driven self-correction mechanism. These components address adversarial configurations that range from isolated attackers to fully compromised neighborhoods.

4.1 System and Trust Graph Formalization

We consider a decentralized peer-to-peer learning system composed of n nodes. Each node maintains and updates a local model through iterative training and exchanges with its direct neighbors. The structure of communication and trust is represented by a directed graph $G = (V, E, R)$, where $V = \{1, \ldots, n\}$ denotes the set of nodes, $E \subseteq V \times V$ the directed communication links and $R = \{r_{ij}^{(t)} \in [0,1]\}$ the dynamic reputation scores at each round t. A directed edge $(j,i) \in E$ indicates that node i receives an update from node j, with $r_{ij}^{(t)}$ reflecting the trust placed by i in j in round t.

Each round t, node i aggregates the model parameters received from neighbors $\mathcal{N}(i)$ together with its own model $w_i^{(t)}$ using:

$$w_i^{(t+1)} = a_{ii}^{(t)} \cdot w_i^{(t)} + \sum_{j \in \mathcal{N}(i)} a_{ij}^{(t)} \cdot \tilde{w}_j^{(t)},$$

where $\tilde{w}_j^{(t)}$ is the update of node j, and the weights satisfy:

$$a_{ii}^{(t)} + \sum_{j \in \mathcal{N}(i)} a_{ij}^{(t)} = 1.$$

The aggregation weights are computed as:

$$a_{ij}^{(t)} = \frac{r_{ij}^{(t)}}{r_{ii}^{(t)} + \sum_{k \in \mathcal{N}(i)} r_{ik}^{(t)}}, \quad \text{for } j \in \mathcal{N}(i) \cup \{i\}.$$

This mechanism ensures that the most trusted nodes have greater influence, while the nodes with lower reputation scores are reduced, allowing each node to adaptively balance external inputs against its own updates in adversarial settings.

4.2 Variance-Based Reputation Update Mechanism

The trust-based aggregation framework uses reputation scores $r_{ij}^{(t)}$ to weigh the influence of neighbor j on node i. These scores are updated at each round based on the consistency of received updates, with the aim of down-weighting neighbors whose updates deviate significantly from expected behavior.

In each round t, node i receives updates $\{\tilde{w}_j^{(t)}\}_{j \in \mathcal{N}(i)}$ and computes a local consensus model $\bar{w}_i^{(t)}$, for example, using the coordinate median. The deviation of each neighbor j is quantified by:

$$d_{ij}^{(t)} = \left\| \tilde{w}_j^{(t)} - \bar{w}_i^{(t)} \right\|_2.$$

The node i then calculates the empirical variance $\mathrm{Var}_i^{(t)}$ in all deviations. A soft trust score $s_{ij}^{(t)} \in (0,1]$ is assigned by:

$$s_{ij}^{(t)} = \exp\left(- \frac{d_{ij}^{(t)}}{\sqrt{\mathrm{Var}_i^{(t)}} + \varepsilon} \right),$$

where $\varepsilon > 0$ prevents division by zero.

The reputation score is updated using exponential smoothing:

$$r_{ij}^{(t+1)} = \lambda \cdot r_{ij}^{(t)} + (1 - \lambda) \cdot s_{ij}^{(t)},$$

where $\lambda \in [0,1]$ controls the stability-speed trade-off: small λ adapts quickly, large λ emphasizes long-term behavior.

By reinforcing consistent behavior and penalizing outliers, this mechanism dynamically adjusts trust to maintain robustness against adversarial updates.

4.3 Detection of Coalition Attacks via Local Byzantine-Aware Thresholding

The variance-based reputation mechanism effectively detects isolated adversarial behaviors, where a single poisoned update deviates significantly from others. However, it fails when multiple malicious neighbors collude by sending similarly crafted updates. In such cases, the statistical deviation becomes artificially low, making malicious nodes appear trustworthy and allowing them to gain influence while evading local anomaly detection.

To address this, each node i monitors not only the variance of incoming updates but also the wider consistency of its own learning dynamics. We propose a local detection strategy based on three jointly satisfied conditions.

