FLIFRA: Hybrid Data Poisoning Attack Detection in Federated Learning for IoT Security

Mulualem Bitew Anley*, Angelo Genovese*, Tibebe Beshah Tesema†, and Vincenzo Piuri*
* Department of Computer Science, Università degli Studi di Milano, Italy. *firstname.lastname@unimi.it*
† School of Information Science, Addis Ababa University, Ethiopia. *tibebe.beshah@aau.edu.et*

***Abstract*— The rapid expansion of IoT devices has transformed numerous industries by enabling extensive data collection and real-time analytics. Federated Learning (FL) offers a decentralized model training paradigm that ensures data privacy, making it particularly suitable for IoT environments. Yet, it remains vulnerable to poisoning attacks that can severely compromise model integrity, wherein malicious clients compromise the global model by injecting poisoned updates. Existing defenses, which focus primarily on global model performance, often fail to effectively integrate local anomaly detection with global weighting mechanisms, thus limiting their efficacy against such threats. Addressing this research gap, we propose FLIFRA (Federated Learning Isolation Forest with Robust Aggregation), a hybrid defense framework that combines client-side anomaly detection using Isolation Forest (iForest) with dynamic reputation-based robust aggregation at the server. This dual-layer approach filters out malicious updates before aggregation and adjusts client reputations to mitigate adversarial influence. Our evaluation of three cybersecurity datasets (CIC-IDS2018, BoT-IoT, and UNSW-NB15) under various intensities of poisoning (10%, 20%, 30%, and 40%) demonstrates that the proposed method outperforms the traditional aggregation schemes of FedAvg, Krum, Trimmed Mean, DRRA, and Wei-Detect in the literature. In particular, our framework achieves higher detection accuracy, faster convergence, and improved stability, even in highly heterogeneous data environments.**

I. INTRODUCTION

The exponential growth of IoT ecosystems has significantly transformed the digital landscape, creating interconnected devices that facilitate seamless data exchange and automation. These devices generate a large amount of data, essential for improving services and enabling predictive capabilities. However, the sensitivity and distributed nature of the data have also made IoT environments attractive targets for sophisticated cyberattacks. Ensuring the security of these decentralized networks is paramount, particularly against malicious activities that threaten their reliability and functionality. Thus, Federated Learning (FL) has emerged as a promising paradigm for securing IoT ecosystems, enabling decentralized model training across IoT devices while ensuring data privacy and reducing the communication burden by avoiding the transfer of raw data [1].

This work was supported in part by the EC under projects Chips JU EdgeAI (101097300) and GLACIATION (101070141) and by project SERICS (PE00000014) under the MUR NRRP funded by the EU - NGEU. We also thank the NVIDIA Corporation for the GPU donated. Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the Italian MUR. Neither the European Union nor Italian MUR can be held responsible for them.

With its decentralized nature and reliance on clients, FL introduces critical vulnerabilities for poisoning attacks, aiming to compromise the performance of the global model. These attacks are particularly dangerous in IoT settings, where diverse, non-IID data distributions exacerbate their impact. For instance, backdoor attacks embed triggers in input data to induce targeted misclassifications, allowing adversaries to manipulate predictions under specific conditions [2], thus inducing IoT-based intrusion detection systems to misclassify malicious traffic as benign. Moreover, label-flipping attacks can corrupt the training process by altering class labels, confusing the model, and degrading overall accuracy [3]. Another attack example is represented by gradient manipulation, which directly alters the shared gradients during training and corrupts the global model by injecting subtle but harmful updates during aggregation [4].

Existing approaches to mitigate data poisoning attacks in FL focus on client-side detection [5], [6] or server-side detection [7]–[10], rather than adopting a holistic strategy. This division reduces overall effectiveness: client-side methods may fail to detect sophisticated attacks, while server-side techniques lack granular insights into local data distributions [11]. These limitations underscore the need for a robust, scalable, and adaptive solution tailored to the unique challenges of IoT ecosystems [12]. To address these limitations, we introduce a novel dual-layer framework that integrates local anomaly detection iForest with a global dynamic reputation-based robust aggregate (DRRA) mechanism: at the client level, iForest rigorously examines local training data and model behavior to identify anomalous updates effectively, while at the global level, the DRRA mechanism dynamically adjusts client contributions based on historical performance.

This paper makes three primary contributions: *i)* we define the problem of defense against data poisoning attacks in FL, focusing on IoT scenarios marked by non-IID data and resource constraints; *ii)* we introduce Federated Learning Isolation Forest with Robust Aggregation (FLIFRA)[1], a scalable and adaptive dual-layered framework that synergistically integrates iForest for local anomaly detection with a DRRA mechanism to globally enhance resilience against adversarial attacks; *iii)* we conduct a rigorous evaluation of our approach under varying intensities of poisoning attacks using real-world IoT datasets.

The remainder of the paper is organized as follows.

[1] Source code available at https://github.com/mulerkal/FLIFRA

Section II reviews the literature on FL and data poisoning, focusing on current mitigation strategies. Section III details our proposed methods for detecting data poisoning in FL systems. Section IV presents experimental setups, while Section V describes results and discussions of the proposed strategies. Finally, Section VI concludes the paper and outlines the directions for future research.

II. Literature Review

A. Data Poisoning Attacks In FL

The architecture of FL in IoT-Edge environments spans multiple domains of IoT applications, including smart transport, healthcare, industrial IoT (IIoT) and smart cities. In this FL setting, IoT-Edge can include smart sensors, wearable devices, as well as industrial machinery that collects and processes local data to train ML models. Each device operates independently within its domain, maintaining data privacy by keeping raw data locally and only sharing model update gradients or parameters with a central server. The central server aggregates these local model updates to form a global model, which is then redistributed to all IoT-Edge devices for further local training. However, this decentralized approach exposes the FL system to data poisoning attacks [11].

