FedDLAD: A Federated Learning Dual-Layer Anomaly Detection Framework for Enhancing Resilience Against Backdoor Attacks

Binbin Ding[1,2] , **Penghui Yang**[3] , **Sheng-Jun Huang**[1,2*]
[1]MIIT Key Laboratory of Pattern Analysis and Machine Intelligence, Nanjing, China
[2]College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics
[3]College of Computing and Data Science, Nanyang Technological University
{dingbinb, huangsj}@nuaa.edu.cn, phyang.cs@gmail.com

Abstract

In Federated Learning (FL), the decentralized nature of client training introduces vulnerabilities, notably backdoor attacks. Prevailing anomaly detection approaches typically perform binary classification, dividing clients into trusted and untrusted groups. However, these methods face two critical challenges: the *insider threat*, where malicious clients concealed within the trusted group compromise the global model, and the *benign exclusion*, where legitimate contributions from benign clients are mistakenly classified as untrusted and disregarded. These issues weaken both the robustness and fairness of FL systems, exposing inherent defense vulnerabilities. To address these challenges, we propose FedDLAD, a **Fed**erated Learning **D**ual-**L**ayer **A**nomaly **D**etection framework designed to enhance resilience against backdoor attacks. The framework leverages the Connectivity-Based Outlier Factor (COF) module to perform a robust initial classification of clients by analyzing structural data connectivity. The Interquartile Range (IQR) module further reinforces this by mitigating the *insider threat* through the removal of residual malicious influences within the trusted group. Furthermore, the Pardon module dynamically reintegrates misclassified benign clients from the untrusted group, thereby preserving their valuable contributions and addressing the *benign exclusion*. We conduct extensive evaluations of FedDLAD against state-of-the-art defenses on real-world datasets, demonstrating its superior ability to reduce backdoor attack success rates while maintaining robust model performance. Code is available at: https://github.com/dingbinb/FedDLAD.

1 Introduction

Federated Learning (FL) [McMahan *et al.*, 2017] is an emerging distributed machine learning paradigm that allows multiple devices or organizations to collaboratively train models while keeping raw data decentralized. In recent years, FL has attracted considerable attention across various domains, including healthcare [Alzubi *et al.*, 2022; Salim and Park, 2022], financial services [Basu *et al.*, 2021; Chatterjee *et al.*, 2023], and intelligent transportation systems [Manias and Shami, 2021; Zhao *et al.*, 2022; Zhu *et al.*, 2023].

While FL provides substantial privacy benefits by keeping data decentralized, it also introduces security vulnerabilities stemming from the server's limited visibility into local training processes. Existing research [Sun *et al.*, 2019; Bagdasaryan *et al.*, 2020; Fang and Chen, 2023; Zhang *et al.*, 2024] has demonstrated that FL is particularly vulnerable to backdoor attacks. These attacks embed hidden malicious behaviors into the global model, which performs normally on standard inputs but generates attacker-controlled outputs when specific triggers are present. This covert manipulation allows the model to behave normally on typical inputs but produce malicious outputs when triggered, thus evading detection during standard evaluation.

Defense mechanisms against backdoor attacks in FL generally fall into two categories: *robust aggregation* and *anomaly detection*. Robust aggregation enhances model resilience by refining the aggregation process to minimize the influence of malicious updates. Representative algorithms in this category include Krum [Blanchard *et al.*, 2017], Bulyan [Guerraoui *et al.*, 2018], Median [Yin *et al.*, 2018], Trimmed Mean [Yin *et al.*, 2018], RLR [Ozdayi *et al.*, 2021], and FLTrust [Cao *et al.*, 2020]. Byzantine-robust methods such as Krum, Bulyan, Median, and Trimmed Mean are typically designed under the assumption of independent and identically distributed (IID) client data, which limits their effectiveness in non-IID settings. RLR [Ozdayi *et al.*, 2021] mitigates backdoor threats through directional voting and adaptive learning rates, but its performance remains sensitive to variations in data distribution and the malicious client ratio (MCR). FLTrust [Cao *et al.*, 2020] assigns trust scores based on cosine similarity between client and server updates; however, its reliance on a server-side root dataset reduces practicality and may conflict with FL's privacy-preserving principles.

Anomaly detection methods, such as FLAME [Nguyen *et al.*, 2022], FLDetector [Zhang *et al.*, 2022], MultiMetrics [Huang *et al.*, 2023], MASA [Xu *et al.*, 2024], and FedDMC [Mu *et al.*, 2024], are widely used to identify and filter backdoored clients. These methods generally aim to retain only clients considered trustworthy for aggregation, while ex-

[*] Corresponding author

cluding those flagged as suspicious or potentially malicious. However, as single-layer detection approaches, they face considerable limitations in non-IID settings. The increased heterogeneity, data imbalance, and behavioral variability among clients make it significantly more difficult to accurately distinguish between benign and malicious participants. As a result, some malicious clients may evade detection and be mistakenly included, whereas some benign clients may be unjustly excluded. This misclassification not only compromises the security and robustness of the global model, but also introduces unnecessary computational and communication overhead, ultimately degrading the overall efficiency and reliability of the FL system.

To address the limitations of single-layer anomaly detection in enhancing robustness against backdoor attacks, we propose the **Fed**erated Learning **D**ual-**L**ayer **A**nomaly **D**etection Framework (`FedDLAD`). First, the Connectivity-Based Outlier Factor (COF) module assigns anomaly scores to each client's uploaded model parameters, effectively classifying clients into trusted and untrusted groups. To further mitigate residual backdoor threats within the trusted group, we introduce the Interquartile Range (IQR) module as a secondary anomaly detection mechanism. While COF focuses on detecting anomalies in the uploaded model parameters, the IQR module specifically targets client updates. In this phase, anomaly detection is performed across all dimensions of both trusted and untrusted client updates, and detected anomalies are addressed by flipping the direction of abnormal entries.

In addition, to preserve the contributions of benign clients mistakenly classified as untrusted, we aggregate updates from the trusted group to generate a reference update. We then calculate the cosine similarity between each untrusted update and this reference. For updates exhibiting high similarity, a Pardon mechanism is applied, which helps improve overall model performance by retaining valuable contributions.