First, node i tracks the variance $\mathrm{Var}_i^{(t)}$ among the received updates. A very low variance may indicate either stability or artificial agreement among adversaries. Second, node i evaluates the proportion of trusted neighbors:

$$\mathcal{H}_i^{(t)} = \left\{ j \in \mathcal{N}(i) \mid r_{ij}^{(t)} \geq \delta \right\},$$

with high reputation ratio:

$$\eta_i^{(t)} = \frac{|\mathcal{H}_i^{(t)}|}{|\mathcal{N}(i)|}.$$

A large $\eta_i^{(t)}$ usually reflects neighborhood trust, but, when combined with low variance, can suggest collusion.

Third, node i monitors its own local loss $\mathcal{L}_i^{(t)}$. If trusted neighbors provide consistent updates but $\mathcal{L}_i^{(t)}$ remains high, this indicates adversarial influence.

Node i indicates a possible coalition poisoning attack if:

$$\mathrm{Var}_i^{(t)} < \varepsilon_v, \quad \eta_i^{(t)} > \theta, \quad \mathcal{L}_i^{(t)} > \mathcal{L}_{\max},$$

where ε_v, θ, and $\mathcal{L}_{\max}$ are system-defined thresholds.

This tri-criteria mechanism provides a robust local indicator of coalition-based poisoning by integrating structural, reputational, and learning-performance signals. Upon detection, node i initiates a protective strategy described in the next section.

Upon diagnosing a possible coalition poisoning attack: low variance, high neighbor trust, and degraded local performance, node i initiates a self-healing protocol to mitigate malicious influence.

The strategy temporarily isolates the node from poisoned updates by prioritizing its own model. Let $\gamma \in (0,1)$ denote the self-reliance factor; the updated aggregation rule is:

$$w_i^{(t+1)} = \gamma \cdot w_i^{(t)} + (1-\gamma) \cdot \sum_{j \in \mathcal{N}(i)} a_{ij}^{(t)} \cdot \tilde{w}_j^{(t)},$$

with γ close to 1 during healing. This reduces external influence while preserving the learning structure.

The self-healing phase lasts a fixed number of rounds Δ, during which node i monitors its local loss $\mathcal{L}_i^{(t)}$. If $\mathcal{L}_i^{(t)} < \mathcal{L}_{\max}$, the node gradually reintroduces external updates by decreasing γ; otherwise, it prolongs the healing phase, relying primarily on local data.

This mechanism provides a local, reactive defense without requiring global coordination, thus maintaining full decentralization.

4.4 Algorithmic Overview and Temporal Adaptation Strategy

We summarize our defense strategy as a dynamic algorithm operating on a finite learning horizon of T communication rounds. In each round, every node i receives updates from neighboring nodes and performs local computations: reputation updates, aggregation, anomaly detection, and, if needed, self-healing.

The model adapts over time through three interconnected phases: (1) trust-based aggregation informed by variance-aware reputation updates, (2) coalition attack detection based on structural, reputational, and performance signals, and (3) a feedback-driven self-healing mechanism that rebalances aggregation toward local updates during compromise.

This layered strategy enables each node to refine neighbor trustworthiness, detect coordinated poisoning without central oversight, and react autonomously to adversarial conditions.

The algorithm 1 details the entire procedure at each node, highlighting the modularity and adaptability of the defense process to various levels of threat and network configurations.

5 EXPERIMENTAL EVALUATION

We evaluate the effectiveness of the proposed defense model in protecting a target node against poisoning attacks in a decentralized peer-to-peer machine learning environment. The experiments are carried out under four adversarial scenarios, each representing a specific neighborhood configuration. We evaluate performance in terms of local learning accuracy, loss, and attacker detection rate.

Algorithm 1: Defense Against Poisoning Attacks in P2P Machine Learning.