In data poisoning attacks, one or more IoT-Edge devices are compromised, allowing an adversary to inject poisoned data into local training. For instance, in a smart home scenario, a compromised device trains its local model on malicious data and sends tainted updates to the central server. When these updates are aggregated with benign ones, they degrade the global model's performance and introduce bias. This corrupted global model is then redistributed to all devices, propagating the attack [13].

B. Poisoning Attack Detection in FL

FL is increasingly vulnerable to data poisoning attacks, where malicious clients inject corrupted data to degrade the global model's performance, posing significant threats to model integrity, particularly in IoT ecosystems, where the resource constraints and dynamic conditions inherent to such environments exacerbate the challenges of detecting and mitigating attacks [9].

Existing detection strategies can be broadly categorized into client-side and server-side approaches [14]. Client-side techniques include iForests [5], which focuses on identifying outliers by analyzing local training data and algorithms. However, these methods struggle to detect coordinated attacks that mimic benign behavior, limiting their effectiveness in real-world scenarios. To address this issue, the approach described in [6] proposes FL-WBC, which identifies the attack parameter space and perturbs it during local training to mitigate long-lasting attack effects. Although promising, such client-side method often lacks scalability and adaptability to dynamic environments.

On the server side, various techniques were proposed to reduce the influence of malicious updates by filtering or aggregating model updates from clients. Provenance-based methods [9], for instance, trace the origin of model updates to identify malicious contributions, but rely heavily on accurate provenance data, which is often impractical in resource-constrained IoT environments. Similarly, FreqFed [15] leverages frequency-domain analysis to detect malicious updates but assumes stationary data distributions, which restricts its effectiveness in dynamic IoT environments.

Other server-side approaches, which include client filtering [8] and clustering-based methods [16], aggregate updates only from trustworthy clients. For example, the FLAME-based approach introduced in [17] leverages model clustering and weight clipping to neutralize adversarial client contributions. Robust aggregation methods, including Krum [10], Trimmed Mean, and median aggregation [7], have also been widely adopted to mitigate poisoning attacks by reducing the influence of outliers. These methods average the least likely tainted updates, providing resilience against a small number of Byzantine (adversarial) clients. However, their effectiveness reduces as the number of compromised clients increases. Furthermore, the method described in [18] introduces WeiDetect, an approach that uses a trusted validation set to filter out malicious or low-quality updates before aggregation. Differently than the previous approaches, FedLSA [19] employs autoencoders to detect poisoned updates in the latent space, offering a novel approach to identifying malicious contributions. However, its performance degrades in non-IID data scenarios, limiting its applicability.

Despite significant progress, challenges persist in detecting and mitigating data poisoning attacks in FL. Existing methods often focus exclusively on either client-side detection or server-side detection, resulting in incomplete defenses. Client-side approaches may fail to detect sophisticated attacks [6], while server-side methods lack granular insights into local data distributions [11].

III. Proposed Methodology

This subsection details the client-side iForest anomaly-based detection, the server-side Dynamic Reputation-based Robust Aggregation (DRRA), and the hybrid Federated Learning Isolation Forest with Robust Aggregation (FLIFRA) approaches, along with their respective algorithms (Figure 1).

A. Client-Side: iForest Anomaly-based Detection

The first layer of the proposed FLIFRA hybrid detection approach employs the iForest algorithm to identify anomalies in model updates. The detailed steps of the algorithm are described below and outlined in Algorithm 1.

- *Step 1: Local application of iForest.* After initializing iForest $\mathcal{M}$ with n trees and contamination level η, each i-th IoT device trains the model locally on the local dataset D_i. Then, the device applies the iForest algorithm to its model update. The algorithm randomly selects a feature and a split value to partition the data, constructing trees where outliers are isolated near the root due to their uniqueness. This process is repeated across multiple trees in the ensemble.

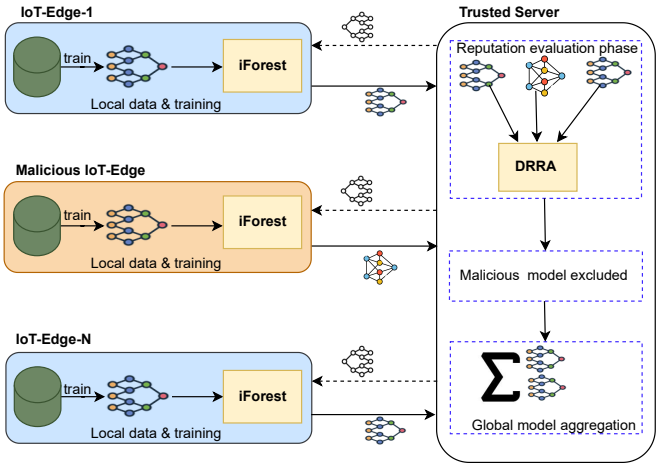


Fig. 1: Architecture of an FL-based data poisoning attack in IoT-Edge environments.

- *Step 2: Anomaly score calculation.* For each model update, iForest calculates an anomaly score s based on the average path length required to isolate the point in all trees in the forest. The path length is defined as the number of edges traversed from the root node to the terminating node. The score is calculated as:

$$s(x) = 2^{-\frac{E(h(x))}{c}}, \tag{1}$$

where $E(h(x))$ is the expected path length of a point x in the iForest, n is the number of samples, and $c(n)$ is the average path length of an unsuccessful search in a Binary Search Tree (BST).

- *Step 3: Flagging potentially malicious updates.* If a model update's anomaly score exceeds a certain threshold, it is considered an outlier and is flagged as potentially malicious; otherwise, it is considered normal. We employ a dynamic thresholding mechanism to enhance the detection of malicious activities. The threshold θ is defined as:

$$\theta = Q_{1-\eta}(\mathcal{S}) \;;$$
$$\mathcal{S} = \{s(x_j)\}_{j=1}^{|D_i|}, \tag{2}$$

where Q is the $(1-\eta)$-quantile of $\mathcal{S}$, with η indicating the contamination level and $\mathcal{S}$ the set of anomaly scores. The threshold θ is continuously updated based on the distribution of anomaly scores over time, utilizing a rolling-window approach. The filtered dataset D_i' is constructed as the set of normal samples.