The main contributions of this paper are as follows:

- We leverage the COF algorithm to classify clients into trusted and untrusted groups based on their uploaded model parameters. To further address potential backdoor clients within the trusted group, we apply the IQR module for secondary anomaly detection, flipping the direction of detected anomalous updates to strengthen the model's robustness against malicious interference.
- To preserve the contributions of benign clients mistakenly assigned to the untrusted group, we introduce a Pardon mechanism. This mechanism aggregates updates from the trusted group to create a reference update, then computes the similarity between each untrusted update and this reference. Updates exhibiting high similarity are pardoned and reintegrated into the global model.
- We conduct comprehensive experiments on multiple datasets and validate the effectiveness of the IQR and Pardon modules through ablation studies. The results demonstrate that the IQR module effectively reduces potential backdoor risks within the trusted group, while the Pardon module successfully reintegrates misclassified benign clients, leading to significant improvements in both overall model performance and fairness.

2 Related Work

Recent advancements in FL defending against backdoor attacks have mainly focused on two defense categories. The first focuses on anomaly detection methods that filter anomalous clients before aggregation, reducing the impact of malicious contributions on the global model. For instance, [Li *et al.*, 2021] employs K-means clustering [Hamerly and Elkan, 2003] to identify suspicious clients; FedDMC [Mu *et al.*, 2024] utilizes binary tree-based clustering with noise (BT-BCN); and FLAME [Nguyen *et al.*, 2022] leverages HDB-SCAN [McInnes *et al.*, 2017] to distinguish between benign and malicious clients. Additionally, FLDetector [Zhang *et al.*, 2022] detects malicious clients based on model consistency, while MultiMetrics [Huang *et al.*, 2023] integrates multiple metrics, including Euclidean distance, Manhattan distance, and Cosine similarity, to identify anomalies. Despite their effectiveness, these methods still face challenges in accurately distinguishing malicious clients from benign ones, resulting in potential misclassification that can compromise system security and degrade overall model performance.

Robust aggregation methods refine aggregation rules to resist malicious updates from backdoored clients. Krum [Blanchard *et al.*, 2017] computes the sum of Euclidean distances between each client's update and all others, selecting the update with the smallest sum as the global model update. The Median [Yin *et al.*, 2018] algorithm selects the median for each dimension, while Trimmed Mean [Yin *et al.*, 2018] removes outliers and computes the average accordingly. However, under non-IID client data, these Byzantine-resilient methods often suffer performance degradation due to the increased diversity and distribution discrepancies in updates, which impair their ability to mitigate malicious contributions effectively. The RLR [Ozdayi *et al.*, 2021] method adjusts the learning rate by voting on update directions in each dimension, but its effectiveness decreases in non-IID scenarios where benign and malicious updates are mixed [Qin *et al.*, 2024]. FLTrust [Cao *et al.*, 2020] relies on the server possessing a root dataset to compute cosine similarity-based trust scores, yet this assumption conflicts with FL's decentralized nature and becomes ineffective when client and server data distributions differ substantially.

3 Preliminaries and Problem setting

3.1 Preliminaries

Connectivity-Based Outlier Factor

The Connectivity-Based Outlier Factor (COF) [Kara and Eyüpoğlu, 2023] is a local anomaly detection algorithm that identifies outliers by evaluating the average chaining distance between each data point and its k-nearest neighbors. For a given point p, COF quantifies its anomaly score based on the degree of connectivity to its neighborhood, where connectivity is measured through the cumulative path length required to traverse from the point to its neighbors. A higher COF score reflects weaker connectivity, indicating that the point deviates from the local structure and is more likely to be an outlier. Conversely, a lower score implies stronger local connectivity, suggesting the point conforms to the surrounding data distribution.

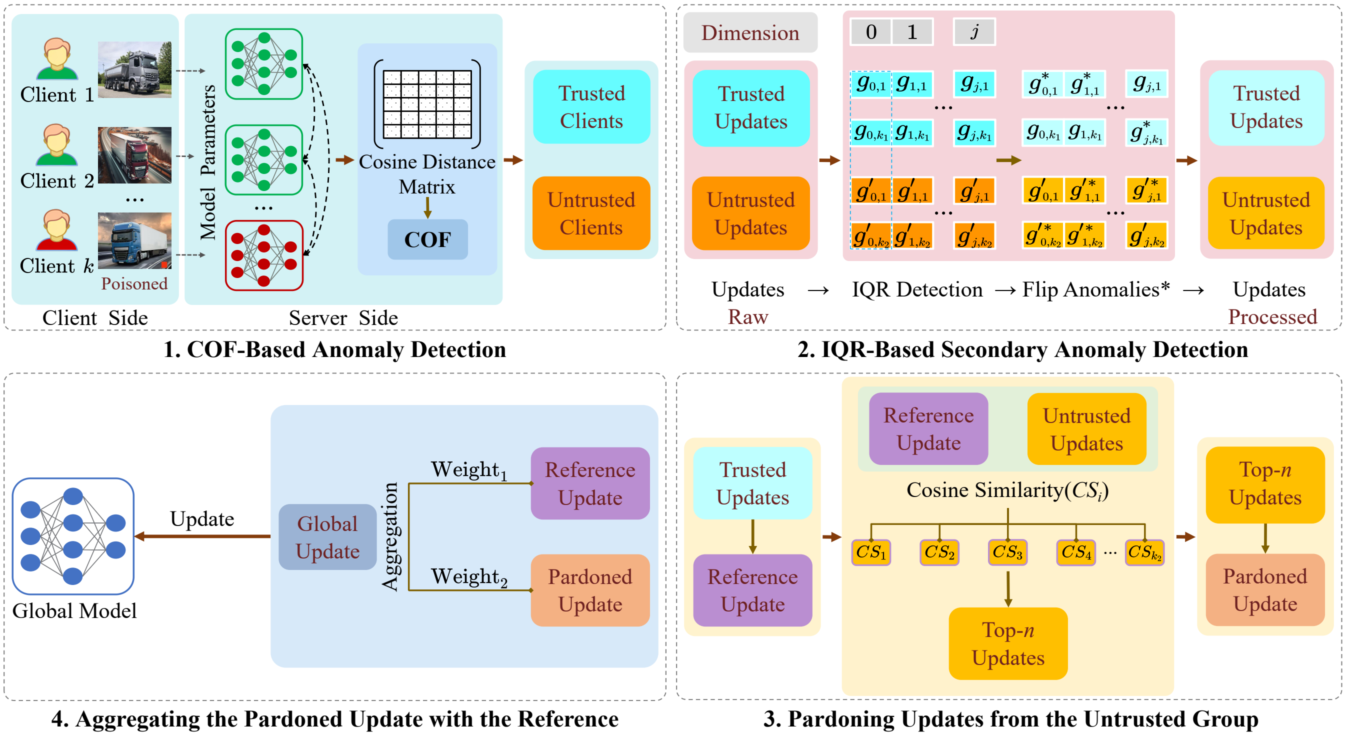
Client 1
Client 2
...
Client k
Poisoned
Model Parameters
Cosine Distance Matrix
COF
Trusted Clients
Untrusted Clients
Client Side
Server Side
1. COF-Based Anomaly Detection
Dimension
0
1
j
Trusted Updates
Untrusted Updates
$g_{0,1}$ $g_{1,1}$ $g_{j,1}$ $g^*_{0,1}$ $g^*_{1,1}$ $g_{j,1}$
g_{0,k_1} g_{1,k_1} g_{j,k_1} g_{0,k_1} g_{1,k_1} g^*_{j,k_1}
$g'_{0,1}$ $g'_{1,1}$ $g'_{j,1}$ $g'_{0,1}$ $g'^*_{1,1}$ $g'^*_{j,1}$
g'_{0,k_2} g'_{1,k_2} g'_{j,k_2} g'^*_{0,k_2} g'^*_{1,k_2} g'_{j,k_2}
Updates → IQR Detection → Flip Anomalies* → Updates
Raw
Processed
2. IQR-Based Secondary Anomaly Detection
Global Model
Update
Global Update
Aggregation
Weight$_1$
Weight$_2$
Reference Update
Pardoned Update
4. Aggregating the Pardoned Update with the Reference
Trusted Updates
Reference Update
Untrusted Updates
Cosine Similarity(CS_i)
CS_1 CS_2 CS_3 CS_4 ... CS_{k_2}
Top-n Updates
Top-n Updates
Pardoned Update
3. Pardoning Updates from the Untrusted Group

