# Exploiting Gaussian Noise Variance for Dynamic Differential Poisoning in Federated Learning

Md Tamjid Hossain, Shahriar Badsha, Hung La, Shafkat Islam, and Ibrahim Khalil

***Abstract*—The emerging field of Federated Learning (FL) is reshaping privacy-preserved data analysis and decision support mechanisms within several critical infrastructure (CIs) sectors such as autonomous transportation, energy, and healthcare. To shield sensitive operational and client data from privacy attackers, Differential Privacy (DP) has been proposed to integrate on top of the FL process. Yet, we identify that integrating Gaussian noise for achieving DP guarantee can inadvertently create a new vector for differential model poisoning attacks in FL. Moreover, exploiting the variance in Gaussian noise enables attackers to camouflage their activities within the legitimate noise of the system, a significant yet largely overlooked security flaw in the differentially private federated learning (DPFL) framework. Addressing this research gap, we introduce a novel adaptive model poisoning through episodic loss memorization ($\alpha$−MPELM) technique. This method enables attackers to dynamically inject adversarial noise into the differentially private local model parameters. The technique has a dual purpose: hindering the optimal convergence of the global FL model and simultaneously avoiding detection by the anomaly detectors. Our evaluation of the $\alpha$−MPELM attack reveals its capability to deceive Norm, Accuracy, and Mix anomaly detection algorithms, surpassing the conventional random malicious device (RMD) attacks with attack accuracy improvements of $6.8\%$, $12.6\%$, and $13.8\%$, respectively. Additionally, we introduce a reinforcement learning-based DP level selection strategy, rDP, as an effective countermeasure against $\alpha$−MPELM attack. Our empirical findings confirm that this defense mechanism steadily progresses to an optimal policy.**

***Impact Statement*—The need for trustworthy AI/ML applications at the edge is now more critical than ever, necessitating secure and privacy-conscious systems. Federated Learning (FL) emerges as a beacon in this context, offering a decentralized approach to AI/ML that enhances data privacy while leveraging the collective intelligence of diverse datasets. However, this study unveils a critical vulnerability in the differentially private federated learning process, regarded as a promising learning technique in next-generation critical infrastructures. It reveals how differential privacy (DP) mechanisms while enhancing privacy, inadvertently open doors to stealthy model poisoning attacks. By devising a novel adaptive model poisoning technique, we demonstrate how attackers can exploit DP noise to evade advanced anomaly detection and hinder FL model convergence. To counteract this, we also propose a reinforcement learning-assisted privacy level selection strategy. This research not only exposes a significant security vulnerability in edge computing but also charts a path for strengthening AI/ML defenses.**

***Index Terms*—Edge Computing, Differential Privacy, Federated Learning, Reinforcement Learning, Artificial Intelligence, Model Poisoning, and Anomaly Detection.**

## I. Introduction

In the fast-evolving landscape of artificial intelligence (AI)-enabled edge computing, federated learning (FL) [1] has emerged as a game-changer in protecting critical infrastructures (CIs) [2] such as transportation, energy, and healthcare industries. FL's rise to prominence is largely attributed to its inherent ability to safeguard sensitive mission-critical data, thus facilitating privacy-preserved learning and decision-making within these vital sectors [3]. Distinguished from traditional centralized machine learning (ML) methods, FL enables the training of a global model directly at the network's edge. Each edge node shares only its trained model parameters (e.g., weights and biases), instead of transmitting sensitive raw data to a central server. This approach ensures that the training data remains securely within the original edge node, thereby significantly reducing the risks of exposing sensitive data. Numerous research efforts have been made to integrate FL and its variants into next-generation AI/ML-driven CIs [4], [5], [6], [7], which have increasingly become targets for adversarial attacks, evidenced by the Stuxnet attack [8] and the Dragonfly alert [ICS-ALERT-14-176-02A]. For example, the energy industry, crucial to the functionality of other CIs, can benefit from FL in applications such as energy consumption forecasting, state estimation, and generator synchronization.

However, despite FL's ability to minimize the exposure of sensitive operational data, vulnerabilities still exist. A prominent threat is the man-in-the-middle (MITM) attack, where an adversary could potentially intercept and extract valuable information from the in-transit model parameters [9]. To mitigate this, a significant body of research including data-driven privacy-preservation methods [10], [11], [12], [13] has been carried out lately. Particularly, differential privacy (DP) [14]– a standard privacy specification– has been proposed in the literature to safeguard FL's training, testing, and parameter-sharing processes. DP utilizes a randomized noise-adding mechanism following well-known statistical distributions to keep individual contributions indistinguishable. In particular, the DP-noise allows for aggregate data analysis where patterns and insights can still be accurately extracted without exposing sensitive information [14]. This balance between privacy and

This work was partially funded by the U.S. National Science Foundation (NSF) under grants: NSF-CAREER: 1846513, and NSF-PFI-TT: 1919127. The views, opinions, findings, and conclusions reflected in this publication are solely those of the authors and do not represent the official policy or position of the NSF.

Md Tamjid Hossain is with the Department of Computational, Engineering, and Mathematical Sciences at the Texas A&M University-San Antonio, USA.

Shahriar Badsha is with the General Motors, USA.

Hung La is with the Department of Computer Science and Engineering at the University of Nevada, Reno, USA.

Shafkat Islam is with the Department of Computer Science at Purdue University, USA.

Ibrahim Khalil is with the School of Computing Technologies, RMIT University, Australia.

data utility is vital in contexts where data-driven decisions are made, such as in CI applications. For example, in the energy sector, DP can enable the analysis of power consumption data to optimize energy distribution and manage demand more effectively while ensuring the privacy of individual users [15], [16]. Due to its provable privacy guarantee and low computational cost, DP is proposed to be integrated into various FL-based applications across CIs [17], [18], [19], [7].

Yet, our adversarial analysis in this paper indicates a striking vulnerability of differentially private FL (henceforth referred to as DPFL) methods: the very differential noise, added to achieve data privacy guarantee, can be exploited to conduct poisoning attacks in FL. In particular, an intelligent attacker can exploit the inherent characteristics of DP to orchestrate model poisoning attacks, a threat not widely recognized in current DPFL implementations. Such poisoning attacks can put any system in an unsafe operating condition and cause severe hazards in CIs. For instance, poisoned information can mislead an autonomous vehicle to take unsafe maneuvers (e.g., sharp turns, sudden lane changes, etc.).

The significance of addressing this identified research gap lies in its potential to improve the resilience of DPFL systems against sophisticated adversarial attacks. Understanding the security implications of integrating DP with FL allows us to develop advanced countermeasures that are specifically designed to detect and mitigate these novel vulnerabilities. This, in turn, informs the creation of more robust privacy-preserving mechanisms that can effectively balance the dual objectives of maintaining data privacy and ensuring model integrity. By closing this research gap, we can enhance the overall security posture of DPFL systems, making them more trustworthy and reliable for deployment in CI sectors.

### A. Motivations

The motivation behind our research primarily comes from a largely unexplored, yet increasingly important area: *maintaining verifiable security at the edge*. This includes the potential exploitation of DP in compromising the security and integrity of systems. While numerous studies have highlighted how DP can protect data privacy [14], [17], [18], [19], [20], [7], [6], [21], [22], [23], a limited effort has been made on how it can be exploited as a tool for conducting covert security and integrity attacks. More specifically, only a few recent research on the privacy and security challenges of CIs [24], [15], [16], [25] point out the exploitation opportunity of DP.

Integrating Gaussian noise to achieve DP in FL systems can introduce a new attack window, which threat actors can exploit for poisoning attacks. Specifically, an attacker can disguise malicious activity as legitimate DP noise by subtly adjusting the DP noise parameters, allowing harmful noise to be injected into compromised model parameters without triggering anomaly detection systems. Since exploiting this vulnerability can have catastrophic consequences for DPFL-based CIs, the development of more robust privacy-preserving mechanisms is crucial. These vulnerabilities underscore the necessity for mechanisms capable of detecting and counteracting adaptive adversarial strategies. Our research explores how such adaptive poisoning can occur.

Fig. 1 illustrates the attack vectors in the context of a DPFL process. We differentiate between two types of attacks: data poisoning, where adversarial noise is introduced into the training data, and the more complex yet potent model poisoning, where the noise is integrated into the model parameters themselves. Our primary focus is on model poisoning, given its heightened potential for disruption and the subtlety required for its execution. The specifics of these adversarial strategies are further elaborated in section IV-A of our paper.

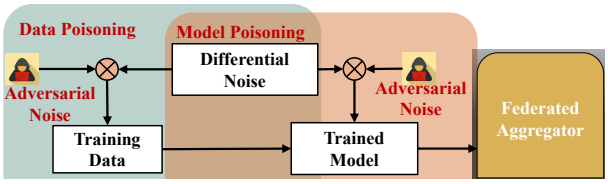


Fig. 1. DP-exploited data and model poisoning attacks in FL.

### B. Research Gap

While several model poisoning attacks and defense models have been proposed in the literature in this direction, several limitations are observed. For instance, Byzantine-robust aggregation [26], along with algorithms like *Krum* [27], *Bulyan* [28], *Trimmed Mean* [29], *Median* [29], have been developed to counteract poisoning attacks that exploit the vulnerabilities in FL aggregation rules. However, these methods often overlook two critical aspects: (1) the subtlety of maintaining attack stealthiness and (2) the specific exploitation of DP noise.

In a notable advancement, a novel DP-exploited false data injection attack is introduced in [15], which successfully evades standard anomaly detection classifiers. This study also proposes a log-likelihood ratio test-based anomaly detector as a countermeasure against such DP-exploited attacks. Yet, the performance of these DP-exploited attacks in federated settings remains unexplored, and the strategies used by attackers to adjust the intensity of poisoning are not adequately addressed.

A separate study [25] has explored DP exploitation in FL, formulating a global DP (GDP) exploited stealthy model poisoning attack. However, it leaves open the question of how such attacks achieve persistence in a Local DPFL (henceforth referred to as $\mathcal{L}$-DPFL) environment. We seek to fill this gap by analyzing these research problems and employing comparative analysis and empirical evidence to enhance the understanding of DP exploitation in the FL context.

### C. Our Work

We perform the first systematic study on the exploitation of the differential Gaussian noise to craft stealthy local model poisoning attacks within the DPFL framework. In particular, *we devise a persistent and stealthy model poisoning attack that exploits the local DP (LDP) technique to evade the state-of-the-art anomaly detectors, while simultaneously degrading the FL model utility*. The core contributions are as follows:

- We perform the first in-depth analysis on exploiting differential Gaussian noise to conduct stealthy local model poisoning attacks in FL-driven CIs.

TABLE I
LIST OF MAJOR SYMBOLS AND THEIR DESCRIPTION USED IN THIS PAPER

| Symbols | Description | Symbols | Description | Symbols | Description | Symbols | Description |
|---|---|---|---|---|---|---|---|
| $f_a$ | Adversarial distribution | $\mathcal{N}_a$ | Adversarial noise profile | $\eta_a$ | Adversarial noise | $\mu_a$ | Attack impact |
| $A$ | Action space | $R$ | Average global reward | $\beta_2$ | Accuracy benchmark | $f_0$ | Benign Gauss. distribution |
| $\eta_b$ | Benign noise | $\psi$ | Balancing param. | $\mathcal{C}$ | Clipping threshold | $\xi$ | Clipping technique |
| $\mathcal{R}$ | Detection range | $\tau$ | Detection threshold | $\upsilon$ | Deviation of model updates | $\gamma$ | Degree of poisoning |
| $\zeta$ | Discount factor | $k$ | Edge node | $\Delta w_e$ | Expected update | $\Delta w_g$ | Global update |
| $\eta$ | Gaussian DP-noise | $\rho$ | Factor of proportionality | $f_l$ | Historical federated loss | $\alpha$ | Learning rate |
| $\mathcal{D}$ | Local training data | $k_1$ | Lagrange multiplier | $\mathcal{V}$ | Local validation set | $\ell$ | Loss function |
| $\mathcal{R}$ | Loss ratio | $\Delta w$ | Model update | $\jmath$ | Mini batch of local data | $\theta$ | Mean |
| $\hat{\eta}$ | Max DP-noise | $\tau^{'}$ | Modified detection threshold | $\mathcal{W}$ | Max $\ell_2$-norm | $\beta_1$ | Norm detection benchmark |
| $b$ | No. of benign models | $a$ | No of malicious models | $\Phi$ | Objective function | $\delta$ | Probability of privacy leakage |
| $\Pi$ | Privacy accountant | $\varepsilon$ | Privacy loss | $Q$ | Q-table | $\Delta Q^*$ | Q-table converged |
| $t$ | episode | $\Delta w_r$ | Received update | $r$ | Reward | $\beta$ | Reward function |
| $N$ | Set of total nodes | $\mathcal{S}$ | Sensitivity | $\Delta w_m$ | Set of malicious updates | $M$ | Set of compromised nodes |
| $B$ | Set of benign nodes | $\xi$ | Set of noisy clipped updates | $S$ | State space | $m_l$ | Set of attacker's loss |
| $s$ | State | $d$ | Square of $L_2$ distance | $\mathcal{K}$ | Total available nodes | $t$ | Total participating nodes |
| $m$ | Total compromised nodes | $T$ | Total communication episodes | $\sigma^2$ | Variance | $\mathcal{L}$ | Validation loss |

- We propose and design an adaptive model poisoning method utilizing episodic loss memorization technique, which we refer to as $\alpha-$`MPELM`, that ensures (a) attack persistence while maintaining (b) attack stealthiness in DPFL environments. Our comprehensive evaluation of this novel approach against leading-edge anomaly detection algorithms demonstrates its significant capability to evade these detection systems.
- We innovate in limiting the attack surface by strategically determining the DP-noise levels at the nodes using reinforcement learning (RL) [30]. This defense approach, referred to as an RL-assisted DP level selection algorithm (`rDP`), shows promising results in our evaluations, effectively converging to an optimal policy that disincentivizes the adversarial motivations.