- *Step 4: Filtered dataset and anomaly report.* The filtered dataset D_i' is used to train the models at subsequent iterations. The set A_i of flagged updates, along with their anomaly scores, are then sent to the central server.

B. Server Side: Dynamic Reputation-Based Robust Aggregation (DRRA)

Performed at the server side, DRRA dynamically adjusts the weight of updates for each client during the aggregation process based on the evolving reputation scores from the training rounds. The algorithm assigns higher weights to reliable clients and penalizes those exhibiting malicious behavior. The details are provided in the following and illustrated in Algorithm 2.

Algorithm 1: iForest Anomaly Detection

Input: Local dataset D_i, contamination level η, number of estimators n.
Output: Filtered dataset D_i'.

Initialization.
  Initialize iForest model $\mathcal{M}$ with n trees and contamination η;
Step 1: Local application of iForest.
  Apply $\mathcal{M}$ on D_i;
Step 2: Anomaly score computation.
  foreach *data point* $x_j \in D_i$ **do**
    $s(x_j) \leftarrow \mathcal{M}.score(x_j)$;
Step 3: Flagging potentially malicious updates.
  Let $\mathcal{S} = \{s(x_j)\}_{j=1}^{|D_i|}$;
  $\theta = Q_{1-\eta}(\mathcal{S})$;
  $D_i' \leftarrow \emptyset$;
  $A_i \leftarrow \emptyset$;
  foreach *data point* $x_j \in D_i$ **do**
    if $s(x_j) > \theta$ **then**
      Label x_j as *anomaly* and append $(x_j, s(x_j))$ to A_i;
    else
      Label x_j as *normal* and append x_j to D_i';
Step 4: Filtered dataset and anomaly report.
  return D_i, A_i

- *Step 1: Reputation score initialization.* At the beginning of the training process, all clients are assigned equal reputation scores $R_i(0)$, where $R_i(0) = 1$ for each client i. This uniform initialization assumes no prior knowledge about client behavior and treats all clients equally trustworthy. As training progresses, the reputation scores are dynamically updated based on the consistency and reliability of the client's contributions.

- *Step 2: Reputation score update.* Each client i updates its reputation score $R_i(t+1) \in [0,1]$ at training round $t+1$, according to:

$$R_i(t+1) = \alpha\, R_i(t) \;+\; (1-\alpha)\, S_i(t), \quad \alpha \in [0,1] \tag{3}$$

where α is a decay factor controlling the historical behavior's influence and $S_i(t)$ is a performance score computed based on the similarity of the client's update to the median of all client updates at round t. We compute $S_i(t)$ by measuring the Euclidean distance between the client's update $U_i(t)$ and the central model update $U_m(t)$, defined as the component-wise median of all client updates:

$$S_i(t) = 1 - \frac{\text{distance}(U_i(t), U_m(t))}{\max_j(\text{distance}(U_j(t), U_m(t)))}, \tag{4}$$

Algorithm 2: Dynamic Reputation–Based Robust Aggregation (DRRA)

Input: Client updates $U_1, \ldots, U_N$, reputations $R_1, \ldots, R_N$, decay α, reputation threshold τ_r, small constant ε.
Output: Global update U_{global}, weights $w_1, \ldots, w_N$.

Step 1: Reputation score initialization.
  foreach *Client* $i, i = 1, \ldots, N$ **do**
    $R_i(0) = 0$;
Step 2: Reputation score update.
  $U_m \leftarrow \text{median}(U_1, \ldots, U_N)$;
  foreach $i = 1, \ldots, N$ **do**
    $d_i \leftarrow \|U_i - U_m\|_2$;
  $D_{\max} \leftarrow \max(d_i, \varepsilon)$;
  foreach $i = 1, \ldots, N$ **do**
    $S_i \leftarrow 1 - (d_i / D_{\max})$;
    $R_i \leftarrow \alpha R_i + (1 - \alpha) S_i$;
Step 3: Weighted aggregation of client updates.
  foreach $i = 1, \ldots, N$ **do**
    if $R_i \geq \tau_r$ **then**
      $\tilde{R}_i \leftarrow R_i$;
    else
      $\tilde{R}_i \leftarrow 0$;
    $w_i \leftarrow \tilde{R}_i / \sum_{j=1}^{N} \tilde{R}_j$;
  $U_{\text{global}} \leftarrow \sum_{i=1}^{N} w_i U_i$;
return U_{global}

where: $U_i(t)$ is the update from client i, $U_m(t)$ is the median update, and distance$(\cdot, \cdot)$ is a metric Euclidean distance. Clients with updates closer to the median receive higher scores, while those with significant deviations are penalized.

- *Step 3: Weighted aggregation of client updates.* After computing the updated reputations $R_i(t+1)$ for all i, the server filters the reputation scores by applying the threshold $\tau_r \in [0, 1]$:

$$\tilde{R}_i(t+1) = \begin{cases} R_i(t+1), & R_i(t+1) \geq \tau_r \\ 0, & R_i(t+1) < \tau_r \end{cases} , \quad (5)$$

The aggregation weight for client i is then computed as:

$$w_i(t+1) = \frac{\tilde{R}_i(t+1)}{\sum_{j=1}^{N} \tilde{R}_j(t+1)}, \quad i = 1, \ldots, N, \quad (6)$$

where N is the total number of clients. Finally, the global model is updated as:

$$U_{\text{global}}(t+1) = \sum_{i=1}^{N} w_i(t+1)\, U_i(t) \ . \quad (7)$$

C. FLIFRA: Hybrid Approach

Our proposed approach combines local anomaly detection using an iForest with robust DRRA aggregation on the server. The details are described in Algorithm 3.