Figure 1: Overview of the FedDLAD Framework.

Interquartile Range

The Interquartile Range (IQR) [Vinutha *et al.*, 2018] is a statistical method widely used to identify outliers by analyzing the distribution of data. This technique assesses how far a specific data point deviates from the typical range by examining the dataset's quartiles. The process starts with sorting the dataset, followed by calculating the first quartile (Q_1) and the third quartile (Q_3). The difference between these two quartiles, known as the interquartile range, represents the expected spread of the data. Data points falling outside this range are regarded as potential outliers, which makes the IQR an effective tool for anomaly detection.

Federated Learning

In the FL scenario, each client locally trains the global model broadcast by the server and uploads the locally updated model parameters for aggregation. Assume the FL system consists of N clients, where each client k holds a dataset $\mathcal{D}_k = \{(x_{k,i}, y_{k,i})\}_{i=1}^{n_k}$ of size n_k. The server controls client participation in each training round, and the global training objective is formally defined as follows:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \textstyle\sum_{k=1}^{N} \lambda_k f(\mathbf{w}, \mathcal{D}_k), \tag{1}$$

$$f(\mathbf{w}, \mathcal{D}_k) = \frac{1}{n_k} \textstyle\sum_{i=1}^{n_k} f(\mathbf{w}, (x_{k,i}, y_{k,i})), \tag{2}$$

where $\mathbf{w}^*$ denotes the optimal global model parameters, $f(\mathbf{w}, \mathcal{D}_k)$ represents the average loss computed over client k's dataset $\mathcal{D}_k$, $(x_{k,i}, y_{k,i})$ denotes the i-th sample in $\mathcal{D}_k$, and λ_k indicates the weight assigned to client k's loss.

3.2 Problem Setting

Attacker's Capabilities

We assume that the attacker has access to the local data and model parameters of compromised clients. Under this assumption, the attacker is able to manipulate the samples of targeted classes and alter the local model parameters during training. However, the attacker cannot interfere with the training processes of uncompromised clients nor influence the server's aggregation procedure.

Attacker's Goals

The backdoored model is crafted to predict a specific target class when inputs contain embedded triggers. For instance, in image classification tasks, a backdoored model may misclassify airplane images with triggers as birds, while correctly classifying other clean images. For any input x, the expected output of the backdoored model $\mathbf{M}_\mathbf{w}$ can be formulated as follows:

$$\mathbf{M}_\mathbf{w}(x) = \begin{cases} y & \text{if } x \in \mathcal{D}_{clean}, \\ y_{target} & \text{else}, \end{cases} \tag{3}$$

and the training objective can be formulated as follows:

$$\begin{aligned} &\min_{\mathbf{w}} \underbrace{\mathbb{E}_{(x,\,y)\sim\mathcal{D}_{clean}}\,\ell(\mathbf{M}_\mathbf{w}(x),\,y)}_{\text{Average loss on clean data}} \\ &\quad + \lambda \cdot \underbrace{\mathbb{E}_{(x',\,y_{target})\sim\mathcal{D}_{poison}}\,\ell(\mathbf{M}_\mathbf{w}(x'),\,y_{target})}_{\text{Average loss on poisoned data}}, \end{aligned} \tag{4}$$

where ℓ is the loss function, y_{target} is the target label, $\mathcal{D}_{clean}$ represents the clean dataset, $\mathcal{D}_{poison}$ represents the poisoned dataset, and λ is a balancing parameter.

Limitations of Single-Layer Anomaly Detection

Existing backdoor defense strategies based on single-layer anomaly detection often overlook the necessity of secondary processing [Nguyen *et al.*, 2022; Zhang *et al.*, 2022; Huang *et al.*, 2023; Xu *et al.*, 2024; Mu *et al.*, 2024]. These approaches typically aggregate information solely from the trusted client group, thereby overlooking critical issues such as residual backdoor clients within the trusted group and the exclusion of benign clients mistakenly flagged as untrusted. The effectiveness of anomaly detection algorithms largely depends on the distribution of client data, with optimal performance generally achieved under homogeneous conditions. Nevertheless, the prevalent non-IID nature of client data in FL entangles benign and malicious information, posing substantial challenges for accurate anomaly detection. Therefore, these methods suffer from two major limitations:

Limitation 1. *Residual Backdoor Information*: Although single-layer anomaly detection methods are capable of identifying many malicious clients, they often fall short in reliably distinguishing between benign and malicious clients in complex non-IID scenarios. Consequently, residual malicious contributions may remain within the trusted group, leading to the embedding of backdoor triggers into the global model across multiple training rounds.