**Roadmap, Notation, and Keywords.** Section II, covers the preliminaries. Section III outlines some contrasting points between this work and state-of-the-art literature. Section IV outlines the threat model and the basic mechanism of a DP-exploited poisoning attack. We present our proposed attack-defense strategies in section V and VI. Section VII provides empirical evaluations of the attack-defense strategies while section VIII serves as a conclusion and future research direction. Table I describes the major symbols used in this paper. We use 'smart meters', 'clients', 'edge nodes', and 'nodes' interchangeably throughout the rest of the article. Also, 'aggregator' & 'remote station' have been used interchangeably.

## II. PRELIMINARIES

In CIs, the sensory data hold the private and confidential information of the clients and organizations. Typically, CI authorities gather this data for storage on central servers, utilizing it to enhance the performance of their Machine Learning (ML) applications and optimize operational states. For instance, an electric utility company might analyze energy consumption data to refine load balancing and project future demands. However, this practice poses a risk: adversaries could potentially extract and exploit sensitive information, such as client whereabouts and energy usage patterns, from ML training data. Furthermore, critical operational details of a CI could be inferred and manipulated by altering model parameters like weights and biases [5], [31].

### A. Mechanism of Federated Learning (FL)

Federated Learning (FL), proposed by McMahan et al. [1] offers a solution to data privacy concerns in Machine Learning (ML) by employing a multi-node environment. In this approach, the model is trained on a diffuse network of edge nodes, each using its local data. This setup ensures that clients' private data remain confined to the edge nodes, thereby providing a degree of privacy protection. The FL process unfolds over several episodes, each comprising three key steps.

**Step I:** The global aggregator shares the parameters of the global model with all participating edge nodes.

**Step II:** Using their local data and the global model parameters, the nodes train their local models. Various optimization algorithms, such as batch gradient descent (BGD), stochastic gradient descent (SGD), or mini-batch gradient descent (mBGD), can be employed for this purpose, with mBGD being particularly suitable in this context. For instance, the $k$th node (where $k \in N$) updates its local model $\Delta w_k^{(t)}$ at episode $t = 1, 2, ..., T$ as: $\Delta w_k^{(t)} = w_k^{(t)} - w_g^{(t)}$. Here $w_g^{(t)}$ is the global model and $w_k^{(t)}$ is the optimized local model. Here, $w_k^{(t)}$ is computed by taking a step towards the mini-batch gradient descent as follows:

$$w_k^{(t)} \leftarrow w_g^{(t)} - \alpha . \frac{\partial \Phi(w_g^{(t)}, \jmath_k^{(t)})}{\partial w_g^{(t)}} \tag{1}$$

where $\alpha$ is the learning rate, $\jmath_k^{(t)}$ is the mini batches of the local training data ($\mathcal{D}_k$) and $\Phi(w_g^{(t)}, \jmath_k^{(t)})$ is the objective function to be minimized. The updated local models are then communicated back to the global aggregator.

**Step III:** The global aggregator integrates these trained local models using advanced aggregation rules, such as *FedAvg*, *FedSGD*, *Krum*, *Trimmed Mean*, or *Median*. For example, the naive mean aggregation rule updates the global model ( $w_g^{(t+1)}$) at the end of each episode, $t$ as:

$$w_g^{(t+1)} \leftarrow w_g^{(t)} + \frac{1}{n} \left[ (\sum_{k=1}^{n} \Delta w_k^{(t)} \right] \tag{2}$$

However, this simple aggregation rule is vulnerable under an adversarial setting, as an attacker can manipulate the global model through a single edge node [29], [13]. Therefore, we

adopt a more robust averaging-based aggregation approach (detailed in section V-A). While some methods (e.g., [32]) recommend stochastic client selection processes to enhance convergence and accuracy, our study focuses on the adversarial impacts on privacy-enhancing FL processes. Consequently, we employ general random sampling methods [33] for simplicity, as detailed in section V-A. This decision aligns with our primary objective of investigating the security aspects of FL, rather than developing accuracy-enhanced FL processes.

### B. Local Differential Privacy (LDP) with Gaussian Noise

Despite the inherent data privacy protection, FL is found to be vulnerable to membership inference attacks (MIAs) [18]. To address this vulnerability and protect the client's confidentiality, DP has been proposed to integrate with FL processes [7], [18]. DP employs noise-adding mechanisms, such as Gaussian noise, to perturb data before it is shared or aggregated [18]. This ensures that individual data points remain confidential even if the data is intercepted [7], [34]. By preserving privacy at the data collection stage and providing quantifiable privacy guarantees, DP enhances the reliability and trustworthiness of data analysis and decision support mechanisms [20]. It allows analysts to derive meaningful insights and make informed decisions without compromising individual privacy.

Two primary approaches of DP are– (1) global DP (GDP) [17], [34], and (2) local DP (LDP) [22], [23]. GDP perturbs models during aggregation, whereas LDP perturbs models at the edge nodes before transmission. Given its stricter privacy standards, LDP is increasingly favored in FL to protect client privacy [35]. Nonetheless, many variants of both GDP and LDP can be found in the literature [36], [7], [18], [34], [23], [19], [6], [20], each employing different DP mechanisms, including randomized response (satisfies $\varepsilon$-DP) [6], [19], [36], Laplace (satisfies $\varepsilon$-DP), Gaussian (satisfies $(\varepsilon, \delta)$-DP) [18], [7], [34], [20], Exponential (satisfies $\varepsilon$-DP) [23], Geometric (satisfies $\varepsilon$-DP) [22] and Binomial (satisfies $(\varepsilon, \delta)$-DP) mechanisms. In general, their underlying principles are the same: adding randomized noise or responses to the original data to protect the sensitive information of the clients [37].

The Gaussian mechanism is particularly popular due to two advantages: (a) additive noise, which allows for straightforward statistical analysis, and (b) natural noise, which mimics the statistical properties of typical database query noise. Here, noise is drawn from a zero-mean Gaussian distribution, with its probability density function (PDF) as

$$f_0(r) = \frac{1}{\sqrt{2\pi}\sigma_r} e^{-\frac{(r-\theta)^2}{2\sigma_r^2}} \tag{3}$$

where $\theta$ is the mean and $\sigma^2$ is the variance. However, $(\varepsilon, \delta)$-DP only satisfies if $\sigma \geq c\mathcal{S}/\varepsilon$ where $c^2 > 2\ln(1.25/\delta)$. Here, $\varepsilon$ is the permitted privacy loss or simply, privacy budget, $\delta$ is the probability of exceeding the privacy budget, and $\mathcal{S}$ is the local sensitivity. This can be formally defined as [38]:

*Definition 1:* Let $\mathcal{X}$ be a set of possible values and $\mathcal{Y}$ the set of noisy values. A local randomizer $\mathcal{M}$ is $(\varepsilon, \delta)$-locally differentially private (LDP) if $\forall x, x' \in \mathcal{X}$ and $\forall y \in \mathcal{Y}$: $Pr\left[\mathcal{M}(x) = y\right] \leq e^{\varepsilon} \times Pr\left[\mathcal{M}(x') = y\right] + \delta$

Privacy loss is a key metric in DP that evaluates the potential risk of identifying an individual's data from the output of a randomized algorithm. It assesses the change in the likelihood of two possible outcomes depending on whether a specific data point is included or excluded from the dataset. More formally, in DP, the privacy loss function quantifies the influence of adding or removing a single data point on the output of a differentially private mechanism, governed by the privacy parameter $\varepsilon$, often referred to as the privacy budget. Under the aforementioned constraint and definition, in our $\mathcal{L}$-DPFL approach, keeping track of the spent privacy budget is crucial, especially with multiple queries, as $\delta$ accumulates over time. As a solution, we employ a moments accountant technique, akin to [17], to monitor and control this budget, ceasing training when a predefined threshold is reached. The final local model update of the $k$th edge node is $\widetilde{\Delta w}_k^{(t)} \leftarrow \Delta w_k^{(t)} + \eta$. After episode $t$, the global model update is

$$w_g^{(t+1)} \leftarrow w_g^{(t)} + \frac{1}{n}\left[\sum_{k=1}^{n} \widetilde{\Delta w}_k^{(t)}\right] \tag{4}$$

## III. Literature Review

### A. Model Poisoning in Federated Learning (FL)

Recent developments in the ML community have introduced numerous model poisoning attack-defense methods applicable to advanced FL mechanisms [26], [27], [28], [29], [7], [39], [40], [9], [13]. Most of these methods focus on addressing *Byzantine failures*, where a group of curious/semi-honest/malicious nodes manipulates local raw data (data poisoning) or model parameters (model poisoning) before sending updates to the global aggregator. A notable solution to Byzantine failures is the *Krum* algorithm [27], which uses *majority-based* and *squared-distance* approaches for computing local models. Specifically, it computes $n - f - 2$ local models for each local model $w_k$, where $n$ is the total participating models and $f$ is the Byzantine models and provides theoretical guarantees for the convergence if $f < (n-2)/2$. However, *Krum*'s effectiveness is limited in environments with a large number of nodes, common in CIs.

Additionally, Krum [27]– even though effective against obvious outliers since it selects a single model update that is most consistent with others– can reject legitimate updates, particularly when they contain Differential Privacy (DP)-induced noise. More specifically, Gaussian noise introduces variability in local updates, increasing their inconsistency with other updates. Krum [27] may mistakenly interpret this inconsistency as evidence of malicious intent. This result in higher global loss even in non-attack scenarios. The $\alpha$−MPELM attack amplifies this limitation by crafting malicious updates that exploit the noise threshold and camouflage adversarial updates¬, eventually misleading Krum [27] into misclassifying benign updates as adversarial. This dual misclassification raises both attack and non-attack losses. Likewise, the other Byzantine-robust aggregation methods are designed to discard extreme values (Trimmed Mean [29], Median [29]) or aggregate only consistent subsets of updates (Bulyan [28]). They reduce the attack's relative impact by rejecting outliers, including

malicious updates. However, the rejection of legitimate noisy updates increases absolute loss across all scenarios.

Two defense techniques against local model poisoning in Byzantine-robust FL are introduced in [26], focusing on directed deviation and deviation goals for attackers. Following the *directed deviation* goal, the attacker aims to deviate a global model parameter the most towards the inverse of the before-attack direction. Under the *deviation* goal, the direction change of the global model parameter is not considered. The study reveals vulnerabilities in *Krum*, *Trimmed Mean*, and *Median*, and proposes *ERR* and *LFR* as countermeasures, generalizing earlier techniques like *RONI* [41] and *TRIM* [42]. A mixed detection method (*ERR* + *LFR*) is also effective in some scenarios. However, these approaches only focus on attaining maximum degradation in model utility and do not address the stealthiness aspect of attacks nor the privacy concerns often required by FL users [15].

A closely related study [7] explores both data and model poisoning attacks in FL, proposing a weight-based detection method using a validation dataset. This method, consisting of Norm and Accuracy detection mechanisms, resembles the *ERR* and *LFR* approaches from [26] in operational principles. Nonetheless, unlike [26], they introduce $\gamma$ as a degree of influence for their Mix detection technique (*remark:* we use the symbol $\gamma$ in this paper for describing the degree of poisoning which bears a different meaning than this). They evaluate their Norm, Accuracy, and Mix detection approaches in the presence of randomized malicious devices (RMD). They also propose a multi-layer $(\varepsilon, \delta)$-GDP technique for balancing privacy-utility trade-offs in DP. In contrast, to realize a stringent definition of privacy without loss of generality, we make use of LDP in this paper. Our work differs as we focus on protecting local model parameters rather than raw training data and consider DP-noise not only as a privacy tool but also as a means for model poisoning attacks. We argue that the in-transit model parameters are more vulnerable to inference attacks than the raw training data. Thereby model poisoning attacks are more likely to cause irrevocable utility damages than data poisoning attacks. Later, we show that our proposed attack can deceive their anomaly detection techniques more effectively than conventional RMD attacks.

Similarly, [43] introduces *LoMar*, a two-phase defense algorithm against FL poisoning attacks, utilizing kernel density estimation to score and filter local model updates. However, *LoMar* does not account for attacks leveraging additional DP-noise. In contrast, our approach considers malicious noise drawn from an adversarial distribution resembling benign Gaussian distribution, making it challenging for techniques like *LoMar* to distinguish between malicious and benign updates. Fig. 2 illustrates the mechanism of Gaussian DP-noise exploitation in our proposed attack model. It demonstrates the process where LDP is applied to node updates, incorporating noise that could cause the anomaly detection to incorrectly classify certain anomalous updates ($\eta_a$) as non-anomalous, owing to the adjusted anomaly detection range, $\mathcal{R}'$.

### B. Exploitation of Differential Privacy (DP)

Research has explored the potential misuse of DP in classification settings, with studies like [24], [15], [16] examining how DP-noise can be used to diminish system utility. These works introduce optimal adversarial distributions for generating noise and propose bad data detection (BDD) mechanisms as a defense. A game-theoretic framework is employed to evaluate such defenses, with solutions framed as Nash equilibria. However, the application of these models in multi-agent systems like FL and the control of poisoning intensity remain unaddressed, as do preemptive attack surface reduction issues that our `rDP` algorithm in this paper aims to resolve.

Our investigation extends beyond existing studies by focusing on a meticulous challenge– keeping the attack persistent, and stealthy. We introduce methods to maintain persistent, robust, and stealthy attacks across FL communication episodes. The random node selection prevalent in FL can neutralize adversarial efforts if not consistently applied. The comparative analysis with related works is detailed in Table II.