Require: Trust graph $G = (V, E, R)$, training rounds T, smoothing factor λ, variance threshold ε_v, reputation threshold δ, trust density threshold θ, loss threshold $\mathcal{L}_{\max}$, self-healing duration Δ, self-reliance factor γ

Initialize local model $w_i^{(0)}$ and reputations $r_{ij}^{(0)} = 1$ for all $j \in \mathcal{N}(i)$
for each round $t = 1$ to T **do**
  for each node $i \in V$ **do**
    Receive updates $\{\tilde{w}_j^{(t)}\}_{j \in \mathcal{N}(i)}$
    Compute local consensus $\bar{w}_i^{(t)} \leftarrow \text{median}(\{\tilde{w}_j^{(t)}\})$
    for each neighbor $j \in \mathcal{N}(i)$ **do**
      Compute deviation $d_{ij}^{(t)} = \|\tilde{w}_j^{(t)} - \bar{w}_i^{(t)}\|_2$
      Compute trust score: $s_{ij}^{(t)} = \exp\left(-\frac{d_{ij}^{(t)}}{\sqrt{\text{Var}_i^{(t)} + \varepsilon}}\right)$
      Update reputation: $r_{ij}^{(t+1)} = \lambda \cdot r_{ij}^{(t)} + (1-\lambda) \cdot s_{ij}^{(t)}$
    end for
    Compute normalized weights: $a_{ij}^{(t)} = \frac{r_{ij}^{(t)}}{r_{ii}^{(t)} + \sum_{k \in \mathcal{N}(i)} r_{ik}^{(t)}}$
    Compute trust density ratio: $\eta_i^{(t)} = \frac{|\{j \in \mathcal{N}(i) | r_{ij}^{(t)} \geq \delta\}|}{|\mathcal{N}(i)|}$
    if $\text{Var}_i^{(t)} < \varepsilon_v$ and $\eta_i^{(t)} > \theta$ and $\mathcal{L}_i^{(t)} > \mathcal{L}_{\max}$ **then**
      for each healing step $\tau = 1$ to Δ **do**
        Self-healing aggregation: $w_i^{(t+1)} = \gamma \cdot w_i^{(t)} + (1-\gamma) \cdot \sum_{j \in \mathcal{N}(i)} a_{ij}^{(t)} \cdot \tilde{w}_j^{(t)}$
      end for
    else
      Normal aggregation: $w_i^{(t+1)} = a_{ii}^{(t)} \cdot w_i^{(t)} + \sum_{j \in \mathcal{N}(i)} a_{ij}^{(t)} \cdot \tilde{w}_j^{(t)}$
    end if
  end for
end for

5.1 Simulation Setup

The simulation involves 50 nodes structured as a random directed graph. Each node trains a local model on a private, non-i.i.d. subset of the MNIST dataset and exchanges updates with its direct neighbors. In each experiment, we focus on a specific target node and vary the nature of its neighbors to simulate different adversarial settings. The target learning dynamics is monitored over 200 communication rounds.

The parameters used in the simulations are summarized in Table 1.

Table 1: Simulation Parameters.

Parameter	Value
Number of nodes (n)	50
Graph topology	Random directed graph (avg. degree = 4)
Learning rounds (T)	200
Local model	2-layer MLP (ReLU, softmax)
Local data per node	1,200 MNIST samples
Optimizer	SGD (learning rate = 0.01)
Batch size	32
Reputation smoothing factor (λ)	0.7
Variance threshold (ε_v)	10^{-3}
Reputation threshold (δ)	0.8
Trust density threshold (θ)	0.6
Loss threshold ($\mathcal{L}_{\max}$)	0.5
Self-reliance factor (γ)	0.95
Self-healing duration (Δ)	10 rounds

5.2 Defense Strategies Compared

We compare four approaches:

- **No Defense:** neighbor updates are aggregated without filtering or weighting.
- **Static Weighting:** aggregation with fixed trust weights.
- **Variance Only:** reputation scores updated based on deviation from consensus.
- **Full Defense:** complete model including variance scoring, Byzantine-aware filtering, and self-healing.

5.3 Scenario-Based Evaluation

Each scenario simulates a different adversarial configuration around the target node:

- **Scenario A (Isolated Attack):** one neighbor sends poisoned updates.
- **Scenario B (Coalition - 40%):** 40% of neighbors coordinate poisoned updates.
- **Scenario C (Full Compromise):** all neighbors are malicious and colluding.
- **Scenario D (Dynamic Adversaries):** attackers appear and disappear during training.