Limitation 2. *Misclassification of Benign Information*: Due to inherent limitations in detection accuracy, some benign clients are mistakenly classified as untrusted. As such, their valuable contributions are excluded from the global model update, resulting in both wasted communication and computational resources. This misclassification not only impedes potential model improvements but also undermines overall system efficiency.

4 Methodology

4.1 Overview of the FedDLAD Framework

As illustrated in Figure 1, the FedDLAD framework consists of the following steps: (1) COF-Based Anomaly Detection, (2) IQR-Based Secondary Anomaly Detection, (3) Pardoning Updates from the Untrusted Group, and (4) Aggregating the Pardoned Update with the Reference.

Building upon the COF-based anomaly detection applied to model parameters, the IQR module serves as a secondary detector designed to address **Limitation 1**. This module thoroughly inspects updates across all dimensions within both trusted and untrusted groups. It effectively identifies anomalous updates and mitigates their effects by flipping their directions. To further resolve **Limitation 2**, FedAvg [McMahan *et al.*, 2017] is utilized to aggregate the IQR-processed updates from the trusted group and produce a reliable reference update. Subsequently, the cosine similarity between this reference and each update from the untrusted group is computed. Updates exhibiting the highest similarity scores are pardoned and reintegrated into the global model.

4.2 Detailed Methodology

COF-Based Anomaly Detection

At the end of communication round t, the server collects model parameters from k clients and constructs a cosine distance matrix $\mathbf{M} \in \mathbb{R}^{k\times k}$, where each entry represents the pairwise distance between two clients' model parameters. The matrix $\mathbf{M}$ is then processed by the COF algorithm to compute anomaly scores for each client, denoted as $score_p = \mathrm{COF}(\mathbf{M})(p = 1, 2, 3, \ldots, k)$. Based on these scores, the server partitions the clients into two disjoint groups: the trusted group $C = \{c_1, c_2, c_3, \ldots, c_{k_1}\}$ and the untrusted group $C' = \{c'_1, c'_2, c'_3, \ldots, c'_{k_2}\}$, where $k_1 + k_2 = k$, with k_1 and k_2 representing the number of clients in C and C', respectively.

IQR-Based Secondary Anomaly Detection

Following the COF module, a secondary anomaly detection is performed using the IQR module to mitigate the limitations of COF. For the current round t, the server obtains updates $G = \{g_1, g_2, g_3, \ldots, g_{k_1}\}$ from trusted group and $G' = \{g'_1, g'_2, g'_3, \ldots, g'_{k_2}\}$ from untrusted group. Then, for each dimension $d \in \{0, 1, 2, ..., j\}$, the server computes the first quartile ${Q_1}^{(d)}$ and third quartile ${Q_3}^{(d)}$ over all k data points and derives the interquartile range: $\mathrm{IQR}^{(d)} = {Q_3}^{(d)} - {Q_1}^{(d)}$. Based on this, the lower and upper bounds of normal values in dimension d are calculated as:

$$\Phi^{(d)}_{\text{lower}} = Q^{(d)}_1 - \mu \times \mathrm{IQR}^{(d)}, \tag{5}$$

$$\Phi^{(d)}_{\text{upper}} = Q^{(d)}_3 + \mu \times \mathrm{IQR}^{(d)}, \tag{6}$$

where the parameter μ controls the sensitivity of anomaly detection by adjusting the width of the normal range. A smaller μ results in a narrower range, causing more data points to be identified as anomalies, whereas a larger μ expands the range, thereby classifying more points as normal.

For any dimension d of a data point $x_{i,d}(i = 1, 2, 3, ..., k)$, if the value $x_{i,d}$ falls outside the interval $[\Phi^{(d)}_{\text{lower}}, \Phi^{(d)}_{\text{upper}}]$ computed for that dimension, a flipping operation is applied to adjust it. The adjustment procedure for the value $x_{i,d}$ in each dimension is defined as follows:

$$x_{i,d} = \begin{cases} x_{i,d} & \text{if } x_{i,d} \in [\Phi^{(d)}_{\text{lower}}, \Phi^{(d)}_{\text{upper}}], \\ -x_{i,d} & \text{else.} \end{cases} \tag{7}$$

Pardoning Updates from the Untrusted Group

After applying the IQR module, the adjusted updates $G^* = \{g^*_1, g^*_2, g^*_3, \ldots, g^*_{k_1}\}$ from the trusted group are aggregated using FedAvg [McMahan *et al.*, 2017] to produce the reference update φ:

$$\varphi = \frac{\sum_{i=1}^{k_1} n_i g^*_i}{\sum_{i=1}^{k_1} n_i}, \tag{8}$$

where n_i denotes the data size of the i-th client in the trusted group, and g^*_i is the corresponding adjusted update.

The server computes the cosine similarity scores $cs_i(i = 1, 2, 3, \ldots, k_2)$ between the reference update φ and each adjusted update $G'^* = \{{g'_1}^*, {g'_2}^*, {g'_3}^*, \ldots, {g'_{k_2}}^*\}$ from the untrusted group. This step enables the server to evaluate how closely each untrusted update aligns with the reference, providing a more informed basis for assessing their trustworthiness.

$$cs_i = \frac{\varphi \cdot {g'_i}^*}{\|\varphi\| \|{g'_i}^*\|}, \tag{9}$$

where $g_i'^*$ denotes the adjusted update of the i-th client in the untrusted group.

Given the set $S = \{cs_1, cs_2, cs_3, \ldots, cs_{k_2}\}$ of cosine similarity scores between the reference update and each update in the untrusted group, the server proceeds as follows:

- *Filter the scores*: Retain only the cosine similarity scores greater than 0 from the set S, resulting in the subset $S^+ = \{cs_i \in S \mid cs_i > 0\}$.
- *Select the $Top - n$ scores*: From the subset S^+, select the top n highest cosine similarity scores to form a new set $S' = \{cs_i \in S^+ \mid cs_i \in \mathrm{Top}_n(S^+)\}$.