TABLE II
COMPARATIVE ANALYSIS AMONG DPFL SYSTEMS. SYMBOL: ADDRESSED(✓), NOT ADDRESSED(□). "**F**"EDERATED LEARNING. "**Po**"ISONING ATTACKS. "**D**"IFFERENTIAL PRIVACY ("$\mathcal{L}$"OCAL DP OR "$\mathcal{G}$"LOBAL DP). "**E**"XPLOITATION OF DP TO CONDUCT POISONING ATTACKS. "**T**"RACKING OF PRIVACY BUDGET SPENDING. "**I**"NTELLIGENT "**P**"RIVACY "**L**"EVEL "**S**"ELECTION STRATEGY

| System | F | Po | D | | E | T | IPLS |
|---|---|---|---|---|---|---|---|
| | | | $\mathcal{L}$ | $\mathcal{G}$ | | | |
| Fang et al., 2020 [26] | ✓ | ✓ | □ | □ | □ | □ | □ |
| Giraldo et al., 2020 [15] | □ | ✓ | ✓ | ✓ | ✓ | □ | □ |
| Zhao et al., 2020 [19] | ✓ | □ | ✓ | □ | □ | ✓ | □ |
| Hu et al., 2020 [20] | ✓ | □ | ✓ | □ | □ | ✓ | ✓ |
| Wen et al., 2021 [6] | ✓ | □ | ✓ | □ | □ | □ | □ |
| Zhou et al., 2022 [7] | ✓ | ✓ | □ | ✓ | □ | ✓ | ✓ |
| Li et al., 2022 [21] | ✓ | ✓ | ✓ | □ | □ | □ | □ |
| Zhu et al., 2023 [3] | ✓ | □ | ✓ | □ | □ | □ | □ |
| Lu et al., 2023 [13] | ✓ | ✓ | □ | ✓ | □ | □ | □ |
| Chen et al., 2024 [9] | ✓ | ✓ | □ | □ | □ | □ | □ |
| **This work** | ✓ | ✓ | ✓ | □ | ✓ | ✓ | ✓ |

## IV. PROBLEM FORMULATION

This section outlines the Gaussian noise exploitation mechanism and formulates the adversarial noise crafting challenges.

### A. Basic Mechanism of Gaussian Noise Variance Exploitation

**DP not included.** In a non-DP scenario, anomaly detectors expect local model updates within a certain range, $\mathcal{R} = [\Delta w_e^{(t)} \pm \tau]$, where $\Delta w_e^{(t)}$ is the expected update and $\tau$ is a predefined threshold. An update from $k$th node ($\Delta w_{k_r}^{(t)}$) that exceeds $\mathcal{R}$ triggers an alarm.

**DP included.** Now, consider that the authority enforces DP as a privacy-preservation tool. Hence, the local update adjusts to accommodate Gaussian noise $\pm\hat{\eta}$, resulting in a modified update as $\widetilde{\Delta w}_{k_r}^{(t)} \leftarrow \Delta w_{k_r}^{(t)} \pm \hat{\eta}$. Then, to prevent false positives, the anomaly detector also needs to adjust its detection range as, $\mathcal{R}' = [\Delta w_e^{(t)} \pm \tau']$ where the new detection threshold, $\tau' = \pm(\tau + \hat{\eta})$. This little adjustment in the detection range

potentially opens an additional (false) noise injection window for the attacker. The range is as follows:

$$\begin{aligned} Lower: \left[0, (\Delta w_{k_r}^{(t)} - \hat{\eta}) - (\Delta w_e^{(t)} - \tau')\right] \\ \Rightarrow \left[0, \Delta w_{k_r}^{(t)} - \Delta w_e^{(t)} + \tau\right] \Rightarrow [0, \tau - \upsilon] \\ Upper: \left[0, (\Delta w_e^{(t)} + \tau') - (\Delta w_{k_r}^{(t)} + \hat{\eta})\right] \\ \Rightarrow \left[0, \Delta w_e^{(t)} - \Delta w_{k_r}^{(t)} + \tau\right] \Rightarrow [0, \tau + \upsilon] \end{aligned} \quad (5)$$

where $\upsilon$ is the deviation of the local update from the expected update, i.e., $\upsilon = \Delta w_{k_r}^{(t)} - \Delta w_e^{(t)}$. The adversary can exploit this false noise injection or poisoning window (i.e., $[0, \tau \pm \upsilon]$) to craft an adversarial noise profile, $\eta_a \leftarrow \mathcal{N}_a(\mu_a, \frac{S}{\varepsilon})$ where $\mu_a$ is the desired deviated mean or simply attack impact. Here, adversarial noise profile describes the intentional disturbances generated by attackers to interfere with the training process of federated learning systems. Unlike arbitrary or random noise, adversarial noise is strategically designed to achieve specific goals, such as impairing model accuracy or evading detection mechanisms. In this context, an adversarial noise profile $\eta_a$ represents the attacker's method of injecting noise into the system to disrupt the learning process while mimicking legitimate privacy-preserving noise, such as DP-induced noise, to avoid detection. If noise is increased, deviation $\upsilon$ increases which in turn expands the poisoning window. Larger DP-noise correlates with increased $\upsilon$ and a wider attack window, implying *more privacy leads to greater attack surfaces and utility degradation*.

Fig. 2 visualizes this concept. Without DP, updates like $w_{1_r}^{(t)}$ are deemed non-anomalous if within $\mathcal{R}$. With LDP, Gaussian noise $\eta$ may push $\upsilon$ beyond $\mathcal{R}$, causing benign updates ($w_{2_r}^{(t)}$) to appear anomalous. Adjusting $\mathcal{R}'$ to incorporate $\eta$ opens a poisoning window ($[0, (\tau \pm \upsilon)]$) for attackers, as malicious updates like $w_{3_r}^{\prime(t)}$ might not be distinguishable from benign ones when they fall within the modified range. Thus, the detector faces a dilemma: it cannot differentiate between benign and malicious updates if the adversarial noise magnitude remains within the newly accommodated window.

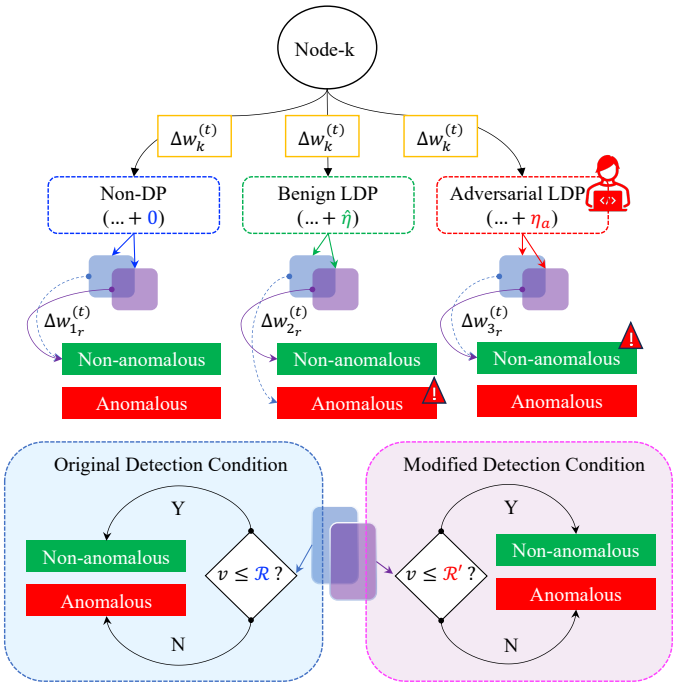


Fig. 2. Gaussian noise exploitation: A non-adjusted anomaly detection range ($\mathcal{R}$) incorrectly flags valid LDP updates ($\Delta w_{2_r}^{(t)}$) as anomalous. Conversely, an adjusted anomaly detection range ($\mathcal{R}'$), meant to reduce false positives, may mistakenly classify adversarial updates ($\Delta w_{3_r}^{(t)}$) as non-anomalous.

### B. Challenges in Crafting Adversarial Noise Profile

Crafting an adversarial noise profile presents multiple challenges due to the hidden parameters within the anomaly detection mechanisms. For instance, it is not feasible to presume that an attacker can straightforwardly determine the value of $\eta_a$, as the parameters $\tau$ and $\upsilon$ are secured and concealed within the anomaly detection system. To understand the problem more clearly, let us consider that $i$ is a particular compromised edge node and $M$ is the set of all compromised nodes having cardinality of $m$ (i.e., $i \in M$ and $|M| = m$) in a $\mathcal{L}$-DPFL setting. Then the number of benign edge nodes is $b = n - m$ where $n$ is the total number of participating edge nodes (i.e., $|N| = n$). Let us also consider $j$ is an individual benign edge node while the set of benign nodes is $B$ (i.e., $j \in B$ and $|B| = b$). Now, if the malicious noise is $\eta_{ai}^{(t)}$ and the benign DP-noise is $\eta_{bj}^{(t)}$, then $i$'s adversarial local update ($\widetilde{\Delta w}_i^{(t)}$), $j$'s benign local update ($\widetilde{\Delta w}_j^{(t)}$), and aggregated global update ($w_g^{(t+1)}$) can be represented by (6), and (7) respectively.

$$\widetilde{\Delta w}_i^{(t)} \leftarrow \Delta w_i^{(t)} + \eta_{ai}^{(t)}, \ \widetilde{\Delta w}_j^{(t)} \leftarrow \Delta w_j^{(t)} + \eta_{bj}^{(t)} \quad (6)$$

$$w_g^{(t+1)} \leftarrow w_g^{(t)} + \frac{1}{n}\left[\sum_{i=1}^{m} \widetilde{\Delta w}_i^{(t)} + \sum_{j=1}^{b} \widetilde{\Delta w}_j^{(t)}\right] \quad (7)$$

However, the major challenge for the attacker here is to craft the adversarial noise profile, $\mathcal{N}_a(\mu_a, \frac{S}{\varepsilon})$ and choose the magnitude of the adversarial noise, $\eta_a$ for subsequent FL episodes. To tackle this adversarial challenge, [26] followed a maximum utility degradation approach, where a large amount of noise was injected into the model parameters through subsequent FL episodes. Nonetheless, this large adversarial noise may potentially lead to easier attack detection– thus violating the stealthiness objective of the attack.

Another potential attack strategy involves mimicking the benign Gaussian noise distribution. Specifically, the attacker could draw $\eta_a$ from an adversarial distribution ($f_a$) similar to a benign Gaussian distribution ($f_0$) and, then inject $\eta_a$ into total $m$ compromised local models. In other words, if $i$ is a compromised *edge node* ($i \in N$- set of all participating nodes) out of the total $m$ compromised nodes, then the set of the malicious local model at episode $t$ is

$$\Delta w_m^{(t)} = \{\Delta w_i^{(t)} + \eta_{ai}^{(t)}\}_{i=1,2,\ldots m} \ \forall i \in N \text{ if } 0 \leq m \leq n \quad (8)$$

Such optimal attack distribution, $f_a^*$, and the optimal attack impact, $\mu_a^*$ are derived and presented in [15] by solving a

multi-criteria optimization problem that addresses two conflicting adversarial goals: *(1) maximum damage,* and *(2) minimum disclosure*. The goals are contradicting in nature from the adversarial point of view since *maximum damage* can lead to easier attack detection whereas *minimum disclosure* limits the damage. The optimal adversarial distribution, $f_a^*$, and the optimal attack impact, $\mu_a^*$ for the Gaussian mechanism are expressed as:

$$f_a^*(x) = \frac{1}{\sqrt{2\pi}\sigma_x} e^{-\frac{(x-\theta-\sqrt{2\gamma}\sigma_x)^2}{2\sigma_x^2}} \text{ and } \mu_a^* = \theta + \sqrt{2\gamma}\sigma_x \quad (9)$$

where $\theta$ is the mean, $\sigma_x^2$ is the variance, and $\gamma$ is the stealthiness parameter (i.e., degree of poisoning). The choice of $\gamma$ is critical; too high a value increases detectability, while too low leads to negligible impact ($\gamma = 0$, $f_a^* = f_0$ and $\mu_a^* = \theta$, comparing (3) and (9)). Thus, the attacker must fine-tune $\gamma$ for each episode to ensure the attack remains covert while still inflicting significant damage. Now, this raises below research questions that we subsequently answer through our theoretical and empirical analysis, as detailed in section V-B, VI-B, and VII-B of this paper.

- How does the attacker tune $\gamma$ at every FL episode?
- What are the attack impacts in $\mathcal{L}$-DPFL based CIs?
- What could be an effective defense against this attack?

### C. Threat Model

*1) Attacker's Capability:* We consider an attacker who falsifies model parameters returned to the aggregator. This attack could be launched by an insider compromising vulnerable edge nodes (insider threat) or an outsider intercepting communication paths to the aggregator (outsider threat). As we are agnostic to how the attacker can modify the model parameters, our proposed attack model covers both insider and outsider threats. Particularly, we assume that the attacker gets unauthorized access to a few local models irrespective of the attack vector. We assume the attacker gains unauthorized access to a selected number of local models; if too many are compromised, the attack would trivially manipulate the global model and be easily detectable [26]. The attacker's knowledge of benign participants is minimal, and direct control over the FL aggregation algorithm is out of their reach, especially since the aggregator is typically a secure, benign server.

*2) Attacker's Background Knowledge:* For an insider threat, the attacker could access both local training data and model parameters, whereas an outsider might only access in-transit model parameters. *Following a conservative approach, we assume the attacker's background knowledge is only limited to a few local model parameters for both threat models.* We argue that our proposed attack algorithm would perform much better if the attacker gets access to the local training data. In alignment with established practices in DP implementations [44], [45], we assume that attackers have access to the publicly disclosed privacy budget ($\varepsilon$) and noise distribution mechanisms. This assumption reflects the transparency commonly adopted in FL systems to maintain trust among participants. DP parameters, including $\varepsilon$ values and noise distributions (e.g., Gaussian, Laplace), are often shared publicly as part of system documentation. Cynthia Dwork, one of the pioneers of DP, has advocated for the establishment of an "Epsilon Registry"—a communal repository to promote awareness and adoption of robust DP implementations [45]. Similarly, [46] emphasized that user trust and willingness to share data improve when $\varepsilon$ values are transparently disclosed. Major organizations such as Apple, the US Census Bureau, and Google have already publicized $\varepsilon$ values for their DP systems, with ranges such as Apple's 2–16, the US Census Bureau's 19.61, and Google's 2.64 [47]. These disclosures illustrate a growing trend in transparency– reinforcing the feasibility of our assumption that attackers in real-world FL systems could gain knowledge of Gaussian noise parameters.