5.4 Accuracy of the Target Node

Figure 2 shows the average classification accuracy of the target node in all scenarios. In Scenario A, even the *Variance Only* defense performs well, as

Figure 2: Target node accuracy across scenarios and defense strategies.

the isolated attack is easily detectable. However, in Scenarios B and C, involving coordinated poisoning, variance-based defense becomes less effective, failing to detect consistent malicious updates. In contrast, the *Full Defense* mechanism—combining statistical filtering, trust thresholding, and self-healing - maintains high precision and effectively mitigates coalition attacks. In Scenario D, with dynamic adversaries, the full strategy again proves adaptive, supporting robust learning under evolving attack patterns.

5.5 Local Loss of the Target Node

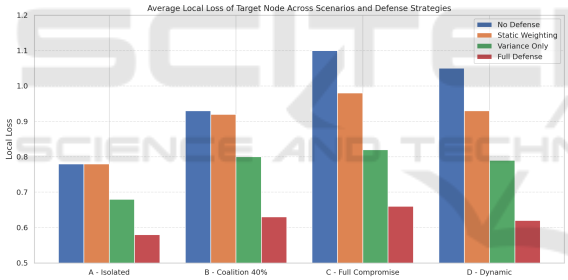

Figure 3: Local loss observed by the target node across scenarios.

Figure 3 reports the average local loss observed by the target node in different attack scenarios and defense strategies. The loss reflects the model's ability to fit its local data despite adversarial interference.

In Scenario A, all strategies limit the loss, and variance-based defense already provides noticeable improvement over baseline, confirming that isolated attackers can be effectively down-weighted by deviation detection.

In Scenarios B and C, adversarial coordination becomes more evident: coherent poisoned updates reduce variance, weakening the variance-only defense. Consequently, local loss remains elevated even with statistical filtering, and static weighting also fails to mitigate the poisoning effect.

In contrast, the full defense consistently achieves the lowest loss in all scenarios, highlighting the benefits of combining reputation filtering, threshold-based rejection, and self-healing to preserve model integrity.

Scenario D further emphasizes the need for adaptivity: as attack patterns evolve, the full defense dynamically adjusts, maintaining bounded local loss. Overall, these results confirm that our model not only prevents convergence failures but also maintains robust learning under realistic adversarial conditions.

5.6 Malicious Node Detection Rate

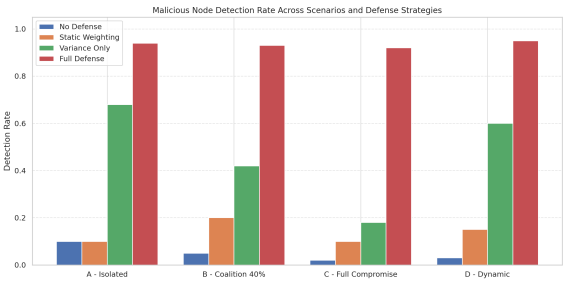

Figure 4: Detection rate of malicious neighbors by the target node.

Figure 4 presents the detection rate of malicious neighbors by the target node across scenarios. This metric measures the ability to correctly identify poisoned updates.

In Scenario A (isolated attacker), the *Variance Only* defense achieves a detection rate above 65%, effectively capturing outlier updates. However, in Scenario B (coalition of 40%), the detection rate drops to 42% as colluding attackers reduce the variance, making malicious behavior statistically indistinguishable from honest peers.

The situation worsens in Scenario C (full compromise), where variance-based detection almost fails, with rates as low as 18%. This highlights the limits of relying solely on variance signals.

In contrast, the *full defense mechanism* consistently exceeds 90% detection in all scenarios, including dynamic attacks (Scenario D). This robustness results from combining variance monitoring, reputation filtering, and self-healing, enabling the system to identify poisoning sources even under subtle adversarial conditions.

Overall, these results confirm that single-layer anomaly detection is insufficient in adversarial peer-to-peer environments. A layered and adaptive approach, which integrates multiple signals over time, is essential for robust defense.