The server aggregates the updates $g_i'^*$ corresponding to the scores $cs_i \in S^{'}$ to obtain the pardoned update ϕ:

$$\phi = \frac{\sum_{cs_i \in S'} cs_i g_i'^*}{\sum_{cs_i \in S'} cs_i}. \tag{10}$$

Aggregating the Pardoned Update with the Reference

We assign the number of clients in the trusted group as the weight for the reference update, denoted by $w_1 (w_1 = k_1)$. Since the number of pardoned clients in the untrusted group may vary across training rounds, the weight w_2 for the pardoned update is computed as follows:

$$w_2 = \begin{cases} |S^+| & \text{if } |S^+| < n, \\ n & \text{else.} \end{cases} \tag{11}$$

Here, $|S^+|$ denotes the size of the set S^+.

The server performs a weighted aggregation of the reference update φ and the pardoned update ϕ to produce the final aggregated global update ω:

$$\omega = \frac{w_1}{w_1 + w_2}\varphi + \frac{w_2}{w_1 + w_2}\phi. \tag{12}$$

Ultimately, the server updates the global model at round t using the final aggregated global update ω.

5 Experimental Evaluation

5.1 Experimental Setup

Training Setup. We deploy experiments using the PyTorch framework [Paszke *et al.*, 2019], employing the SGD optimizer with a learning rate of 0.01. The FL system consists of 50 clients, with 20% of them assumed to be malicious by default. These malicious clients select samples with a ground-truth label of '0' and modify their labels to the target class '5' to carry out attacks. In each training round, the server randomly selects 50% of the clients to participate, and the global learning rate is set to 1.

Attack Setup. We implement several attack methods, including CBA [Bagdasaryan *et al.*, 2020], DBA [Xie *et al.*, 2019], and SPA [Wang *et al.*, 2020], to evaluate the robustness of our defense mechanism against diverse adversarial threats.

- **Centralized Backdoor Attack (CBA).** Each malicious client has full access to the trigger, meaning they possess the entire trigger pattern utilized for backdoor attacks. In CBA, a square trigger is embedded in the bottom-right corner of the attacked samples.
- **Distributed Backdoor Attack (DBA).** The DBA decomposes the complete trigger into multiple parts, with each malicious client using a local trigger in their attack samples. This attack enhances stealthiness while reducing the risk of detection. In DBA, we employ a cross-shaped trigger, with each malicious client holding one quarter of the full trigger.
- **Single-shot combined with PGD Attack (SPA).** The Single-shot attack, essentially a model replacement attack, has the malicious client replace the global model with its local model during aggregation. The Projected Gradient Descent (PGD) attack limits the norm of malicious updates to evade the server's Euclidean distance-based anomaly detection. Both methods are combined to enhance the attack's effectiveness and stealth.

Datasets. We conduct experiments using four widely-used benchmark datasets: MNIST [Deng, 2012], FashionMNIST [Xiao *et al.*, 2017], SVHN [Netzer *et al.*, 2011], and CIFAR-10 [Krizhevsky *et al.*, 2009], which are well recognized in the machine learning community for their diverse characteristics and challenges.

Models. We utilize a Convolutional Neural Network (CNN) architecture for both MNIST and FashionMNIST, each comprising 5 convolutional layers. The MNIST model includes 2 fully connected layers, whereas the FashionMNIST model incorporates 4 fully connected layers. For SVHN and CIFAR-10, we employ the VGG-9 architecture for training.

Baselines. We conduct a comprehensive comparison of our proposed method, `FedDLAD`, against several baseline defense algorithms: FedAvg [McMahan *et al.*, 2017], Krum [Blanchard *et al.*, 2017], Median [Yin *et al.*, 2018], RLR [Ozdayi *et al.*, 2021], FLTrust [Cao *et al.*, 2020], FoolsGold [Fung *et al.*, 2020], FLAME [Nguyen *et al.*, 2022], MultiMetrics [Huang *et al.*, 2023], and SnowBall [Qin *et al.*, 2024]. FedAvg serves as the primary baseline with its standard aggregation rule, providing a foundational reference for evaluating the performance of other defense methods.

Evaluation Metrics. We employ the following 3 metrics to evaluate the performance of various defense algorithms.

- **Attack Success Rate (ASR).** ASR measures the proportion of trigger samples that the model classifies as the target class. This metric reflects the effectiveness of the backdoor attack by indicating how successfully the attack manipulates the model to produce the desired output for malicious inputs.
- **Natural Accuracy (ACC).** ACC quantifies the model's performance on clean data, reflecting its effectiveness in standard classification tasks. This metric indicates how well the model generalizes to benign inputs.
- **Overall Performance Score (OPS).** Following [Huang *et al.*, 2023], we use the OPS metric to evaluate the performance improvements of different defense methods in terms of ACC and ASR relative to the baseline FedAvg. The OPS is calculated as OPS $= \frac{D_{\mathrm{ACC}} - B_{\mathrm{ACC}}}{B_{\mathrm{ACC}}} - \frac{D_{\mathrm{ASR}} - B_{\mathrm{ASR}}}{B_{\mathrm{ASR}}}$, where D_{ACC} and D_{ASR} denote the ACC and ASR of the evaluated defense method, respectively; B_{ACC} and B_{ASR} correspond to the baseline values.