*3) Attacker's Goal:* The attacker's primary goal is to achieve *(a) maximum damage* while *(b) avoiding detection* in any stage of the attack. Achieving significant damage requires introducing very large (or very small) adversarial noise, which risks detection by conventional anomaly detectors. Conversely, to avoid detection, the noise should closely align with the boundaries of the poisoning range (5), potentially compromising damage. Hence, the attacker must strike a balance between these conflicting goals to optimize damage and stealth.

## V. Proposed Model Poisoning Attack

In this section, we outline our model poisoning attack strategy within the $\mathcal{L}$-DPFL framework.

### A. $\mathcal{L}$-DPFL Architecture

The $\mathcal{L}$-DPFL structure we employ, akin to a smart metering network (Fig. 3), is represented by a two-layer network where edge nodes serve as FL clients and the remote station acts as the FL server or aggregator. This simplification, while practical, does not diminish the realism of a multi-layer network as highlighted by [7], where aggregation occurs at multiple layers, but FL training and LDP integration are primarily at the edge layer. The potential for additional DP noise in successive layers, creating new attack vectors, is acknowledged but not the focus of our model. The entire $\mathcal{L}$-DPFL process is pseudocoded in Algorithm 1.

*1) Training Phase:* In our $\mathcal{L}$-DPFL method, one remote station collaborates with $n$ randomly selected edge nodes from a total of $\mathcal{K}$ nodes. These edge nodes, with similar neural network structures, are initialized with shared global model parameters $w_g^{(t)}$. Next, the nodes perform local optimization that involves mini-batch gradient descent on sampled datasets $\jmath_k^{(t)}$ from local training datasets $\mathcal{D}_k$, resulting in trained parameters $w_k^{(t)}$ and subsequent local updates $\Delta w_k^{(t)} = w_k^{(t)} - w_g^{(t)}$.

*2) Update Clipping Phase:* Let $w = (w_1, w_2, ..., w_d)$ is a weight vector, and $\|p\|$ denotes the $\ell_2$-norm of a $q$-dimensional vector $p = (p_1, p_2, ..., p_q)$, i.e., $\|p\| = \sqrt{\sum_{i=1}^{q} p_i^2}$. Assume that $\mathcal{W}$ is the maximum $\ell_2$-norm value of all weights for any given weight vector $w_k^{(t)}$ and sampled dataset $\jmath_k^{(t)}$, i.e., $\mathcal{W} = max_{w_k^{(t)} \in \mathbb{R}, \jmath_k^{(t)} \in \mathcal{D}_k} \mathbb{E}\left[\|w_k^{(t)}(\jmath_k^{(t)})\|\right]$. To keep the model usable and prevent over-fitting, each edge node clips its local model updates by a clipping threshold value $\mathcal{C} \in (0, \mathcal{W}]$ as

$$\Delta w_{k\chi}^{(t)} \overset{\text{clip}}{=} \Delta w_k^{(t)} / max(1, \frac{\|\Delta w_k^{(t)}\|}{\mathcal{C}}) \quad (10)$$

**Algorithm 1:** $\mathcal{L}$-DPFL Protocol. $N$: Set of edge nodes with cardinality $\mathcal{K}$, $\sigma^2$: variance, $\mathcal{C}$: Clipping param., $T$: Total episode, $\alpha$: Learning rate, $\varepsilon$: Privacy loss, $\delta$: Privacy leakage probability, $\mathcal{S}$: Sensitivity, $\mathcal{D}$: Training dataset, $\mathcal{W}$: Max $\ell_2$-norm, $\xi$: Noisy clipped local model updates, $\eta$: Gaussian noise, $\mathcal{N}$: Noise profile, $\Pi$: Privacy accountant, $w$: Model parameter

**Input:** $N$, $\sigma$, $\mathcal{C}$, $T$, $\alpha$
**Output:** New global model parameters, $w_g^{(t)}$
**Data:** Mini batch of training set, $\{\jmath_k \subset \mathcal{D}_k\}_{k=1}^n$
**Privacy Guarantee:** satisfies $(\varepsilon, \delta)$-LDP with Gaussian noise $\mathcal{N}(0, \mathcal{S}^2\sigma_k^2\mathbb{I}_q)$

$w_g^{(t)} \leftarrow$ random initialization
Initialize privacy accountant, $\Pi(\varepsilon, \mathcal{K})$
**for** *each* $t = 1, 2, ..., T$ *episode* **do**
  $\delta \leftarrow \Pi(n_t, \sigma_t)$
  **if** $\delta > \tau_\delta$ **then** return $w_g^{(t)}$
  **else** $\xi^{(t)} \leftarrow$ NoisyUpdates$(N, \sigma, C, w_g^{(t)})$
    $w_g^{(t+1)} = w_g^{(t)} + \frac{1}{n}\sum_{k=1}^n \xi^{(t)}$
**end**
**Function** NoisyUpdates$(N, \sigma, C, w_g^{(t)})$**:**
  **for** *each edge node* $k \in N$ **do**
    $w_k^{(t)} \leftarrow w_g^{(t)} - \alpha.\frac{\partial\Phi(w_g^{(t)}, \jmath_k^{(t)})}{\partial w_g^{(t)}}$
    $\Delta w_k^{(t)} \leftarrow w_k^{(t)} - w_g^{(t)}$
    $\mathcal{W} \leftarrow max_{w_k\in\mathbb{R}, \jmath_k^{(t)}\in\mathcal{D}_k} \mathbb{E}\left[\|w_k^{(t)}(\jmath_k^{(t)})\|\right]$
    Set clipping threshold $\mathcal{C} \in (0, \mathcal{W}]$
    Clip the local model updates as
    $\Delta w_{k\chi}^{(t)} \overset{\text{clip}}{\leftarrow} \Delta w_k^{(t)}/max(1, \frac{\|\Delta w_k^{(t+1)}\|}{\mathcal{C}})$
    Add Gaussian noise to obtain
    $\widetilde{\Delta w}_{k\chi}^{(t)} \leftarrow \Delta w_{k\chi}^{(t)} + \eta_k \sim \mathcal{N}(0, \mathcal{S}^2\sigma_k^2\mathbb{I}_q)$
  **end**
  Set $\xi^{(t)} \leftarrow \{\widetilde{\Delta w}_{k\chi}^{(t)}\}_{k=1}^n$
  **return** $\xi^{(t)}$

where $\chi$ denotes the clipping technique.

*3) LDP Integration Phase:* After clipping the local model updates by clipping threshold $\mathcal{C}$, the $k$th node implements the $(\varepsilon, \delta)$-LDP by adding a Gaussian noise component $\eta_k$. Since, $\Delta w_{k\chi}^{(t)}$ is bounded by $\mathcal{C}$ and can be changed at most by $\mathcal{C}$, the local sensitivity, $\mathcal{S}$ of the aggregation operation is equivalent to $\mathcal{C}$. Therefore, the Gaussian noise variance of each dimension is proportional to $\mathcal{S}^2$, i.e., $\eta_k \sim \mathcal{N}(0, \mathcal{S}^2\sigma_k^2\mathbb{I}_q)$ for some $\sigma_k^2 > 0$, where $\mathbb{I}_q$ is the $q \times q$ identity matrix. Then, the noisy clipped local model updates can be represented as

$$\widetilde{\Delta w}_{k\chi}^{(t)} = \Delta w_{k\chi}^{(t)} + \eta_k \sim \mathcal{N}(0, \mathcal{S}^2\sigma_k^2\mathbb{I}_q) \tag{11}$$

The noisy clipped local model updates from all edge nodes $\xi^{(t)} \leftarrow \{\widetilde{\Delta w}_{k\chi}^{(t)}\}_{k=1}^n$ are sent to the server for aggregation.

*4) Aggregation Phase:* The *remote station* aggregates these noisy clipped updates to form the new global model. It can be formally expressed as

$$w_g^{(t+1)} = w_g^{(t)} + \frac{1}{n}\sum_{k=1}^n \widetilde{\Delta w}_{k\chi}^{(t)} \tag{12}$$

### *B. The Adaptive Model Poisoning through Episodic Loss Memorization ($\alpha$−MPELM) Attack*

In $\mathcal{L}$-DPFL, we consider an attacker targeting $m$ compromised nodes. The attacker aims to achieve the optimal balance of attack impact $\mu_a^* = \theta + \sqrt{2\gamma}\sigma_x$ and stealthiness by adjusting the degree of poisoning $\gamma$ in each FL episode and drawing adversarial noise $\eta_a \sim \mathcal{N}_a(\mu_a, \frac{\mathcal{S}}{\varepsilon})$ from $f_a^*$, as expressed in (9). To understand how an intelligent attacker can compute the required *episodic degree of poisoning*, $\gamma^{(t)}$ at every $t$ episode in FL, we introduce the concept of the adaptive model poisoning process through episodic loss memorization ($\alpha$−MPELM) technique. Here, the episodic loss memorization is an adaptive attack technique where the attacker adjusts the poisoning intensity based on the validation losses across episodes. $\alpha$−MPELM enables attackers to deceive anomaly detection algorithms by first calculating the FL model loss for any arbitrary $\gamma$ value, and then adjusting $\gamma$ based on the losses of subsequent episodes. By leveraging the memorization of episodic loss patterns, attackers can dynamically fine-tune their strategies. Specifically, the attacker increases the value of $\gamma$ (i.e., increasing attack intensity) if the loss is relatively low and decreases when the loss remains the same as previous. When the loss increases, $\alpha$−MPELM instructs the attacker to halt injecting malicious noise temporarily to avoid triggering early detection by anomaly detection systems. This adaptive approach allows the attacker to systematically alter the attack's

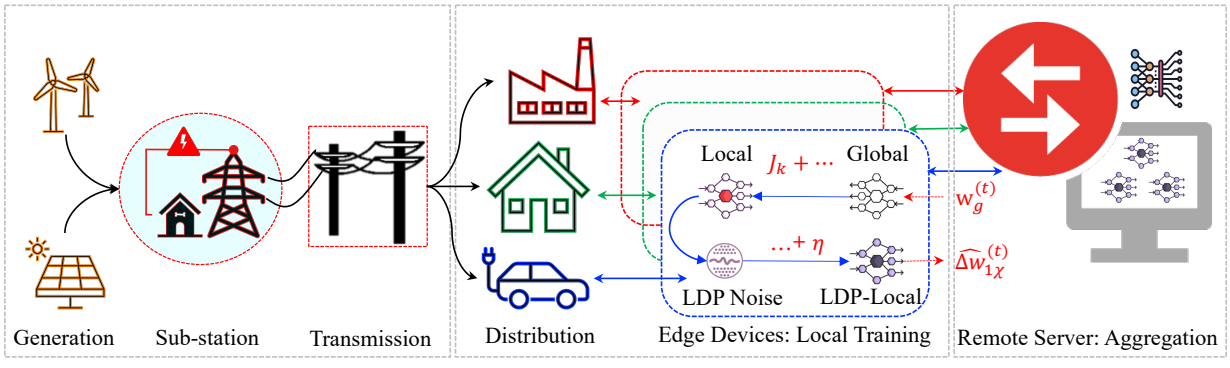


Fig. 3. Local differentially private federated learning ($\mathcal{L}$-DPFL) architecture in smart grid network.

intensity based on the observed model loss— making the malicious activity blend seamlessly with legitimate operations. By continuously adapting to the model's performance, $\alpha-$MPELM can maintain the attack over multiple episodes without raising suspicion. This method of episodic loss memorization and adaptive poisoning is detailed in the following steps and the pseudocode of $\alpha-$MPELM is given in algorithm 2.

*1) Choosing the Initial Degree of Poisoning:* In the early stages of Federated Learning (FL) training, model losses are typically high but decrease as the system converges. Anomaly detection systems may either exclude all models with high losses or accept all, assuming the fluctuations are normal. To exploit this, the attacker sets the initial degree of poisoning $\gamma_0$ close to the privacy budget $\varepsilon$, ensuring poisoned updates blend with benign ones and avoid detection. Unlike static attacks, which apply a fixed poisoning level throughout training, episodic loss memorization dynamically adjusts the poisoning intensity $\gamma$ based on model loss patterns. This approach balances the need to prevent an excessive surge in gradient values in subsequent episodes while also retaining a significant portion of the adversarial influence Eventually, this adaptability allows the attack to evade detection by continuously monitoring and fine-tuning $\gamma$ in response to the evolving state of the model, contrasting with the predictable nature of static attacks.

*2) Calculating Episodic Loss:* At the start of each FL episode, the attacker calculates the validation loss for each compromised node using its local dataset, $\mathcal{V}_i^{(t)}$, and global model parameters, $w_g^{(t)}$. The average validation loss across all $m$ compromised nodes at episode $t$ is:

$$\tilde{\mathcal{L}}_m^{(t)} = \frac{1}{m}\sum_{i=1}^{m} \ell(w_g^{(t)}, \mathcal{V}_i^{(t)})$$

This average is then used to compute the loss ratio, which compares the current episode's performance to previous episodes to guide the attack's next steps.

*3) Computing Loss Ratio:* The loss ratio $\mathcal{R}$ is computed by comparing the current average validation loss $\widetilde{\mathcal{L}}_m^{(t)}$ with the average of all previous validation losses $\widetilde{\mathcal{L}}_m^{(-t)}$, where $(-t)$ represents all episodes before $t$. The loss ratio is formally expressed as:

$$\mathcal{R} = \widetilde{\mathcal{L}}_m^{(t)} \Big/ \widetilde{\mathcal{L}}_m^{(-t)} \quad \forall\ \widetilde{\mathcal{L}}_m^{(-t)} \neq 0;\ t > 1 \tag{13}$$

This ratio helps determine how the attack progresses based on model loss changes over time, hence it is referred to as the *episodic loss memorization (ELM)* process. Essentially, the *ELM* is a strategy used by adversaries to iteratively adapt their poisoning attacks based on feedback from the FL system. The attacker records and analyzes the loss incurred across multiple training episodes to refine the attack parameters for maximum impact.