Algorithm 3: FLIFRA: Federated Learning Isolation Forest with Robust Aggregation

Input: Client datasets $\{D_i\}_{i=1}^{N}$, contamination η, iForest trees n, reputation decay α, reputation threshold τ_r, rounds T, initial model $M^{(0)}$, server learning rate γ, small constant ε.
Output: Robust global model $M^{(T)}$.

$M^{(0)} \leftarrow$ initial model;
for $t = 1$ **to** T **do**
  `// CLIENT SIDE: iForest`
  foreach $i = 1, \ldots, N$ ***in parallel*** **do**
    Run **Algorithm 1** on D_i with (η, n) to obtain filtered D_i';
    Train local model $M_i^(t)$ on D_i', starting from $M^{(t-1)}$;
    $U_i \leftarrow M_i - M^{(t-1)}$;
    Send U_i to server;
  `// SERVER SIDE: DRRA`
  Run **Algorithm 2** with inputs $\{U_i\}_{i=1}^{N}$, $\{R_i\}_{i=1}^{N}$, $(\alpha, \tau_r, \varepsilon)$ to obtain global update U_{global} and new weights $\{w_i\}$;
  $M^{(t)} \leftarrow M^{(t-1)} + \gamma\, U_{\text{global}}$;
return $M^{(T)}$;

IV. Experiments and Evaluation

A. Threat Models in FL

In the proposed dual-layer approach, the attacker's goal is to compromise the global model by injecting poisoned updates that both degrade its overall accuracy and implant a stealth backdoor, causing benign IoT traffic to be misclassified as malicious or vice-versa. To accomplish this, the adversary fully controls one or more compromised IoT participants, manipulating their local training data and gradients to upload arbitrary model updates. However, the attacker cannot intercept or modify honest clients' contributions and lacks access to the server's reputation validation set.

B. Datasets

CIC-IDS2018. The dataset was generated within a simulated network environment that incorporates realistic traffic from various applications and services, ensuring its relevance to modern network infrastructures [20].[2]

BoT-IoT. The dataset was collected in a controlled environment that simulates a diverse IoT network with smart thermostats, webcams, motion sensors, and connected home appliances. Data includes benign traffic, representing legitimate activities of the IoT network, and malicious traffic generated through various cyberattack scenarios [21].[3]

UNSW-NB15. The dataset is a contemporary network intrusion benchmark developed at the Australian Center for Cyber Security (ACCS).[4] It contains both benign and malicious

[2] https://www.unb.ca/cic/datasets/ids-2018.html
[3] https://research.unsw.edu.au/projects/bot-iot-dataset
[4] https://research.unsw.edu.au/projects/unsw-nb15-dataset

TABLE I: Training and testing samples with class details used in the experiments for the considered datasets.

Dataset	Num. of samples		Classes
	Training	Test	
CIC-IDS2018	1,476,912	369,229	8 (Benign, DoS, DDoS, Brute Force, Botnet, Web Attacks, Infiltration, SQL Injection)
BoT-IoT	422,766	105,692	5 (Benign, Scan, DDoS, Reconnaissance, Theft)
UNSW-NB15	175,341	82,332	9 (Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, Worms)

network traffic across a range of recent attack types. The number of samples and classes used in the experiment is shown in Table I.

C. Data Preprocessing

The preprocessing of the three considered datasets ensures that the data is clean, consistent, and suitable for the proposed approach. In particular, missing and duplicate values were removed and nonessential features were discarded. Categorical features, including protocols and attack types, were converted to numerical values using label encoding for efficient processing. Continuous features, including flow duration and packet length, were normalized using the Min-Max scaling method to standardize values between 0 and 1, thereby aiding model convergence during training. Feature extraction was performed to reduce dimensionality, with key features selected based on information gain and entropy rankings to retain critical attributes without compromising accuracy [22].

D. Dataset Distribution

We use 80% of each dataset for training and reserve the remaining 20% for testing. In our FL framework, we distribute the datasets across multiple participant nodes to approximate realistic non-IID conditions. Each node receives a distinct subset of samples, characterized by varying proportions of benign versus malicious traffic as well as the presence or absence of specific attack types. This allocation strategy mirrors operational environments, where individual organizations or network segments encounter unique threat landscapes. To generate these heterogeneous data distributions, we employ a Dirichlet distribution, a widely recognized method in the FL literature for creating skewed multivariate data partitions [23]. Through this approach, the local data at each client node exhibits statistical properties representative of real-world network scenarios, thereby enhancing both the fidelity and robustness of our empirical evaluations.

E. Model Architecture

For this study, we have designed a CNN architecture specifically tailored to detect data poisoning attacks in FL settings within IoT environments. The proposed model integrates key features to address the unique challenges of IoT networks. It comprises multiple convolutional layers for feature extraction, followed by pooling layers to reduce dimensionality and capture essential patterns. Additionally, dropout and batch normalization techniques are incorporated to enhance generalization and robustness.

F. Hyperparameters

In this paper, CNN-based models were trained with a client-side batch size of 64, utilizing the Adam optimizer and categorical cross-entropy loss for multi-class classification. To address non-IID data heterogeneity, client-specific hyperparameters, including local learning rates (adaptively tuned per client), local epochs, and dropout rates, were optimized to balance performance and generalization. ReLU activation in hidden layers enabled non-linear feature extraction, while SoftMax at the output layer provided probabilistic class assignments. On the server side, a minimum of 100 clients participate per training round to ensure statistical reliability. To combat overfitting, early stopping (based on validation loss divergence) and dropout regularization were integrated.

G. Baselines

To evaluate the performance of our model in detecting data poisoning attacks, we compared it with a baseline and analyzed four widely recognized aggregation techniques in the literature: Trimmed Mean [7], Krum [10], DRRA, Wei-Detect [18], and standard FedAvg [1]. These techniques were selected for their demonstrated effectiveness in mitigating Byzantine attacks and ensuring robust model aggregation in FL scenarios.