Dirichlet(α)	Defense	MNIST			FashionMNIST			SVHN			CIFAR-10		
		ASR↓	ACC↑	OPS↑	ASR↓	ACC↑	OPS↑	ASR↓	ACC↑	OPS↑	ASR↓	ACC↑	OPS↑
$\alpha = 0.5$	FedAvg	1	0.993	0	0.954	0.913	0	0.810	0.918	0	0.854	0.778	0
	Krum	0.009	0.974	+0.972	0.009	0.782	+0.847	0.615	0.209	-0.532	**0.021**	0.456	+0.562
	Median	0.998	**0.993**	+0.002	0.624	0.883	+0.313	0.832	0.916	-0.029	0.816	0.771	+0.035
	RLR	**0.002**	0.990	+0.995	**0.003**	0.862	+0.941	0.772	0.838	-0.040	0.739	0.715	+0.054
	FLTrust	0.005	**0.993**	+0.995	0.006	0.802	+0.872	0.062	0.884	+0.886	0.507	0.775	+0.402
	FoolsGold	0.993	0.992	+0.006	0.896	**0.898**	+0.044	0.836	**0.923**	-0.027	0.836	**0.781**	+0.025
	FLAME	0.226	0.992	+0.773	0.442	0.883	+0.504	0.751	0.875	+0.026	0.783	0.728	+0.019
	MultiMetrics	0.009	0.989	+0.987	0.009	0.882	+0.957	0.662	0.812	+0.067	0.786	0.721	+0.006
	SnowBall	0.007	0.980	+0.980	0.017	0.870	+0.935	0.817	0.844	-0.089	0.722	0.723	+0.084
	FedDLAD	0.003	0.992	**+0.996**	0.007	0.888	**+0.965**	**0.044**	0.895	**+0.921**	0.025	0.746	**+0.930**
$\alpha = 1$	FedAvg	1	0.990	0	0.967	0.913	0	0.843	0.924	0	0.846	0.783	0
	Krum	0.967	0.958	+0.001	0.009	0.793	+0.859	0.870	0.797	-0.169	**0.006**	0.449	+0.566
	Median	0.997	0.991	+0.004	0.923	0.896	+0.027	0.846	0.920	-0.008	0.764	0.767	+0.076
	RLR	0.002	0.989	+0.997	**0.003**	0.876	+0.956	0.768	0.874	+0.035	0.046	0.714	+0.858
	FLTrust	0.012	0.988	+0.986	0.006	0.810	+0.881	0.091	0.868	+0.831	0.022	0.731	+0.908
	FoolsGold	0.917	0.988	+0.081	0.921	**0.907**	+0.041	0.837	**0.929**	+0.013	0.822	**0.785**	+0.031
	FLAME	0.992	0.984	+0.002	0.944	0.889	-0.003	0.773	0.882	+0.038	0.160	0.724	+0.736
	MultiMetrics	0.032	0.992	+0.970	0.018	0.824	+0.884	0.861	0.872	-0.078	0.296	0.751	+0.609
	SnowBall	**0.001**	0.990	**+0.999**	0.025	0.762	+0.809	0.503	0.837	+0.449	0.546	0.757	+0.321
	FedDLAD	0.004	**0.993**	**+0.999**	0.007	0.882	**+0.959**	**0.003**	0.904	**+0.975**	0.038	0.751	**+0.914**

Table 1: Performance under non-IID data. ↑ means higher is better, ↓ means lower is better. **Bold** numbers indicate the best results.

Defense	CBA		DBA		SPA		$\overline{\text{OPS}}$↑
	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	
FedAvg	0.866	0.802	0.860	0.800	0.764	0.801	0
Krum	0.011	0.695	0.006	0.655	0.016	0.679	+0.831
Median	0.851	**0.801**	0.819	0.794	0.005	0.791	+0.346
RLR	0.274	0.738	0.023	0.739	0.003	0.792	+0.829
FLTrust	0.803	0.780	0.543	0.760	**0.001**	0.784	+0.447
FoolsGold	0.852	0.797	0.796	0.788	0.801	0.790	+0.002
FLAME	**0.009**	0.772	**0.002**	0.782	0.004	0.788	+0.969
MultiMetrics	0.050	0.788	0.233	0.775	0.002	0.795	+0.871
SnowBall	0.317	0.796	0.249	**0.795**	0.013	**0.802**	+0.772
FedDLAD	0.033	0.793	**0.002**	0.791	0.002	0.795	**+0.976**

Table 2: Performance under different attack settings with IID data.

5.2 Analysis of the Experimental Results

Performance under Different Data Distributions

Defenses under non-IID. We simulate real-world FL data distributions by partitioning the MNIST, FashionMNIST, SVHN, and CIFAR-10 datasets with the Dirichlet distribution **Dirichlet**(α) [Li *et al.*, 2022], setting α to 0.5 and 1 to control data heterogeneity.

As shown in Table 1, the defense effectiveness of Median and FoolsGold is limited. The Median algorithm, which aggregates updates by taking the median, is vulnerable to malicious updates in non-IID environments, thereby weakening its defense capability. FoolsGold assumes high similarity among malicious client updates, which does not always hold under diverse data distributions. Krum provides moderate defense on MNIST and FashionMNIST but performs poorly on SVHN and CIFAR-10. RLR reduces ASR to 0.2% and 0.3% on MNIST and FashionMNIST, respectively, with minimal drops in ACC, but struggles on SVHN and CIFAR-10 due to the increased complexity of client updates. FLTrust delivers reasonable performance on MNIST and FashionMNIST; however, on FashionMNIST, the ASR remains at 0.6% with an 11% drop in ACC. FLAME's adaptive noise alleviates some of the detrimental effects of client misclassification, but it still undermines benign performance. MultiMetrics and SnowBall demonstrate relatively effective defense on MNIST and FashionMNIST, while facing difficulties on SVHN and CIFAR-10. In contrast, FedDLAD consistently performs well across all datasets, particularly on SVHN, where it reduces ASR to 2.4% with only a 2.2% drop in ACC. The OPS metric further confirms that FedDLAD effectively suppresses ASR while preserving benign accuracy, outperforming other methods overall.

Defenses under IID. In the IID scenario, we assess the performance of several defense strategies against CBA, DBA, and SPA on the CIFAR-10 dataset.

As shown in Table 2, Krum effectively reduces the average ASR to approximately 1% for all attack types, although it results in an average ACC drop of 12.5%. Median, MultiMetrics, and SnowBall demonstrate better defense performance under SPA compared to CBA and DBA. Both RLR and FLTrust exhibit a marked decline in effectiveness against CBA, with RLR's ASR reaching 27.4% and FLTrust rising to 80.3%. By comparison, FLAME and FedDLAD display greater stability, with FedDLAD maintaining an ASR similar to FLAME while improving ACC by 1.2%. Moreover, FedDLAD achieves the highest average OPS across all attack scenarios, underscoring its superior defense capabilities.

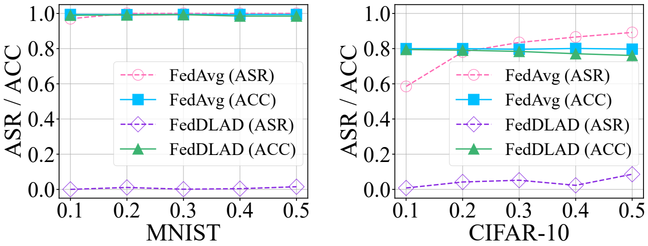

Figure 2: Performance across various MCRs.