*4) Updating Episodic Degree of Poisoning ($\gamma^{(t)}$):* The loss ratio $\mathcal{R}$ serves as a crucial indicator of how much the current global model's performance deviates from previous episodes. If $\mathcal{R} >> 1$, it indicates significant divergence in the model parameters, likely caused by excessive noise or other disruptions. In this situation, further poisoning would worsen the deviation, making the attack more detectable. Therefore, the attacker halts the poisoning for that episode ($\gamma^{(t)} = 0$) and resumes only when the model loss stabilizes.

Conversely, if $\mathcal{R} << 1$, it suggests the attack is too weak, and the model continues to converge normally. In response, the attacker increases the degree of poisoning according to the loss ratio ($\gamma^{(t)} = \gamma + \rho \cdot \mathcal{R} \cdot \gamma$). The proportionality factor $\rho$ ensures that the increase in poisoning is controlled, preventing a drastic change that might raise suspicion.

For cases where $\mathcal{R} \approx 1$, the attacker slightly reduces the poisoning intensity, aligning the updates more closely with benign ones. This strategy ensures that the attack remains stealthy, while gradually degrading the global model's performance over time- making detection difficult but still effective in weakening the model.

**Algorithm 2:** $\alpha-$MPELM Technique. $\gamma$: degree of poisoning, $\gamma^{(t)}$: Episodic degree of poisoning, $\mathcal{V}$: Validation dataset, $\mathcal{L}_m^{(t)}$: Current valid. loss, $\widetilde{\mathcal{L}}_m^{(t)}$: Current avg. valid. loss, $\widetilde{\mathcal{L}}_m^{(-t)}$: Previous avg. valid. loss, $\mathcal{R}$: Loss ratio, $\rho$: proportionality factor

**Input:** $w_g^{(t)}$
**Output:** $\gamma^{(t)}$
**Data:** $\{\mathcal{V}_i^{(t)}\}_{i=1}^{m}\ \forall\ \mathcal{V}_i^{(t)} \neq \mathcal{I}_i^{(t)}$
initialize: $\gamma \leftarrow \gamma_0$ where $\gamma_0 \approx \varepsilon$
**for** *each* $t = 1, 2, ..., T$ *episode* **do**
  set: $\mathcal{L}_m^{(t)} \leftarrow 0; \mathcal{R} \leftarrow 0$
  **for** *each compromised node* $i = 1, 2, ..., m$ **do**
    measure: $\mathcal{L}_i^{(t)} \leftarrow \ell(w_g^{(t)}, \mathcal{V}_i^{(t)})$
    calculate: $\mathcal{L}_m^{(t)} = \mathcal{L}_m^{(t)} + \mathcal{L}_i^{(t)}$
  **end**
  current avg. loss: $\widetilde{\mathcal{L}}_m^{(t)} = \frac{1}{m}(\mathcal{L}_m^{(t)})$
  avg. of episodic losses: $\widetilde{\mathcal{L}}_m^{(-t)} = Avg([\widetilde{\mathcal{L}}_m^{(e)}]_{e=1}^{(t-1)})$
  Loss ratio: $\mathcal{R} = \widetilde{\mathcal{L}}_m^{(t)} \Big/ \widetilde{\mathcal{L}}_m^{(-t)} \quad \forall\ \widetilde{\mathcal{L}}_m^{(-t)} \neq 0;\ t > 1$
  episodic degree of poisoning:

$$\gamma^{(t)} = \begin{cases} 0, & \text{if } \mathcal{R} >> 1 \\ \gamma + \rho \cdot \mathcal{R} \cdot \gamma, & \text{if } \mathcal{R} << 1 \\ \gamma - \rho \cdot \mathcal{R} \cdot \gamma, & \text{otherwise} \end{cases}$$

  Then, save $\gamma$ for the next episode as follows:
  **if** $\gamma^{(t)} \neq 0$ **then** $\gamma \leftarrow \gamma^{(t)}$ **else** $\gamma$
  call: sub-processes to inject false noise with $\gamma^{(t)}$
  append: $\widetilde{\mathcal{L}}_m^{(t)}$ into $[\widetilde{\mathcal{L}}_m^{(t)}]_{t=1}^{(t-1)}$ to obtain $[\widetilde{\mathcal{L}}_m^{(t)}]_{t=1}^{(t)}$
**end**

## VI. Proposed RL-assisted Differential Privacy Level Selection (rDP) Technique

Drawing from the analysis of optimal DP-exploited attacks in sections IV and V, it becomes evident that enhancing data privacy through the DP mechanism, particularly by employing substantial noise, could inadvertently create vulnerabilities for extensive model poisoning attacks. To counter this, one approach is to adopt a lower DP level, but this strategy

might lead to diminished data privacy, thereby increasing susceptibility to privacy breaches. Therefore, an optimal value of privacy is desirable within a DPFL framework. We propose to navigate this trade-off by intelligently adjusting the *privacy loss parameter,* $\varepsilon$, utilizing the RL technique.

### A. Defense Objectives

The primary goal for defenders, or designers of CIs, is to design a learning process that is resilient against the types of attacks we have proposed. This requires a detailed comprehension of DP parameters and the associated threat landscape. Concurrently, it is crucial to identify and establish an optimal value for the privacy loss parameter ($\varepsilon^*$). Achieving this balance is key to diminishing the potential attack surface and curtailing the attack impact ($\mu_a$).

We leverage DP parameters (e.g., privacy loss, information leakage probability, etc.) and historical federated loss to model our `rDP` defense as a countermeasure against $\alpha$−`MPELM` attack. The `rDP` technique aims to enhance FL system security by dynamically adjusting DP noise. To accomplish this, it focuses on— (1) *dynamic adjustment*: continuously monitors and adjusts DP noise levels based on performance and attack patterns, (2) *optimal policy learning*: leverages RL to find the optimal DP levels that counteract attacks while maintaining FL performance, and (3) *detection of stealthy attacks*: anticipates adaptive attacks through loss calculation.

### B. The Reinforcement Learning-based Differential Privacy Level Selection (`rDP`) Algorithm

The proposed `rDP` process, pseudocoded in algorithm 3, adopts a Q-learning approach [30]. Q-learning is based on an action-value function, which predicts the expected utility or benefit of executing a particular action in a given state.

*1) State Space:* We assume that the *state* is initialized as soon the learning starts. We define the *state space*, denoted as $S = (m_l, f_l, \varepsilon)$, where $m_l$ encapsulates the array of losses incurred by the attacker, $f_l$ represents the historical federated loss data, and $\varepsilon$ signifies the set of privacy loss. In the context of designing the `rDP` algorithm, the attacker's loss set ($m_l$) is determined through a series of pre-conducted experiments adhering to the attack methodology described in section V-B. To ensure the representativeness of $m_l$, we utilize various values of the episodic degree of poisoning ($\gamma^{(t)}$) throughout the experiments. Conversely, the federated loss ($f_l$) is computed by evaluating the global federated model in a non-adversarial environment. Importantly, for the efficacy and precision of the `rDP` algorithm, we measure both $m_l$ and $f_l$ under consistent values of $\varepsilon$ within the predefined loss set.

*2) Action Space:* Our approach adopts an event-driven paradigm where the defensive agent is programmed to respond upon the occurrence of new events. This agent continuously monitors the current state of the federated environment, $s \in S$, to inform its decision-making process. We define the *action space*, symbolized as $\mathcal{A}$, which comprises three possible actions: *increase*, *decrease*, and *static*. To enhance the precision of the agent's decision-making capabilities, we allow for the modulation of the privacy loss parameter, $\varepsilon$, either by a single or double unit increment or decrement, contingent upon the prevailing state $s \in S$. The agent then executes an action $i$ from the defined set of actions $\mathcal{A}$.

**Algorithm 3:** `rDP` process. $m_l$: Attacker's loss, $f_l$: Federated loss, $\varepsilon$: Privacy loss set, $S$: State set, $\mathcal{A}$: Action set, $\beta$: Reward func., $r$: Reward, $\alpha$: Learning rate, $Q$: Q-table, $i$: Action, $s$: State, $\pi$: Policy

**Input:** $m_l, f_l, \varepsilon, S, \beta$
**Output:** Optimal privacy loss, $\varepsilon^* \leftarrow i$
**Function** `rDP` $(m_l, f_l, \varepsilon)$ **:**
  **for** $\varepsilon_0$ *in* $\varepsilon$ **do**
    Set of States, $S_t = (m_l, f_l, \varepsilon_0)$
    Choose $i \in \mathcal{A}$ using epsilon-greedy policy
    Observe Reward, $r_{t+1}$ and State, $s_{t+1}$
    Compute: $Q^{new}(s_t, i_t) \leftarrow (1-\alpha)\,.\,Q(s_t, i_t)$ $+ \alpha\,.\,[r_t + \zeta\,.\,\overset{max}{i}\,Q(s_{t+1}, i)]$
    Policy, $\pi(s) = \overset{arg\,max}{\pi}\,Q^*(s, i)$
  **end**
  **return** $i \leftarrow \pi^*(s)$

*3) Reward Space:* In RL, the reward function plays a crucial role in guiding the defensive agent towards achieving the desired learning outcomes. It dynamically adjusts in each episode based on the input data received. With regards to safeguarding against the proposed attack, the defender's goal is twofold: (a) to minimize the maximum accuracy of the attack while simultaneously (b) maximizing the accuracy of the federated model. We assume that the maximum and minimum thresholds are predetermined and regulated by the $\mathcal{L}$-DPFL system designer. The *reward* function, crucial for directing the agent's decisions, is formalized as follows:

$$\beta = \psi_1 \frac{m_l^{max}}{m_l} + \psi_2 \frac{f_l^{max}}{f_l} + \psi_3 \frac{1}{\varepsilon} \tag{14}$$

where $m_l^{max}$ and $f_l^{max}$ represent the maximum values of the attack loss and federated loss, respectively, while $\psi_1$, $\psi_2$, and $\psi_3$ are parameters that balance these factors. We acknowledge that the reward function could theoretically be different than (14). Nonetheless, our choice of $\beta$ is based on empirical validation and theoretical considerations. It has been tailored to our specific scenario to ensure robustness and effectiveness in achieving the desired learning outcomes. Specifically, $\beta$ is designed in such a way that it can balance the objectives of minimizing the attack's accuracy while maximizing the federated model's performance. It dynamically adjusts the reward in each episode based on the input data received.

The balance between exploration and exploitation is managed through an epsilon-greedy policy [48], starting with an exploration probability of 1.0 and gradually decreasing this probability across episodes to a minimum threshold, set at 0.05 in this study. The exploration probability of 1.0 ensures extensive initial exploration, allowing the RL agent to gather diverse experiences and avoid premature convergence to suboptimal policies. This probability is gradually decreased to 0.05 to balance exploration and exploitation, enabling the agent to leverage the knowledge it has acquired to make more informed decisions. Moreover, for simplicity, we select the *maximum number of episodes* as the stopping criterion.

This article has been accepted for publication in IEEE Transactions on Artificial Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAI.2025.3540030



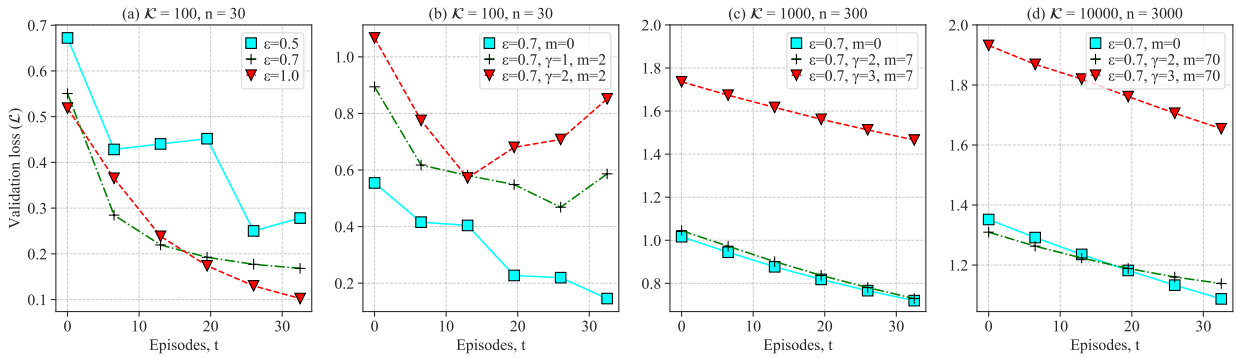


Fig. 4. Adversarial impact for varying privacy loss ($\varepsilon$), degree of poisoning ($\gamma$), and the number of malicious models ($m$).

TABLE III
DATASET DESCRIPTION

| Dataset | Description |
|---|---|
| Number of measurement | 2,075,259 |
| Data collection range | Dec 2006-Nov 2010 |
| Data missing percentage | 1.25% |
| Data recording frequency | per minute |

TABLE IV
HYPERPARAMETERS

| Parameters | Values | Parameters | Values |
|---|---|---|---|
| Optimizer | Adamax | $\mathcal{K}$ | {100, 1000, 10000} |
| Loss metric | MSE | $n$ | {30, 300, 3000} |
| Hidden layers | 2 | $m$ | {2, 7, 70} |
| Batch size | 32 | $\varepsilon$ | {0.5, 0.7, 1.0} |
| Valid. size | 20% | $\delta$ | 0.001 |
| Activation | ReLU | $\gamma$ | {1, 2, 3} |
| Early stop | Enabled | $\alpha$ | 0.001 |

### C. Convergence Analysis of rDP Process

The convergence of our rDP algorithm is assessed by examining the average changes in the Q-values, denoted as $\Delta Q(s,i)$ for all $\Delta Q = Q^{new}(s_t,i_t) - Q(s_t,i_t)$, where $s \in S$ and $i \in \mathcal{A}$. The objective is to demonstrate that the $Q$-value in the rDP process converges to the optimal Q-value ($Q^*$) as defined by the Bellman equation in a stochastic environment:

$$Q^*(s,i) = r(s,i) + \zeta \max_i \sum_{s_{t+1}} P(s_{t+1}|s,i) Q^*(s_{t+1},i) \quad (15)$$

where $r(s,i)$ is the reward for taking the action $i \in \mathcal{A}$ that yields the highest expected return, and $P(s_{t+1}|s,i)$ is the state transition probability. The expectation $E(Q^{new}(s_t,i_t))$ should converge to the optimal value $Q^*(s,i)$ as defined in (15). For simplicity, we focus on the deterministic case where $Q^{new}(s_t,i_t)$ converges to $Q^*(s,i)$, which is formulated as:

$$Q^*(s,i) = r(s,i) + \zeta \max_i Q^*(s_{t+1},i) \quad (16)$$

If the average of $\Delta Q(s,i)$ approaches zero, the rDP process is considered to be stable and effectively converging.