H. Evaluation Metrics

We use accuracy, precision, recall, true positive rates (TPR), false alarm rates (FAR), and F1 scores as evaluation metrics to evaluate the effectiveness of the proposed detection method.

V. Results and Discussion

This section provides a comprehensive analysis of the results, including the model's performance under varying poisoning rates, convergence analysis across communication rounds, the impact of dataset heterogeneity, sensitivity analysis, and the computational and communication overhead of the proposed approach.

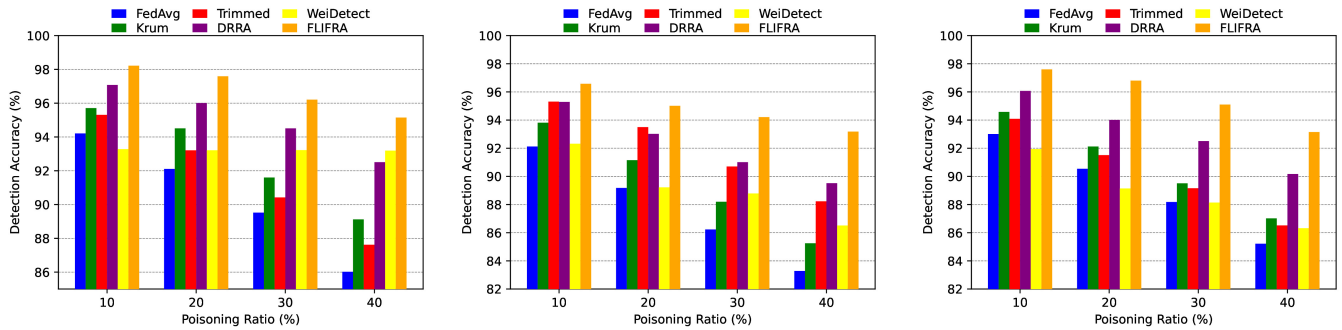
A. Varying Poisoning Attack Intensities

In this experiment, poisoning client rates of 10%, 20%, 30%, and 40% were selected to evaluate the resilience of the framework against different levels of data poisoning in FL. These rates span a range of attack scenarios from mild to severe, thus providing insights into the model's robustness. Similar approaches have been adopted in previous studies such as [5], which used 10% and 30% poisoning rates, and [24], which varied the number of compromised clients to simulate different poisoning intensities. Additionally, in the approaches described in [11], [18], the authors explored poisoning rates up to 50%, further supporting this approach

TABLE II: Impact of poisoning attack intensity on precision and F1-score values across the considered datasets.

Dataset	Int.(%)	FedAvg		Krum		Trimmed Mean		DRRA		WeiDetect		FLIFRA	
		Prec.	F1	Prec.	F1	Prec.	F1	Prec.	F1	Prec.	F1	Prec.	F1
CIC-IDS2018	10	93.8 ± 1.2	94.0 ± 1.1	95.7 ± 1.0	96.0 ± 0.9	95.0 ± 1.1	95.2 ± 1.0	96.8 ± 0.8	97.0 ± 0.8	93.2 ± 0.6	93.0 ± 0.5	**98.0 ± 0.7**	**98.1 ± 0.7**
	20	91.5 ± 1.6	91.8 ± 1.5	94.5 ± 1.2	94.6 ± 1.2	93.2 ± 1.3	93.4 ± 1.3	95.8 ± 1.0	95.9 ± 1.0	92.9 ± 0.3	92.8 ± 0.8	**97.3 ± 0.9**	**97.4 ± 0.9**
	30	88.9 ± 2.1	89.2 ± 2.0	91.6 ± 1.8	91.8 ± 1.7	90.4 ± 2.0	90.5 ± 1.9	94.2 ± 1.6	94.3 ± 1.5	92.8 ± 1.1	92.4 ± 0.6	**96.0 ± 1.2**	**96.1 ± 1.2**
	40	85.5 ± 2.6	85.7 ± 2.5	89.1 ± 2.3	89.3 ± 2.2	87.6 ± 2.4	87.8 ± 2.3	92.2 ± 1.9	92.3 ± 1.8	92.8 ± 0.9	92.5 ± 0.5	**94.8 ± 1.6**	**94.9 ± 1.5**
BoT-IoT	10	91.8 ± 1.3	92.0 ± 1.2	93.5 ± 1.1	93.7 ± 1.0	92.8 ± 1.1	93.0 ± 1.1	95.0 ± 0.9	95.1 ± 0.9	91.5 ± 0.3	91.2 ± 1.1	**96.3 ± 0.8**	**96.4 ± 0.8**
	20	88.5 ± 1.9	88.8 ± 1.8	90.8 ± 1.6	90.9 ± 1.5	90.0 ± 1.6	90.1 ± 1.6	92.8 ± 1.2	92.9 ± 1.2	91.2 ± 0.7	91.0 ± 1.0	**94.8 ± 1.0**	**94.9 ± 1.0**
	30	85.5 ± 2.4	85.7 ± 2.3	87.8 ± 2.1	87.9 ± 2.0	87.3 ± 2.2	87.4 ± 2.1	90.8 ± 1.7	90.9 ± 1.6	91.0 ± 1.0	90.8 ± 1.2	**94.0 ± 1.3**	**94.1 ± 1.3**
	40	82.5 ± 2.9	82.7 ± 2.8	84.8 ± 2.6	84.9 ± 2.5	83.8 ± 2.7	83.9 ± 2.7	89.3 ± 2.0	89.4 ± 2.0	90.6 ± 0.7	90.4 ± 0.9	**92.8 ± 1.8**	**92.9 ± 1.8**
UNSW-NB15	10	92.8 ± 1.0	92.9 ± 1.0	94.3 ± 0.8	94.4 ± 0.8	93.8 ± 0.9	93.9 ± 0.9	95.8 ± 0.7	95.9 ± 0.7	94.2 ± 0.9	93.9 ± 1.0	**96.8 ± 0.6**	**96.9 ± 0.6**
	20	90.2 ± 1.4	90.3 ± 1.4	91.8 ± 1.2	91.9 ± 1.2	91.3 ± 1.3	91.4 ± 1.3	93.8 ± 1.0	93.9 ± 1.0	94.0 ± 1.0	93.9 ± 0.2	**95.8 ± 0.8**	**95.9 ± 0.8**
	30	87.7 ± 1.8	87.8 ± 1.8	89.3 ± 1.5	89.4 ± 1.5	88.8 ± 1.6	88.9 ± 1.6	92.3 ± 1.3	92.4 ± 1.3	93.8 ± 1.0	93.4 ± 1.2	**94.8 ± 1.0**	**94.9 ± 1.0**
	40	84.8 ± 2.2	84.9 ± 2.2	86.8 ± 2.0	86.9 ± 2.0	86.3 ± 2.1	86.4 ± 2.1	89.8 ± 1.7	89.9 ± 1.7	93.6 ± 0.7	93.3 ± 0.4	**92.8 ± 1.5**	**92.9 ± 1.5**