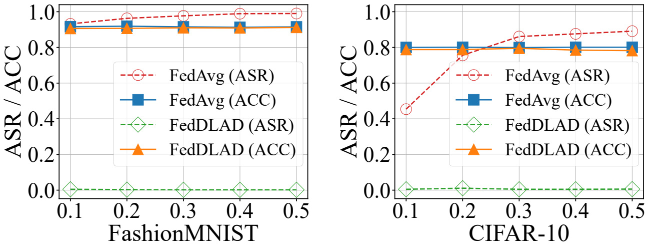

Figure 3: Performance across various PDRs.

Impact of Different MCR and PDR Settings

Figure 2 and Figure 3 present the effects of varying MCR (Malicious Client Ratio) and PDR (Poisoned Data Ratio) on the performance of FedDLAD under IID conditions.

Impact of MCR. As shown in Figure 2, on MNIST, FedDLAD maintains its effectiveness across different MCRs, achieving optimal defense at an MCR of 0.1, where the ASR is 0 and the ACC declines by only 0.4%. As MCR increases, the ASR rises slightly but remains below 1.5%. On CIFAR-10, FedDLAD shows greater sensitivity to MCR, with the ASR increasing to 8.6% and the ACC dropping by 3.6% at an MCR of 0.5, compared to FedAvg.

Impact of PDR. Figure 3 illustrates that as PDR increases, the ASR of FedAvg rises significantly. FedDLAD maintains robust defense performance on FashionMNIST and CIFAR-10, with the ASR remaining close to 0 even at a PDR of 0.5, while the ACC decreases by no more than 2%.

Various Approaches for Handling IQR Anomalies

We evaluate the performance of various methods in handling anomalous updates identified by the IQR detector on the MNIST, FashionMNIST, SVHN, and CIFAR-10 datasets. As shown in Figure 4, Zeroing and Median replacement methods perform poorly against attacks on MNIST, FashionMNIST, and SVHN. Although their performance improves slightly on CIFAR-10, the ASR remains above 40%. Conversely, the flip method maintains a low average ASR of just 2.4% across all datasets, demonstrating significantly greater stability than both Zeroing and Median replacement.

Ablation Study

We conduct ablation studies under IID and non-IID settings to assess the contribution of each component of FedDLAD, which comprises three core modules: (1) the COF module, (2) the IQR module, and (3) the Pardon module.

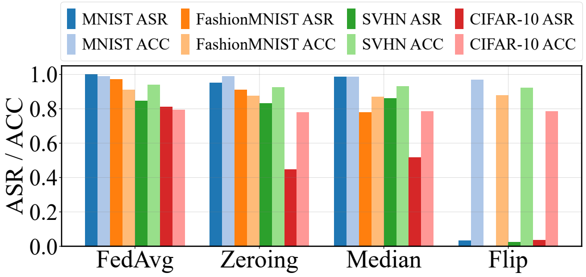

Figure 4: Comparison of various methods for handling outliers.

Data	Module			FashionMNIST		SVHN		CIFAR-10	
	COF	IQR	Pardon	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑
IID	✓			0.003	0.904	0.005	0.921	**0.004**	0.761
	✓	✓		**0.002**	0.901	**0.004**	0.920	0.007	0.765
	✓	✓	✓	0.003	**0.907**	0.007	**0.931**	0.010	**0.785**
non-IID	✓			0.963	0.870	0.856	0.886	0.500	0.733
	✓	✓		**0.018**	0.867	0.018	0.893	0.082	0.744
	✓	✓	✓	0.035	**0.877**	**0.005**	**0.904**	**0.047**	**0.762**

Table 3: Performance of modules under different data distributions.

Under IID Data. As shown in Table 3, the COF module effectively distinguishes between benign and malicious clients, driving the ASR to nearly 0 while maintaining a high ACC. In this context, the contribution of the IQR module is relatively modest. The Pardon module boosts ACC by 1.1% on SVHN and 2% on CIFAR-10.

Under non-IID Data. The COF algorithm struggles with non-IID data, leading to an average ASR of 77.3%. However, incorporating the IQR module significantly reduces the ASR to below 4%, highlighting its effectiveness in secondary detection and improving resilience against backdoor attacks. Additionally, adding the Pardon module boosts the average ACC across all datasets by 1.3%.

In summary, the COF and IQR modules play distinct roles depending on the data distribution. Under IID conditions, the COF module is critical, while in non-IID environments, the IQR module becomes essential for maintaining robust defense performance as the effectiveness of COF diminishes.

6 Conclusion and Future Work

In this paper, we present FedDLAD, a dual-layer framework designed to enhance backdoor defense by overcoming the limitations of single-layer detection methods. The IQR module addresses the *insider threat* posed by malicious clients, while the Pardon module mitigates *benign exclusion* by reintegrating misclassified benign clients. Together, these modules synergistically strengthen the defense's robustness and improve overall performance. Considering the challenges that backdoor attacks pose in FL, future work will aim to further improve the framework's effectiveness, especially in the presence of non-IID data.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (U2441285, 62222605) and the Natural Science Foundation of Jiangsu Province of China (BK20222012).

References

[Alzubi *et al.*, 2022] Jafar A Alzubi, Omar A Alzubi, Ashish Singh, and Manikandan Ramachandran. Cloud-iiot-based electronic health record privacy-preserving by cnn and blockchain-enabled federated learning. *IEEE Transactions on Industrial Informatics*, 19(1):1080–1087, 2022.

[Bagdasaryan *et al.*, 2020] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *International conference on artificial intelligence and statistics*, pages 2938–2948. PMLR, 2020.

[Basu *et al.*, 2021] Priyam Basu, Tiasa Singha Roy, Rakshit Naidu, and Zumrut Muftuoglu. Privacy enabled financial text classification using differential privacy and federated learning. *arXiv preprint arXiv:2110.01643*, 2021.

[Blanchard *et al.*, 2017] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. *Advances in neural information processing systems*, 30, 2017.

[Cao *et al.*, 2020] Xiaoyu Cao, Minghong Fang, Jia Liu, and Neil Zhenqiang Gong. Fltrust: Byzantine-robust federated learning via trust bootstrapping. *arXiv preprint arXiv:2012.13995*, 2020.

[Chatterjee *et al.*, 2023] Pushpita Chatterjee, Debashis Das, and Danda B Rawat. Next generation financial services: Role of blockchain enabled federated learning and metaverse. In *2023 IEEE/ACM 23rd International Symposium on Cluster, Cloud and Internet Computing Workshops (CCGridW)*, pages 69–74. IEEE, 2023.

[Deng, 2012] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE signal processing magazine*, 29(6):141–142, 2012.