## VII. Experimental Analysis

This section presents an empirical evaluation of our proposed $\alpha$-MPELM attack model and its implications on FL within an important CI framework: smart grid.

### A. Dataset and Experimental Setup

For our empirical investigation, we utilize the Individual Household Electric Power Consumption dataset [49], a comprehensive smart grid dataset. Essential attributes of this dataset are outlined in Table III. Despite comprising approximately 1.25% missing entries, the dataset's substantial volume ($2,075,259$ records) renders it suitable for a practical demonstration of our model. The $\mathcal{L}$-DPFL environment parameters selected for experimentation are detailed in Table IV.

To experimentally evaluate our proposed attack and defense policy, we use a smart grid dataset (Individual household electric power consumption dataset [49]). Table III enlist some of the important features of the dataset. For $\mathcal{L}$-DPFL environment, we select the parameters as stated in Table IV. Experimental trials are conducted on a high-performance computing setup, featuring a Lambda Tensorbook equipped with an 11th Gen Intel(R) Core(TM) i7-11800H CPU operating at 2.30 GHz, an RTX 3080 Max-Q GPU, 64 GB RAM, and 2 TB of storage. For software and programming environment, we employ Python version 3.9.7 and PyTorch version 1.10.0+cpu.

### B. Adversarial Impact Analysis

Fig. 4 demonstrates the impact of adversarial actions in the absence of an anomaly detector, while Figs. 5-7 detail the impacts under an anomaly detector. Fig. 4(a) indicate an increase in validation loss ($\mathcal{L}$) with enhanced privacy levels, i.e., reduced privacy loss $\varepsilon$, even in the absence of an attacker. This observation reinforces the theoretical understanding that adding more DP noise to enhance privacy leads to an increase in loss. The presence of an adversarial agent further escalates this effect, as shown in Fig. 4(b). Here, a global model with a single compromised local model (i.e., $m = 1$) exhibits a higher loss compared to a model with no compromised nodes ($m = 0$), given the same number of clients and $\varepsilon$-level. This empirical data substantiates the hypothesis that system performance deteriorates, marked by elevated loss when even one malicious entity contributes adversarial DP noise.

*1) Impact of the Degree of Poisoning:* Fig. 4(c) and 4(d) demonstrate that a higher degree of poisoning ($\gamma$) increases the validation loss ($\mathcal{L}$), making the model sub-optimal. A lower $\gamma$ (e.g., $\gamma = 2$) results in minimal impact, akin to a non-adversarial scenario ($m = 0$), while higher $\gamma$ values noticeably

escalate the loss. However, excessively high $\gamma$ values render the malicious models easily detectable by anomaly detectors.

To balance the attack impact and detection risk, the attacker fine-tunes $\gamma$ to obtain $\gamma^{(t)}$ for each FL episode using the $\alpha$−MPELM process from section V-B. This adaptive attack approach is tested against state-of-the-art anomaly detection methods [26], [7]. We focus on recent detection techniques (i.e., Norm, Accuracy, and Mix (Norm+Accuracy) detection from [7]) for their operational similarity to earlier methods (i.e., ERR, LFR, and Mix (ERR+LFR) detection in [26]), aiming to validate the effectiveness of our attack model. We find that the proposed $\alpha$−MPELM attack outperforms a conventional random malicious device (RMD) attack [7] in terms of *detection accuracy* and *validation loss*, particularly when evaluated against Norm, Accuracy, and Mix (Norm+Accuracy) anomaly detection algorithms, as illustrated in Fig. 5, Fig. 6, and Fig. 7. In the evaluation, $\alpha$−MPELM demonstrates a higher success rate in evading detection and degrading system performance over RMD attacks across these detection algorithms. In this context, the greater the decrease in detection accuracy due to a particular attack technique, the higher its efficiency.

**Deceiving Norm detection.** Norm detection, as defined by [7], involves the aggregator calculating a comparison standard for each local update. This standard is derived by averaging all local model updates, excluding the update in question. For a specific noisy clipped local model update $\widetilde{\Delta w}^{(t+1)}{}_{i\chi}$, its comparison standard ($\Delta w^{(t)} i_{st}$) is computed as

$$\Delta w_{i_{st}}^{(t)} = \frac{1}{n-1}\Big(\sum_{k=1}^{n} \widetilde{\Delta w}_{k\chi}^{(t)} - \widetilde{\Delta w}_{i\chi}^{(t)}\Big). \tag{17}$$

The process then involves calculating the squared $L_2$ distance $d_i^{(t)}$ and determining a reference value $e_1$ as follows:

$$d_i^{(t)} = \|\widetilde{\Delta w}_{i\chi}^{(t)} - \Delta w_{i_{st}}^{(t)}\|^2 \tag{18}$$

$$e_1 = \begin{cases} \frac{d_i^{(t)}}{\|\Delta w_{i_{st}}^{(t)}\|^2}, & \text{if } d_i^{(t)} < d_{max}\|\Delta w_{i_{st}}^{(t)}\|^2 \\ d_{max}, & \text{if } d_i^{(t)} \geq d_{max}\|\Delta w_{i_{st}}^{(t)}\|^2, \end{cases} \tag{19}$$

where $d_{max}$ is the maximum squared $L_2$ distance. The Norm detection accuracy for a model update $\widetilde{\Delta w}_{i\chi}^{(t)}$ is:

$$rate_i^{norm} = 1 - max(0, e_1 - \beta_1), \tag{20}$$

where $\beta_1$ is a predefined Norm detection benchmark. A model update is classified as non-anomalous if $rate_i^{norm} = 1$, and anomalous otherwise. Detected anomalous models are excluded from the aggregation into the global model.

In their evaluation of Norm detection, [7] utilized RMD attack, where malicious participants submit randomly generated local model parameters. However, they did not define the parameter boundary for these RMD updates. If this boundary is very large, detection becomes more straightforward, potentially leading to a high average Norm detection accuracy $D_{acc}^n$. Conversely, a narrower parameter boundary may result in a lower $D_{acc}^n$. To realize a practical RMD attack, we set this boundary equal to the clipping threshold, $\mathcal{C}$ in our experiment.

In our experiments shown in Fig. 5, the Norm detection accuracy ($D_{acc}^n$) decreases under our proposed attack with a privacy loss of $\varepsilon = 0.7$. For example, with 30% malicious participants ($m = 0.3(n)$) and a Norm detection benchmark $\beta_1 = 1$, $D_{acc}^n$ drops to 85.4% in our attack, compared to 92.2% in an RMD attack, thereby achieving an attack accuracy improvement of ($92.2\% - 85.4\% = 6.8\%$) (Fig. 5(a)). This pattern is consistent for lower malicious participant ratios ($m = 0.1(n)$ and $m = 0.2(n)$). When the Norm detection threshold $\beta_1$ is increased to 3, the pattern largely remains, except for $m = 0.3(n)$, where initially high adversarial noise makes our anomalous models easily detectable. However, as adversarial noise growth slows down, the validation losses ($\mathcal{L}$) in our attack surpass those in the RMD attack (Fig. 5(g). Lower $D_{acc}^n$ correlates with higher $\mathcal{L}$. This increase in loss for 10%, 20%, and 30% malicious devices with both $\beta_1 = 1$ and $\beta_1 = 3$ is shown in Fig. 5(b)-(g). In all cases, our proposed attack results in higher $\mathcal{L}$ than both the 'No Attack' and 'RMD' scenarios, signifying the effectiveness of $\alpha$−MPELM attack technique to hamper the DPFL process.

**Deceiving Accuracy detection.** Accuracy detection, similar

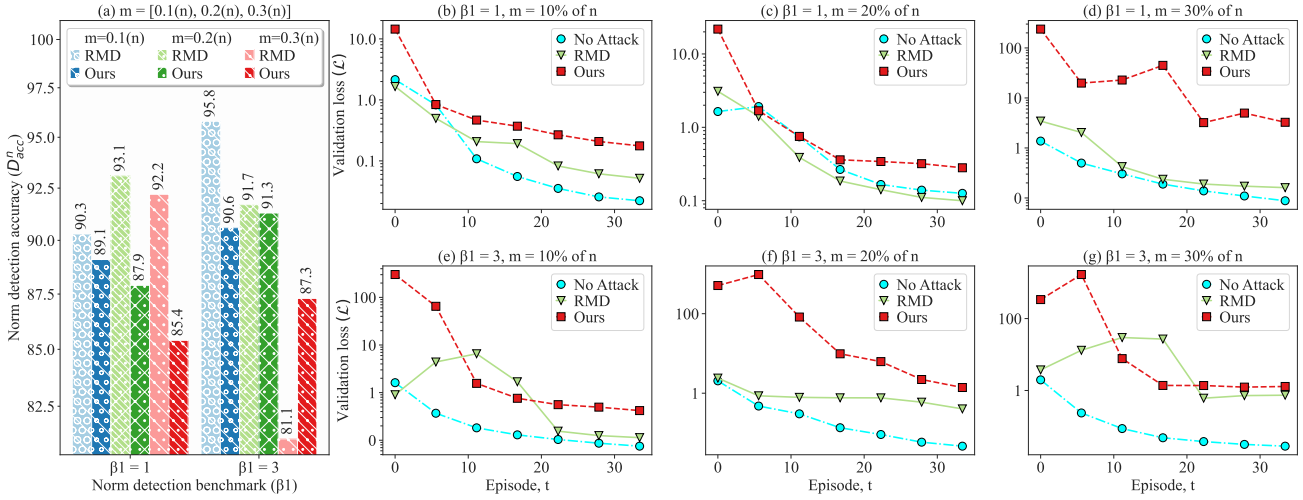


Fig. 5. Deceiving Norm detection: RMD attack vs our attack ($\varepsilon = 0.7$).

This article has been accepted for publication in IEEE Transactions on Artificial Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAI.2025.3540030



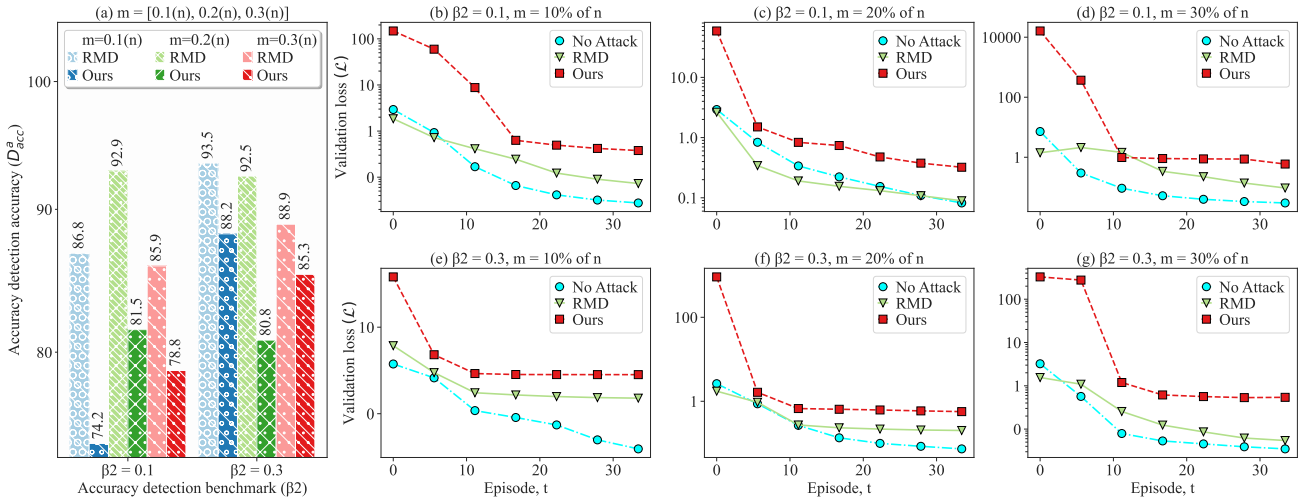


Fig. 6. Deceiving Accuracy detection: RMD attack vs our attack ($\varepsilon = 0.7$).