Note. All values as Mean ± Std. dev. (Prec.: Precision, F1: F1-score).



(a) CIC-IDS2018 (b) BoT-IoT (c) UNSW-NB15

Fig. 2: Detection accuracy (%) of baseline and proposed methods across poisoning ratios ($10\% - 40\%$).

for assessing the impact of adversarial influence on model performance.

The results presented in Table II and Figure 2 show a consistent trend in all the considered datasets, with the precision of detection and the F1 score decreasing as the intensity of the poisoning attack increases. In particular, the FedAvg shows significant performance degradation, especially at higher poisoning rates. Robust aggregation methods, Krum, WeiDetect, and Trimmed Mean, display improved resilience; however, they still experience notable accuracy losses under severe attack conditions. In contrast, DRRA achieves higher accuracy than both FedAvg and traditional robust aggregators, indicating its effectiveness in mitigating the influence of poisoned updates. In particular, the proposed FLIFRA approach, which integrates iForest-based filtering on the client side with DRRA, consistently outperforms all other methods. Across varying poisoning intensities, the proposed method not only attains the highest detection accuracy but also exhibits lower variance, as evidenced by the smaller standard deviations. This enhanced performance highlights the effectiveness of the dual-layer defense mechanism in countering data poisoning attacks in federated learning scenarios.

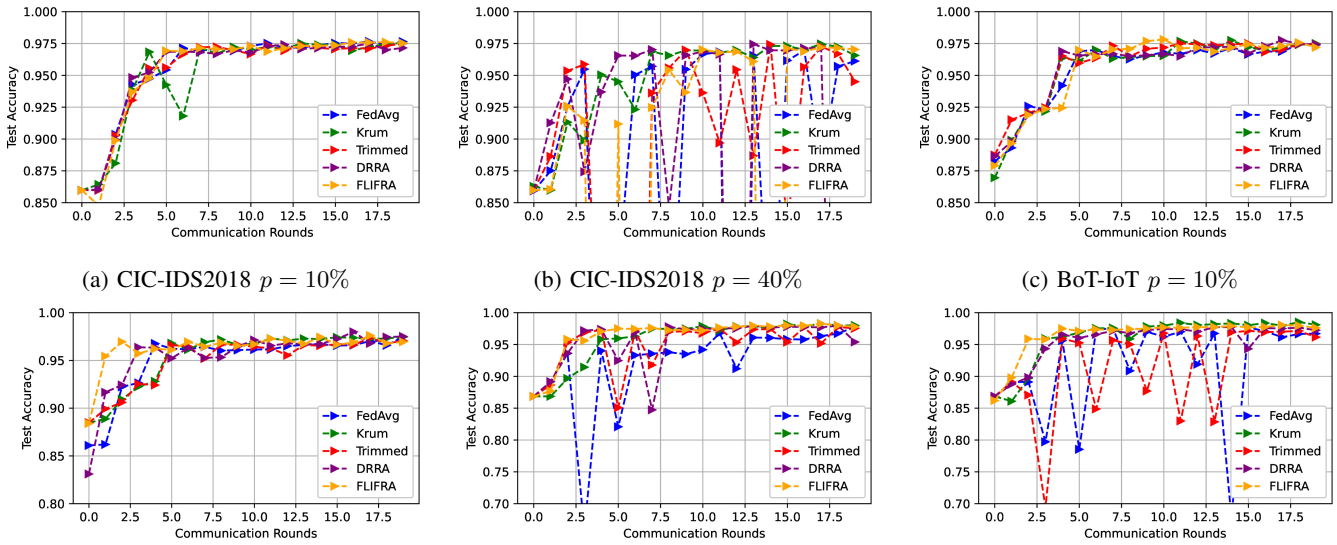
B. Convergence and Communication Round Analysis

The results illustrated in Figure 3 reveal distinct convergence patterns among the evaluated methods: under mild poisoning (10%), all methods tend to converge gradually; however, the proposed FLIFRA quickly reaches a high accuracy level outpacing conventional methods of FedAvg, Krum, and Trimmed Mean. As the intensity of poisoning increases 40%, the convergence gaps become more pronounced. Notably, the FLIFRA method not only achieves a higher steady-state accuracy but also exhibits reduced variance across communication rounds. For example, the results of the BoT-IoT dataset (see Figure 3) demonstrate that while the baseline methods show slower convergence and higher fluctuations, the FLIFRA defense rapidly stabilizes, underscoring its resilience even under severe poisoning scenarios.

The convergence analysis indicates that the proposed FLIFRA method offers significant advantages in terms of convergence speed and stability compared to traditional robust aggregation techniques. The empirical evidence suggests that integrating client-side anomaly detection with server-side dynamic reputation-based aggregation effectively mitigates adversarial influences, ensuring higher final detection accuracy and more consistent performance over time.