[Fang and Chen, 2023] Pei Fang and Jinghui Chen. On the vulnerability of backdoor defenses for federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 11800–11808, 2023.

[Fung *et al.*, 2020] Clement Fung, Chris JM Yoon, and Ivan Beschastnikh. The limitations of federated learning in sybil settings. In *23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020)*, pages 301–316, 2020.

[Guerraoui *et al.*, 2018] Rachid Guerraoui, Sébastien Rouault, et al. The hidden vulnerability of distributed learning in byzantium. In *International Conference on Machine Learning*, pages 3521–3530. PMLR, 2018.

[Hamerly and Elkan, 2003] Greg Hamerly and Charles Elkan. Learning the k in k-means. *Advances in neural information processing systems*, 16, 2003.

[Huang *et al.*, 2023] Siquan Huang, Yijiang Li, Chong Chen, Leyu Shi, and Ying Gao. Multi-metrics adaptively identifies backdoors in federated learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4652–4662, 2023.

[Kara and Eyüpoğlu, 2023] Burak Cem Kara and Can Eyüpoğlu. A new privacy-preserving data publishing algorithm utilizing connectivity-based outlier factor and mondrian techniques. *Computers, Materials & Continua*, 76(2), 2023.

[Krizhevsky *et al.*, 2009] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[Li *et al.*, 2021] Dongcheng Li, W Eric Wong, Wei Wang, Yao Yao, and Matthew Chau. Detection and mitigation of label-flipping attacks in federated learning systems with kpca and k-means. In *2021 8th International Conference on Dependable Systems and Their Applications (DSA)*, pages 551–559. IEEE, 2021.

[Li *et al.*, 2022] Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. Federated learning on non-iid data silos: An experimental study. In *2022 IEEE 38th international conference on data engineering (ICDE)*, pages 965–978. IEEE, 2022.

[Manias and Shami, 2021] Dimitrios Michael Manias and Abdallah Shami. Making a case for federated learning in the internet of vehicles and intelligent transportation systems. *IEEE network*, 35(3):88–94, 2021.

[McInnes *et al.*, 2017] Leland McInnes, John Healy, Steve Astels, et al. hdbscan: Hierarchical density based clustering. *J. Open Source Softw.*, 2(11):205, 2017.

[McMahan *et al.*, 2017] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[Mu *et al.*, 2024] Xutong Mu, Ke Cheng, Yulong Shen, Xiaoxiao Li, Zhao Chang, Tao Zhang, and Xindi Ma. Feddmc: Efficient and robust federated learning via detecting malicious clients. *IEEE Transactions on Dependable and Secure Computing*, 2024.

[Netzer *et al.*, 2011] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 4. Granada, 2011.

[Nguyen *et al.*, 2022] Thien Duc Nguyen, Phillip Rieger, Roberta De Viti, Huili Chen, Björn B Brandenburg, Hossein Yalame, Helen Möllering, Hossein Fereidooni, Samuel Marchal, Markus Miettinen, et al. Flame: Taming backdoors in federated learning. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 1415–1432, 2022.

[Ozdayi *et al.*, 2021] Mustafa Safa Ozdayi, Murat Kantarcioglu, and Yulia R Gel. Defending against backdoors in federated learning with robust learning rate. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 9268–9276, 2021.

[Paszke *et al.*, 2019] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[Qin *et al.*, 2024] Zhen Qin, Feiyi Chen, Chen Zhi, Xueqiang Yan, and Shuiguang Deng. Resisting backdoor attacks in federated learning via bidirectional elections and individual perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 14677–14685, 2024.

[Salim and Park, 2022] Mikail Mohammed Salim and Jong Hyuk Park. Federated learning-based secure electronic health record sharing scheme in medical informatics. *IEEE Journal of Biomedical and Health Informatics*, 27(2):617–624, 2022.

[Sun *et al.*, 2019] Ziteng Sun, Peter Kairouz, Ananda Theertha Suresh, and H Brendan McMahan. Can you really backdoor federated learning? *arXiv preprint arXiv:1911.07963*, 2019.

[Vinutha *et al.*, 2018] HP Vinutha, B Poornima, and BM Sagar. Detection of outliers using interquartile range technique from intrusion dataset. In *Information and decision sciences: Proceedings of the 6th international conference on ficta*, pages 511–518. Springer, 2018.

[Wang *et al.*, 2020] Hongyi Wang, Kartik Sreenivasan, Shashank Rajput, Harit Vishwakarma, Saurabh Agarwal, Jy-yong Sohn, Kangwook Lee, and Dimitris Papailiopoulos. Attack of the tails: Yes, you really can backdoor federated learning. *Advances in Neural Information Processing Systems*, 33:16070–16084, 2020.

[Xiao *et al.*, 2017] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[Xie *et al.*, 2019] Chulin Xie, Keli Huang, Pin-Yu Chen, and Bo Li. Dba: Distributed backdoor attacks against federated learning. In *International conference on learning representations*, 2019.

[Xu *et al.*, 2024] Jiahao Xu, Zikai Zhang, and Rui Hu. Identify backdoored model in federated learning via individual unlearning. *arXiv preprint arXiv:2411.01040*, 2024.

[Yin *et al.*, 2018] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International conference on machine learning*, pages 5650–5659. PMLR, 2018.

[Zhang *et al.*, 2022] Zaixi Zhang, Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. Fldetector: Defending federated learning against model poisoning attacks via detecting malicious clients. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pages 2545–2555, 2022.

[Zhang *et al.*, 2024] Hangfan Zhang, Jinyuan Jia, Jinghui Chen, Lu Lin, and Dinghao Wu. A3fl: Adversarially adaptive backdoor attacks to federated learning. *Advances in Neural Information Processing Systems*, 36, 2024.

[Zhao *et al.*, 2022] Jianxin Zhao, Xinyu Chang, Yanhao Feng, Chi Harold Liu, and Ningbo Liu. Participant selection for federated learning with heterogeneous data in intelligent transport system. *IEEE transactions on intelligent transportation systems*, 24(1):1106–1115, 2022.

[Zhu *et al.*, 2023] Rongbo Zhu, Mengyao Li, Jiangjin Yin, Lubing Sun, and Hao Liu. Enhanced federated learning for edge data security in intelligent transportation systems. *IEEE Transactions on Intelligent Transportation Systems*, 24(11):13396–13408, 2023.