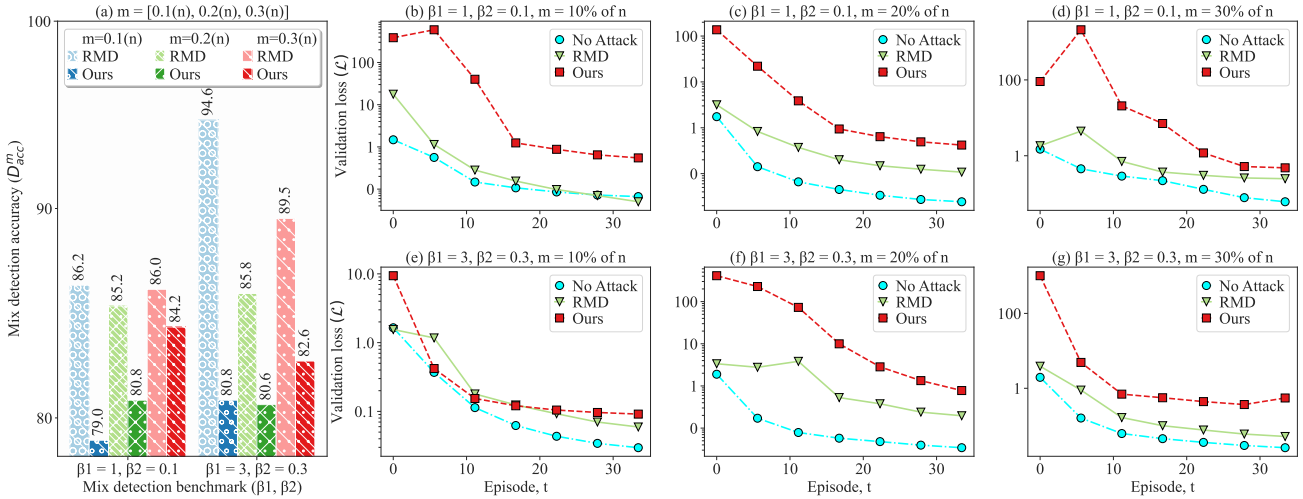


Fig. 7. Deceiving Mix (Norm+Accuracy) detection: RMD attack vs our attack ($\varepsilon = 0.7$).

to Norm detection, computes a standard for comparison using the same formula (17) as defined by [7]. However, instead of calculating the Norm distance, Accuracy detection measures the difference in accuracy using a validation dataset. In this case, two global models are derived from the local model update $\widetilde{\Delta w}^{(t)}_{i\chi}$ and its comparison standard $\Delta w^{(t)}_{i_{st}}$, and their performance is assessed on the validation dataset. Given that our experiment focuses on a regression task, we replace accuracy tests with loss tests (mean squared error). Therefore, if the *loss test* results of $\widetilde{\Delta w}^{(t)}_{i\chi}$ and $\Delta w^{(t)}_{i_{st}}$ are $\mathcal{L}^{(t)}_i$ and $\mathcal{L}^{(t)}_{st}$ respectively, then the loss difference is

$$\Delta\mathcal{L}^{(t)}_i = \begin{cases} 0 & \text{if } \mathcal{L}^{(t)}_{st} \leq \mathcal{L}^{(t)}_i \\ \frac{\mathcal{L}^{(t)}_{st} - \mathcal{L}^{(t)}_i}{\mathcal{L}^{(t)}_{st}} & \text{if } \mathcal{L}^{(t)}_{st} > \mathcal{L}^{(t)}_i \end{cases} \tag{21}$$

The reference value $e_2$ is determined based on the maximum loss difference: $e_2 = max\ (\Delta\mathcal{L}^{(t)})$. Finally, the Accuracy detection rate $rate^{acc}_i$ is computed as

$$rate^{accuracy}_i = \begin{cases} 1 - e_2 & \text{if } e2 > \beta_2 \\ 1 & \text{if } e2 < \beta_2, \end{cases} \tag{22}$$

where $\beta_2$ is a predefined Accuracy detection benchmark. A local model update $\widetilde{\Delta w}^{(t)}_{i\chi}$ is classified as non-anomalous if $rate^{acc}_i = 1$, and anomalous otherwise. To test Accuracy detection, [7] conducted a specialized malicious end device (SMD)-based attack, where a group of malicious participants returns trained local model parameters to modify the sample label of a certain data category. However, our focus on untargeted poisoning in a regression context led us to adopt the RMD attack, setting its boundary to match our clipping threshold for a more stringent comparison. Our results in Fig. 6(a) show that the Accuracy detection algorithm's accuracy ($D^a_{acc}$) decreases more significantly under $\alpha$−MPELM attack (with $\gamma = 0.7$) than the RMD attack. Particularly, we observe a ($86.8\% - 74.2\% = 12.6\%$) attack accuracy improvement for $\alpha$−MPELM attack over RMD attack in deceiving Accuracy detection algorithm (refer to Fig. 6(a), $m = 0.1(n)$ and

$\beta_2 = 0.1$). Concurrently, Fig. 6(b)-(g) illustrates an increase in $\mathcal{L}$ due to a higher rate of misclassification of local models.

**Deceiving Mix (Norm+Accuracy) detection.** In Mix detection, the methodologies of both Norm and Accuracy detections are integrated. This approach entails the aggregator excluding any local models flagged as anomalous by either the Norm or Accuracy detection systems. Fig. 7 presents the efficacy of Mix detection in the context of both our proposed attack model and the RMD attack. The impact of our attack on the Mix detection accuracy ($D^m_{acc}$) mirrors the effects observed in both the standalone Norm and Accuracy detection scenarios. For instance, $\alpha$–MPELM achieves a $(94.6\% - 80.8\% = 13.8\%)$ attack accuracy improvement as compared to RMD attack in deceiving Mix detection algorithm, as illustrated in Fig. 7(a) when $m = 0.1(n)$, $\beta_1 = 3$, and $\beta_2 = 0.3$.

**Changes in the degree of poisoning.** Fig. 8 presents the dynamic adjustments in the degree of poisoning ($\gamma$) across different FL episodes while countering the Norm, Accuracy, and Mix detection methods. As depicted, $\gamma$ typically shows a downward trend after several episodes. This decrease aligns with the adaptive $\alpha$–MPELM process, particularly when the loss ratio $\mathcal{R} >> 1$, leading to $\gamma^{(t)}$ being set to zero, as outlined in Algorithm 2 (lines 11-13). This results in some instances where $\gamma$ remains constant across consecutive episodes. For instance, in Fig. 8(a) (blue curve), there is no change in $\gamma$ between episodes $t = 10$ and $t = 14$, illustrating the adaptability of our poisoning strategy to the learning environment and detection mechanisms. This adjustment of $\gamma$ is key to sustaining the effectiveness of the attack while avoiding detection.

*2) Impact of Attacker-Client Ratio:* The ratio of attackers to clients ($m/n$) in the network plays a crucial role in the impact of an attack. A lower ratio means the global model remains relatively close to its optimal state. Conversely, as this ratio increases, the global model increasingly diverges from its optimal configuration, and in some cases, may even begin to diverge completely. This trend is evident in Fig. 5, Fig. 6, and Fig. 7. In these figures, it is noticeable that the red lines, representing higher $m/n$ ratios, diverge progressively from the cyan lines, which depict scenarios with lower ratios of attackers to clients. This observation highlights the significance of the attacker-client proportion in determining the degree of impact an attack has on the global model.

### C. Performance Analysis of rDP Technique

In mitigating DP-exploited stealthy model poisoning attacks, selecting the appropriate privacy level for nodes during the design phase is crucial. Excessive privacy can deteriorate model performance and enlarge the poisoning window, while insufficient privacy may risk exposing sensitive operational data. We conducted several experiments with varying RL parameters to illustrate these effects, with the findings detailed in Table V. The results indicate that for most discount factors $\zeta$ (except $\zeta = 1.00$), the change in Q-value $\Delta Q(s, a)$ approaches zero as learning concludes, regardless of the learning rate $\alpha$. However, the average global reward, $R$, significantly fluctuates with changes in $\zeta$. Optimal rewards are observed at a learning rate $\alpha = 0.001$ and a discount factor $\zeta = 0.50$, suggesting a balanced consideration of both immediate and future rewards.

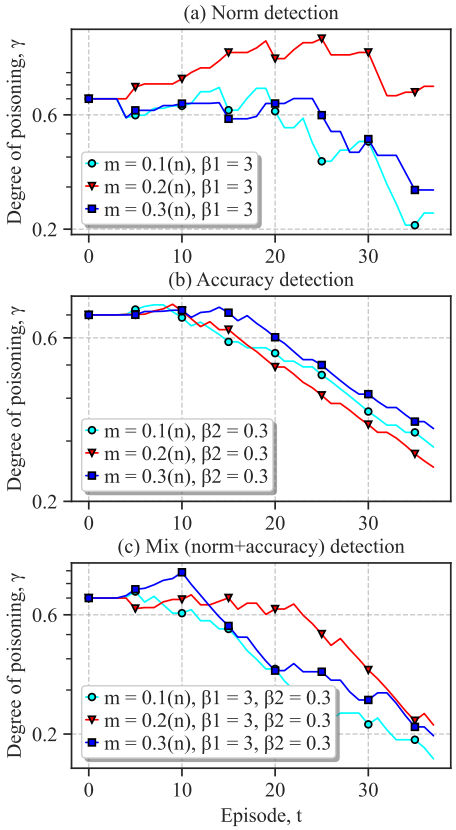


Fig. 8. Changes of $\gamma$ while deceiving the Norm, Accuracy, and Mix detection.

TABLE V
PERFORMANCE ANALYSIS OF THE rDP TECHNIQUE

| Learning rate, $\alpha$ | Discount factor, $\zeta$ | rDP values | |
|---|---|---|---|
| | | Reward, $R$ | Delta, $\Delta Q(s,i)$ |
| $\alpha = 0.01$ | $\zeta = 1.00$ | 8475.33 | $4.13e-00$ |
| | $\zeta = 0.50$ | 13019.58 | $2.86e-07$ |
| | $\zeta = 0.20$ | 11726.13 | $4.25e-08$ |
| | $\zeta = 0.15$ | 10368.59 | $3.32e-08$ |
| $\alpha = 0.001$ | $\zeta = 1.00$ | 9718.91 | $2.92e-00$ |
| | $\zeta = 0.50$ | 13142.54 | $6.24e-04$ |
| | $\zeta = 0.20$ | 10190.45 | $1.63e-04$ |
| | $\zeta = 0.15$ | 11249.85 | $2.94e-04$ |
| $\alpha = 0.0001$ | $\zeta = 1.00$ | 9246.74 | $2.14e-01$ |
| | $\zeta = 0.50$ | 11803.06 | $1.77e-04$ |
| | $\zeta = 0.20$ | 10974.27 | $9.15e-05$ |
| | $\zeta = 0.15$ | 11031.96 | $1.05e-04$ |

*1) Reward Evaluation:* The average global reward is calculated using the formula $R = \sum_{i=1}^{t} r^i$. Fig. 9(a) shows the accumulated reward with a discount factor of $\zeta = 0.50$ and a learning rate of $\alpha = 0.001$. As the number of episodes increases, the RL agent gradually learns and converges to an optimal policy, achieving a balance between privacy, utility, and security. The results demonstrate that the policy stabilizes around episode 100, maintaining a high reward level (approximately 13142.54) thereafter. This suggests that the agent consistently makes optimal decisions beyond this point.

This article has been accepted for publication in IEEE Transactions on Artificial Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAI.2025.3540030



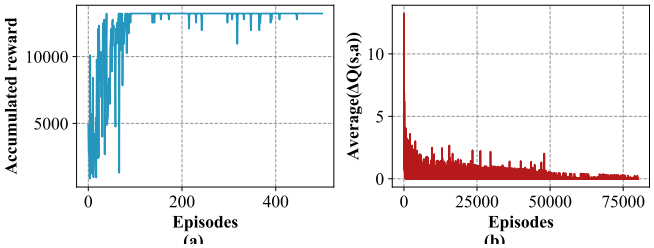


Fig. 9. (a) Accumulated rewards converge after a certain number of episodes (b) Convergence of $\Delta Q$-values.

*2) Q-value Evaluation:* In alignment with the convergence analysis outlined in section VI-C, the stability of our proposed process is indicated by the average of $\Delta Q(s,a)$ approaching zero. The results presented in Fig. 9(b) demonstrate that $\Delta Q(s,a)$ gradually converges to zero after about $60,000$ episodes, suggesting a stable learning process over time.

*3) Assisting Attack Detection:* The reward function $\beta$, as formulated in (14), incorporates variables like the attacker's loss ($m_l$), federated loss ($f_l$), and privacy loss ($\varepsilon$). Given that the RL agent selects an action for each state, a standard federated loss value $f_l^s$ for that state can be determined. When the observed federated loss $f_l^o$ for a particular state diverges notably from $f_l^s$, it can imply one of two scenarios:

- $f_l^s < f_l^o$: This indicates the presence of large-scale attacks, characterized by a high degree of poisoning ($\gamma$).
- $f_l^s \geq f_l^o$: This suggests either an uncompromised system state or a poisoning attack of negligible intensity.

Thus, the rDP algorithm serves a dual purpose: (a) intelligently selects the optimal privacy level during the design phase, thereby reducing the attack surface, and (b) aids in detecting attacks. Importantly, the privacy level selection via the rDP technique can be performed offline through experiments in the design phase of the $\mathcal{L}$-DPFL process, enhancing its utility in safeguarding CI operations against potential threats.

*4) Comparative analysis with Byzantine-robust Aggregation Technique- Krum [27]:* Existing defense techniques (e.g., Krum, Trimmed Mean, Median) operate during the training or learning phase by filtering outliers in local model updates before aggregation. In contrast, the rDP mechanism operates at the design phase of the FL process. Thus, rDP and aggregation techniques like Krum, Trimmed Mean or Median are not direct substitutes but complementary strategies addressing different aspects of the problem. To address this, we conducted experiments integrating Krum with rDP, demonstrating how rDP complements existing Byzantine-robust aggregation methods. Particularly, we compare the losses for "No Attack" and "Attack ($\alpha-$MPELM)" scenarios in both "With Krum" and "Without Krum" FL environments, as shown in Fig. 10(a) and (b). As can be perceived, the loss gap between attack and non-attack scenarios narrows with Krum [27] due to their outlier mitigation mechanisms. However, the absolute loss increases across all scenarios due to the rejection of helpful noisy updates. At lower privacy budgets (high noise levels), this effect is amplified— highlighting the trade-off between robustness and DP-induced performance degradation.

We then apply rDP in both "With Krum" and "Without Krum" scenarios, as illustrated in Fig. 10(c) and Fig. 10(d). When rDP is applied, it successfully identifies the optimal privacy level ($\varepsilon^*$) based on the designer's requirements. Particularly, for the scenario "[Without Krum]+rDP" (refer to Fig. 10(c)), the trained rDP policy reinforces the conclusion that $\varepsilon \approx 0.9776$ and $\varepsilon \approx 1.3053$ are two optimal choices, where $\varepsilon \approx 0.9776$ offers high privacy and $\varepsilon \approx 1.3053$ achieves minimal "No Attack" and "Attack ($\alpha-$MPELM)" loss. This demonstrates that rDP can mitigate the impact of $\alpha-$MPELM even without relying on additional robust aggregation techniques. A similar trend can also be observed for the scenario "[With Krum]+rDP" in Fig. 10(d).