C. Effect of the Dataset Heterogeneity

To simulate realistic non-IID conditions in FL, we partition the datasets using a Dirichlet distribution with three distinct concentration parameters (K-values). In our study, $K = 1.0$ represents a near-IID scenario, while $K = 0.5$ and $K = 0.1$ correspond to progressively higher levels of data heterogeneity.



(a) CIC-IDS2018 $p = 10\%$ (b) CIC-IDS2018 $p = 40\%$ (c) BoT-IoT $p = 10\%$

(d) BoT-IoT $p = 40\%$ (e) UNSW-NB15 $p = 10\%$ (f) UNSW-NB15 $p = 40\%$

Fig. 3: Convergence behavior under poisoning attacks with poisoning rates of 10% and 40%.

Table III summarizes the detection accuracy for each method with different K values. Under the low heterogeneity condition ($K = 1.0$), all methods exhibit robust performance, with the FLIFRA approach consistently achieving the highest accuracy. As the K value decreases, indicating more skewed data distributions, a decrease in precision is observed for all methods. The proposed FLIFRA method is more resilient to the adverse effects of data heterogeneity than conventional aggregation strategies. By combining client-side anomaly detection with dynamic reputation-based aggregation, the hybrid approach not only maintains high detection accuracy under near-IID conditions but also exhibits minimal performance loss as the data becomes increasingly non-IID.

D. Sensitivity Analysis

We conducted a sensitivity by varying the iForest contamination parameter η over the set $\{0.01, 0.02, 0.05\}$. For each η, we ran FL experiments under various data-poisoning scenarios and recorded the overall TPR and FAR. At $\eta = 0.05$, we achieved the highest detection sensitivity (TPR $> 95\%$) at the expense of more false alarms (FAR $< 3\%$); at $\eta = 0.02$, we observed a balanced performance with strong detection (TPR $> 90\%$) and moderate false alarms (FAR $< 1.5\%$); and at $\eta = 0.01$, we minimized false alarms (FAR $< 1\%$) a decrease in detection rate (TPR $\approx 85\%$). These results indicate that setting η between 0.02 and 0.05 offers a tunable trade-off between sensitivity and specificity, delivering robust filtering against poisoning attacks.

E. Computational and Communication Overhead

The results presented in Table IV illustrate that although the proposed FLIFRA method incurs in higher computational overhead per communication round, approximately 2.2 min for each round on average for the BoT-IoT dataset compared to 1.2 min for FedAvg, the additional processing cost is offset by a faster convergence speed. For example, the FLIFRA approach converges in about 40 rounds on BoT-IoT, whereas FedAvg requires approximately 50 rounds. Similar trends are observed for CIC-IDS2018 and UNSW-NB15, where the hybrid method consistently converges in fewer rounds despite its higher computational expense per round. This overhead-benefit trade-off underscores the effectiveness of the FLIFRA approach in enhancing detection performance and robustness against poisoning attacks.

From a scalability perspective, although the iForest filtering on the client side of the hybrid method and the reputation computation on the server side introduce additional computation and communication overhead, these costs scale moderately with the number of clients and the dataset size.

VI. Conclusion

This paper proposed a resilient FL framework to address the critical challenge of data poisoning attacks in IoT security systems. The proposed FLIFRA approach integrates local anomaly detection using iForest with a global DRRA to improve robustness against sophisticated adversarial attacks. The framework demonstrated superior resilience and scalability through an extensive evaluation of real-world IoT data sets, including CIC-IDS2018, BoT-IoT, and UNSW-NB15, compared to the baseline methods FedAvg, Trimmed Mean, Krum, DRRA, and WeiDetect. The FLIFRA approach outperformed these methods in mitigating poisoning attacks, achieving higher accuracy and stability at varying levels of malicious client participation ($10\% - 40\%$). The results demonstrate the effectiveness of combining local anomaly detection with dynamic global aggregation. iForest efficiently filters suspicious updates at the client level, preventing compromised data from reaching the server, while DRRA penalizes malicious clients and prioritizes reliable updates through a reputation-based strategy. This dual-layer defense

TABLE III: Impact of data heterogeneity (Dirichlet K-values) on detection accuracy for the considered datasets.

Dataset	K-value	Detection accuracy (%)					
		FedAvg	Krum	Trimmed Mean	DRRA	WeiDetect	FLIFRA (proposed)
CIC-IDS2018	$K = 1.0$	95.0 ± 1.0	96.5 ± 0.8	96.0 ± 0.9	97.2 ± 0.7	93.06 ± 1.0	**98.5 ± 0.6**
	$K = 0.5$	92.0 ± 1.5	94.0 ± 1.2	93.5 ± 1.3	95.5 ± 1.0	90.09 ± 2.0	**97.0 ± 0.8**
	$K = 0.1$	88.0 ± 2.0	90.0 ± 1.8	89.5 ± 1.9	92.0 ± 1.5	89.64 ± 1.5	**95.0 ± 1.2**
BoT-IoT	$K = 1.0$	93.0 ± 1.1	94.2 ± 0.9	94.0 ± 1.0	95.0 ± 0.8	93.12 ± 1.6	**96.2 ± 0.7**
	$K = 0.5$	89.0 ± 1.7	91.0 ± 1.4	90.5 ± 1.5	93.0 ± 1.2	92.81 ± 1.0	**95.0 ± 1.0**
	$K = 0.1$	85.0 ± 2.1	87.0 ± 1.9	86.5 ± 2.0	90.0 ± 1.5	90.08 ± 1.0	**93.0 ± 1.3**
UNSW-NB15	$K = 1.0$	94.0 ± 1.0	95.0 ± 0.8	94.5 ± 0.9	96.0 ± 0.7	92.80 ± 1.0	**97.5 ± 0.6**
	$K = 0.5$	91.0 ± 1.4	92.0 ± 1.2	91.5 ± 1.3	93.5 ± 1.0	90.98 ± 1.1	**95.0 ± 0.8**
	$K = 0.1$	88.0 ± 1.8	89.0 ± 1.5	88.5 ± 1.6	91.0 ± 1.3	88.22 ± 1.2	**93.0 ± 1.0**

Note. All values are presented as Mean ± Std. dev. Lower K-values correspond to higher data heterogeneity.