6 CONCLUSION

We addressed poisoning attacks in peer-to-peer machine learning, where nodes aggregate updates without central authority. Although scalable and privacy-friendly, this architecture complicates the detection of malicious behaviors, especially in the presence of colluding adversaries.

We proposed a defense framework that combines variance-based reputation scoring, Byzantine-aware thresholding, and feedback-driven self-healing, enabling nodes to detect and mitigate both isolated and coordinated attacks.

Experiments show that variance alone is insufficient against collusions, whereas our full defense preserves model accuracy, reduces loss, and maintains high detection rates under dynamic adversarial conditions.

Future work will explore context-sensitive dynamic trust thresholds to further enhance the adaptability and resilience of decentralized learning systems.

REFERENCES

Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., and Shmatikov, V. (2020). How to backdoor federated learning. In *International conference on artificial intelligence and statistics*, pages 2938–2948. PMLR.

Bhagoji, A. N., Chakraborty, S., Mittal, P., and Calo, S. (2019). Analyzing federated learning through an adversarial lens. In *International conference on machine learning*, pages 634–643. PMLR.

Bouhaddi, M. and Adi, K. (2024). When rewards deceive: Counteracting reward poisoning on online deep reinforcement learning. In *2024 IEEE International Conference on Cyber Security and Resilience (CSR)*, pages 38–44. IEEE.

Cao, X. and Gong, N. Z. (2022). Mpaf: Model poisoning attacks to federated learning based on fake clients. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3396–3404.

Hossain, M. T., Islam, S., Badsha, S., and Shen, H. (2021). Desmp: Differential privacy-exploited stealthy model poisoning attacks in federated learning. In *2021 17th International Conference on Mobility, Sensing and Networking (MSN)*, pages 167–174. IEEE.

Li, H., Sun, X., and Zheng, Z. (2022). Learning to attack federated learning: A model-based reinforcement learning attack framework. *Advances in Neural Information Processing Systems*, 35:35007–35020.

Naseri, M., Hayes, J., and De Cristofaro, E. (2020). Local and central differential privacy for robustness and privacy in federated learning. *arXiv preprint arXiv:2009.03561*.

Panda, A., Mahloujifar, S., Bhagoji, A. N., Chakraborty, S., and Mittal, P. (2022). Sparsefed: Mitigating model poisoning attacks in federated learning with sparsification. In *International Conference on Artificial Intelligence and Statistics*, pages 7587–7624. PMLR.

Rong, D., Ye, S., Zhao, R., Yuen, H. N., Chen, J., and He, Q. (2022). Fedrecattack: Model poisoning attack to federated recommendation. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 2643–2655. IEEE.

Shafahi, A., Huang, W. R., Najibi, M., Suciu, O., Studer, C., Dumitras, T., and Goldstein, T. (2018). Poison frogs! targeted clean-label poisoning attacks on neural networks. *Advances in neural information processing systems*, 31.

Shejwalkar, V., Houmansadr, A., Kairouz, P., and Ramage, D. (2022). Back to the drawing board: A critical evaluation of poisoning attacks on production federated learning. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1354–1371. IEEE.

Sun, Y., Ochiai, H., and Sakuma, J. (2022). Semi-targeted model poisoning attack on federated learning via backward error analysis. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Sun, Z., Kairouz, P., Suresh, A. T., and McMahan, H. B. (2019). Can you really backdoor federated learning? *arXiv preprint arXiv:1911.07963*.

Tolpegin, V., Truex, S., Gursoy, M. E., and Liu, L. (2020). Data poisoning attacks against federated learning systems. In *Computer security–ESORICs 2020: 25th European symposium on research in computer security, ESORICs 2020, guildford, UK, September 14–18, 2020, proceedings, part i 25*, pages 480–501. Springer.

Xie, C., Huang, K., Chen, P.-Y., and Li, B. (2019). Dba: Distributed backdoor attacks against federated learning. In *International conference on learning representations*.

Zhou, X., Xu, M., Wu, Y., and Zheng, N. (2021). Deep model poisoning attack on federated learning. *Future Internet*, 13(3):73.