In summary, without rDP, the designer cannot effectively balance privacy and utility because there is no clear guidance on which $\varepsilon$ level to choose. By introducing rDP, we systematically analyze historical losses across different $\varepsilon$ levels.

### D. Limitations and Future Recommendations

A potential challenge in the DPFL process is the introduction of randomized noise for each client in every episode, which can cause significant fluctuations in the learning process compared to a non-DP environment. This may affect the stability and convergence of the real-world FL systems. One potential solution could be to limit model updates within a certain threshold, but excessive clipping might compromise DP's privacy protections. Therefore, further research is required to explore methods to balance model update clipping without undermining privacy to minimize these fluctuations.

Another potential limitation of this study is the timing of the attacks. We initiated attacks on compromised models from the outset of the learning process for simplicity. However, real-world attacks could occur at various stages, potentially adhering to similar adversarial principles. Future research should investigate the impact of attacks initiated at different stages of the learning process and develop strategies to detect and mitigate these late-stage attacks.

The field of DPFL is rapidly evolving with numerous advanced algorithms and future developments are anticipated. Attackers might exploit new methods that employ Gaussian noise for privacy in FL models. Consequently, defense strategies must be continually updated and adapted to counteract emerging threats as new DPFL techniques are developed.

## VIII. Conclusion

In this paper, we investigate model poisoning attacks in the context of Federated Learning (FL) and Differential Privacy (DP). We uncover that Gaussian noise, added for DP, can be exploited by attackers to conduct stealthy, persistent model poisoning in FL settings. Our proposed $\mathcal{L}$-DPFL attack effectively reduces the accuracy of current detection methods. In response, we introduce a novel defense strategy, rDP, demonstrating its effectiveness in achieving optimal policy convergence. This study is an effort to address model poisoning threats in DPFL-driven CIs, potentially catalyzing further research in adversarial FL. Future work aims to explore targeted model poisoning attacks using non-IID data.

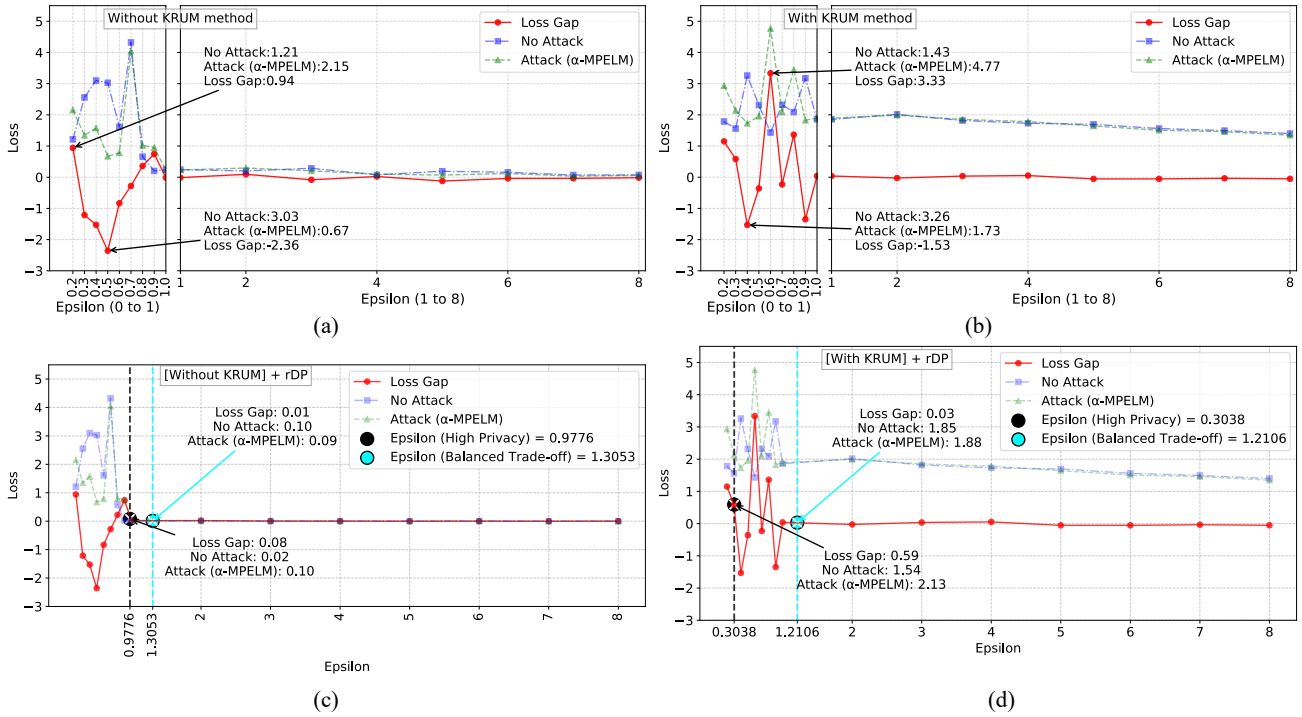


Fig. 10. Comparative analysis of rDP over Krum [27] (Attacker-Client Ratio $(m/n)$ = 6.67%) . (a) No Attack v. Attack ($\alpha-$MPELM) without Krum, (b) No Attack v. Attack ($\alpha-$MPELM) with Krum, (c) Optimal privacy loss ($\varepsilon$) level for FL without Krum, and (d) Optimal privacy loss ($\varepsilon$) level for FL with Krum.

## REFERENCES


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

[2] CISA, "Critical infrastructure sectors," Oct 2020. [Online]. Available: https://www.cisa.gov/critical-infrastructure-sectors

[3] H. Zhu, R. Wang, Y. Jin, and K. Liang, "Pivodl: Privacy-preserving vertical federated learning over distributed labels," *IEEE Transactions on Artificial Intelligence*, vol. 4, no. 5, pp. 988–1001, 2023.

[4] A. Taïk and S. Cherkaoui, "Electrical load forecasting using edge computing and federated learning," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–6.

[5] Z. Su, Y. Wang, T. H. Luan, N. Zhang, F. Li, T. Chen, and H. Cao, "Secure and efficient federated learning for smart grid with edge-cloud collaboration," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 2, pp. 1333–1344, 2021.

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

[7] J. Zhou, N. Wu, Y. Wang, S. Gu, Z. Cao, X. Dong, and K.-K. R. Choo, "A differentially private federated learning model against poisoning attacks in edge computing," *IEEE Transactions on Dependable and Secure Computing*, 2022.

[8] D. U. Case, "Analysis of the cyber attack on the ukrainian power grid," *Electricity Information Sharing and Analysis Center (E-ISAC)*, vol. 388, 2016.

[9] L. Chen, D. Zhao, L. Tao, K. Wang, S. Qiao, X. Zeng, and C. W. Tan, "A credible and fair federated learning framework based on blockchain," *IEEE Transactions on Artificial Intelligence*, pp. 1–15, 2024.

[10] Y. Li, Y. Zhou, A. Jolfaei, D. Yu, G. Xu, and X. Zheng, "Privacy-preserving federated learning framework based on chained secure multi-party computing," *IEEE Internet of Things Journal*, 2020.

[11] Y. Zheng, S. Lai, Y. Liu, X. Yuan, X. Yi, and C. Wang, "Aggregation service for federated learning: An efficient, secure, and more resilient realization," *IEEE Transactions on Dependable and Secure Computing*, 2022.

[12] Q. Feng, D. He, J. Shen, M. Luo, and K.-K. R. Choo, "Ppnnt: Multi-party privacy-preserving neural network training system," *IEEE Transactions on Artificial Intelligence*, 2023.

[13] Z. Lu, S. Lu, X. Tang, and J. Wu, "Robust and verifiable privacy federated learning," *IEEE Transactions on Artificial Intelligence*, pp. 1–14, 2023.

[14] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of cryptography conference*. Springer, 2006, pp. 265–284.

[15] J. Giraldo, A. Cardenas, M. Kantarcioglu, and J. Katz, "Adversarial classification under differential privacy," in *Network and Distributed Systems Security (NDSS) Symposium 2020*, 2020.

[16] M. T. Hossain, S. Badsha, and H. Shen, "Privacy, security, and utility analysis of differentially private cpes data," in *2021 IEEE Conference on Communications and Network Security (CNS)*. IEEE, 2021, pp. 65–73.

[17] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 308–318.

[18] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: A client level perspective," *arXiv preprint arXiv:1712.07557*, 2017.

[19] Y. Zhao, J. Zhao, M. Yang, T. Wang, N. Wang, L. Lyu, D. Niyato, and K.-Y. Lam, "Local differential privacy-based federated learning for internet of things," *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 8836–8853, 2020.

[20] R. Hu, Y. Guo, H. Li, Q. Pei, and Y. Gong, "Personalized federated learning with differential privacy," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9530–9539, 2020.

[21] G. Li, J. Wu, S. Li, W. Yang, and C. Li, "Multi-tentacle federated learning over software-defined industrial internet of things against adaptive poisoning attacks," *IEEE Transactions on Industrial Informatics*, 2022.

[22] L. Sun, J. Qian, and X. Chen, "Ldp-fl: Practical private aggregation in federated learning with local differential privacy," *arXiv preprint arXiv:2007.15789*, 2020.

[23] S. Truex, L. Liu, K.-H. Chow, M. E. Gursoy, and W. Wei, "Ldp-fed: Federated learning with local differential privacy," in *Proceedings of the Third ACM International Workshop on Edge Systems, Analytics and Networking*, 2020, pp. 61–66.

[24] J. Giraldo, A. A. Cardenas, and M. Kantarcioglu, "Security vs. privacy: How integrity attacks can be masked by the noise of differential privacy," in *2017 American Control Conference (ACC)*. IEEE, 2017, pp. 1679–1684.

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

[26] M. Fang, X. Cao, J. Jia, and N. Gong, "Local model poisoning attacks to byzantine-robust federated learning," in *29th {USENIX} Security Symposium ({USENIX} Security 20)*, 2020, pp. 1605–1622.

[27] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 118–128.

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

[29] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5650–5659.

[30] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[31] M. Hao, H. Li, X. Luo, G. Xu, H. Yang, and S. Liu, "Efficient and privacy-enhanced federated learning for industrial artificial intelligence," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 10, pp. 6532–6542, 2019.

[32] T. Huang, W. Lin, L. Shen, K. Li, and A. Y. Zomaya, "Stochastic client selection for federated learning with volatile clients," *IEEE Internet of Things Journal*, 2022.

[33] E. Bressert, "Scipy and numpy: an overview for developers," 2012.

[34] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. Quek, and H. V. Poor, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3454–3469, 2020.

[35] M. T. Hossain, H. La, and S. Badsha, "Rampart: Reinforcing autonomous multi-agent protection through adversarial resistance in transportation," *ACM J. Auton. Transport. Syst.*, jan 2024, just Accepted. [Online]. Available: https://doi.org/10.1145/3643137

[36] P. C. M. Arachchige, P. Bertok, I. Khalil, D. Liu, S. Camtepe, and M. Atiquzzaman, "Local differential privacy for deep learning," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5827–5842, 2019.

[37] A. El Ouadrhiri and A. Abdelhadi, "Differential privacy for deep and federated learning: A survey," *IEEE Access*, vol. 10, pp. 22 359–22 380, 2022.

[38] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, "Local privacy and statistical minimax rates," in *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*. IEEE, 2013, pp. 429–438.

[39] J. Gao, B. Hou, X. Guo, Z. Liu, Y. Zhang, K. Chen, and J. Li, "Secure aggregation is insecure: Category inference attack on federated learning," *IEEE Transactions on Dependable and Secure Computing*, 2021.

[40] S. Awan, B. Luo, and F. Li, "Contra: Defending against poisoning attacks in federated learning," in *Computer Security–ESORICS 2021: 26th European Symposium on Research in Computer Security, Darmstadt, Germany, October 4–8, 2021, Proceedings, Part I 26*. Springer, 2021, pp. 455–475.

[41] M. Barreno, B. Nelson, A. D. Joseph, and J. D. Tygar, "The security of machine learning," *Machine Learning*, vol. 81, no. 2, pp. 121–148, 2010.

[42] M. Jagielski, A. Oprea, B. Biggio, C. Liu, C. Nita-Rotaru, and B. Li, "Manipulating machine learning: Poisoning attacks and countermeasures for regression learning," in *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2018, pp. 19–35.

[43] X. Li, Z. Qu, S. Zhao, B. Tang, Z. Lu, and Y. Liu, "Lomar: A local defense against poisoning attack on federated learning," *IEEE Transactions on Dependable and Secure Computing*, 2021.

[44] M. A. Smart, D. Sood, and K. Vaccaro, "Understanding risks of privacy theater with differential privacy," *Proceedings of the ACM on Human-Computer Interaction*, vol. 6, no. CSCW2, pp. 1–24, 2022.

[45] C. Dwork, N. Kohli, and D. Mulligan, "Differential privacy in practice: Expose your epsilons!" *Journal of Privacy and Confidentiality*, vol. 9, no. 2, 2019.

[46] P. Nanayakkara, M. A. Smart, R. Cummings, G. Kaptchuk, and E. M. Redmiles, "What are the chances? explaining the epsilon parameter in differential privacy," in *32nd USENIX Security Symposium (USENIX Security 23)*, 2023, pp. 1613–1630.

[47] J. Near and D. Darais, "Differential privacy: Future work & open challenges," January 2022, last Accessed: 28 December, 2024. [Online]. Available: https://www.nist.gov/blogs/cybersecurity-insights/differential-privacy-future-work-open-challenges

[48] M. Wunder, M. L. Littman, and M. Babes, "Classes of multiagent q-learning dynamics with epsilon-greedy exploration," in *ICML*, 2010.

[49] G. Hebrail and A. Berard, "Individual household electric power consumption data set," *É. d. France, Ed., ed: UCI Machine Learning Repository*, 2012.