TABLE IV: Computational overhead and convergence speed.

Method	CIC-IDS2018		BoT-IoT		UNSW-NB15	
	Time-Rounds		Time-Rounds		Time-Rounds	
FedAvg	1.3 ± 0.1	57	1.2 ± 0.1	50	1.2 ± 0.1	51
Krum	1.6 ± 0.2	51	1.5 ± 0.2	48	1.5 ± 0.2	48
Trimmed Mean	1.7 ± 0.2	46	1.6 ± 0.2	47	1.6 ± 0.2	47
DRRA	1.9 ± 0.2	48	1.8 ± 0.2	45	1.8 ± 0.2	45
WeiDetect	2.4 ± 0.1	53	2.3 ± 0.1	46	2.2 ± 0.2	42
FLIFRA (proposed)	**2.3 ± 0.3**	**39**	**2.2 ± 0.3**	**40**	**2.2 ± 0.3**	**40**

Note. Values are reported as Mean ± Std. Dev., with average computation time (min/round) and rounds to convergence.

significantly enhances the robustness of the global model, particularly in heterogeneous and non-IID IoT environments. Moreover, the proposed approach ensures efficient convergence, making it well-suited for resource-constrained IoT systems that demand both computational efficiency and resilience.

Future works will consider a theoretical convergence analysis, exploring additional attack scenarios and varying intensities, evaluating the proposed dual-layer defense in larger and more heterogeneous federated networks, and investigating real-time, privacy-preserving deployment strategies.

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 *Proc. of AIS*, 2017.

[2] Y. Xuan, X. Chen, Z. Zhao, B. Tang, and Y. Dong, "Practical and general backdoor attacks against vertical federated learning," in *Proc. of ECML PKDD*, 2023.

[3] N. M. Jebreel, J. Domingo-Ferrer, D. Sánchez, and A. Blanco-Justicia, "Defending against the label-flipping attack in federated learning," *arXiv:2207.01982*, 2022.

[4] X. Zhang, J. Zhang, K.-H. Chow, J. Chen, Y. Mao, M. Rahouti, X. Li, Y. Liu, and W. Wei, "Visualizing the shadows: Unveiling data poisoning behaviors in federated learning," *arXiv:2405.16707*, 2024.

[5] M. Fang, X. Cao, J. Jia, and N. Gong, "Local model poisoning attacks to byzantine-robust federated learning," in *Proc. of USENIX*, 2020.

[6] J. Sun, A. Li, L. DiValentin, A. Hassanzadeh, Y. Chen, and H. Li, "FL-WBC: Enhancing robustness against model poisoning attacks in federated learning from a client perspective," in *Proc. of NIPS*, 2021.

[7] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *Proc. of ICML*, 2018.

[8] D. N. Yaldiz, T. Zhang, and S. Avestimehr, "Secure federated learning against model poisoning attacks via client filtering," *arXiv:2304.00160*, 2023.

[9] A. Khraisat and A. Alazab, "A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges," *Cybersecurity*, vol. 4, 2021.

[10] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," *Proc. of NIPS*, vol. 30, 2017.

[11] V. Tolpegin, S. Truex, M. E. Gursoy, and L. Liu, "Data poisoning attacks against federated learning systems," in *Proc. of ESORICs*, 2020.

[12] D.-P. Khuu, M. Sober, D. Kaaser, M. Fischer, and S. Schulte, "Data poisoning detection in federated learning," in *Proc. of SAC*, 2024.

[13] N. Rodríguez-Barroso, D. Jiménez-López, M. V. Luzón, F. Herrera, and E. Martínez-Cámara, "Survey on federated learning threats: Concepts, taxonomy on attacks and defences, experimental study and challenges," *Information Fusion*, vol. 90, 2023.

[14] R. Xu, S. Gao, C. Li, J. Joshi, and J. Li, "Dual defense: Enhancing privacy and mitigating poisoning attacks in federated learning," *Proc. of NIPS*, vol. 37, 2024.

[15] H. Fereidooni, A. Pegoraro, P. Rieger, A. Dmitrienko, and A.-R. Sadeghi, "FreqFed: A frequency analysis-based approach for mitigating poisoning attacks in federated learning," *arXiv:2312.04432*, 2023.

[16] Z. Sun, P. Kairouz, A. T. Suresh, and H. B. McMahan, "Can you really backdoor federated learning?" *arXiv:1911.07963*, 2019.

[17] T. D. Nguyen, P. Rieger, H. Chen, H. Yalame, H. Möllering, H. Fereidooni, S. Marchal, M. Miettinen, A. Mirhoseini, S. Zeitouni, F. Koushanfar, A.-R. Sadeghi, and T. Schneider, "FLAME: Taming backdoors in federated learning," in *Proc. of USENIX*, 2022.

[18] S. K. M., V. P., A. Rocha, R. R. K. A., and M. Conti, "WeiDetect: Weibull distribution-based defense against poisoning attacks in federated learning for network intrusion detection systems," *arXiv:2504.04367*, 2025.

[19] T. D. Luong, V. M. Tien, N. H. Quyen, D. T. T. Hien, P. T. Duy, and V.-H. Pham, "Fed-LSAE: Thwarting poisoning attacks against federated cyber threat detection system via autoencoder-based latent space inspection," *Journal of Information Security and Applications*, vol. 87, 2024.

[20] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. of ICISSP*, 2018.

[21] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-IoT dataset," *Future Generation Computer Systems*, vol. 100, 2019.

[22] M. B. Anley, A. Genovese, D. Agostinello, and V. Piuri, "Robust DDoS attack detection with adaptive transfer learning," *Computers & Security*, vol. 144, 2024.

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

[24] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *Proc. of AIS*, 2020.