Feedback-Guided Black-box Attack in Federated Learning: A Cautious Attacker Perspective

Anonymous authors
Paper under double-blind review

Abstract

Federated Learning (FL) is a robust approach to collaborative machine learning that upholds the integrity of data privacy by ensuring that data remains with the owners. However, FL systems are vulnerable to sophisticated adversarial attacks from malicious clients, especially those leveraging black-box settings. Unlike centralized data poisoning, attacking FL presents unique challenges (i) server-side defense mechanisms can detect and discard suspicious client updates, requiring attacks to maintain minimal visibility across multiple training rounds, and (ii) malicious clients must repeatedly generate poisoned data using only their local black-box model for each round of training, as previous poisoning attempts may be nullified during global aggregation. This forces adversaries to craft stealthy poisoned data locally in a black-box context for each round, maintaining low visibility while ensuring impact. Existing FL attack methods often show high visibility while maintaining impact due to their attack nature, the scale of the introduced perturbations, and the lack of detection strategies. Also, these methods often rely on maximizing cross-entropy loss on the true class, resulting in delayed attack convergence and highly noticeable perturbations. Hence, it is crucial to develop a stealthy data poisoning attack with low visibility for black-box settings in order to comprehend the motives of a cautious attacker in designing an FL attack. To address these challenges, we propose a Feedback-guided Causative Image Black-box Attack (F-CimBA), which is specifically designed for FL by adding random perturbation noise to the data. F-CimBA minimizes the loss of the most confused class (i.e., the incorrect class that the model confuses with the highest probability) instead of the true class, allowing it to exploit local model vulnerabilities for early attack convergence. This approach ensures that poisoned updates maintain low visibility, reducing the likelihood of server-side rejection. Furthermore, F-CimBA adapts effectively under non-IID data distributions and varying attack scenarios, consistently degrading the global model's performance. Additionally, we analyze its impact on system hardware metrics, highlighting the stealth and efficiency of F-CimBA, considering the computational overhead of repeated poisoning attempts in the FL context. Our evaluation demonstrates F-CimBA's consistent ability to poison the global model with minimal visibility under varying attack scenarios and non-IID data distributions, even in the presence of robust server-side defenses.

1 Introduction

Federated learning (FL) has emerged as a powerful machine learning paradigm because of its use in several applications, such as healthcare, signal processing, mobile user personalization, computer vision, etc. (Wang et al., 2023; Zeng et al., 2023). Despite the advantages of shared intelligence, the decentralized nature of FL makes it highly susceptible to adversarial attacks (Chen et al., 2022; Sun et al., 2023; Lewis et al., 2023). These attacks are known as poisoning attacks (Kumar et al., 2024a;b) and are further classified into (i) *Data poisoning,* where compromised clients only provide malicious data without interfering with the local model training (Shejwalkar et al., 2022). (ii) *Model poisoning,* where, compromised clients directly provide malicious updates (Baruch et al., 2019; Fang et al., 2020). In this paper, we focus on online untargeted black-box data poisoning attack as it is the most common and relevant to real-world FL deployments, as

outlined in the recent work by (Shejwalkar et al., 2022). In the context of black-box attacks, the key distinction between federated and centralized learning lies in the attacker's access to the global model's internal information. In centralized learning, attackers manipulate input data or query the model without access to its parameters. In contrast, attackers participate as malicious clients, indirectly influencing the global model through manipulated local updates in FL. Although the core concept remains similar, attack strategies and implications differ due to the distinct nature of these learning systems (Kumar et al., 2024b). Also, these black-box data poisoning attacks remain undetected for a longer duration of time among a larger population of FL clients. Specifically, our attacker is interested in generic misclassification (untargeted) rather than specific misclassification (targeted).

Limitations of existing attacks. Table 1 provides a summary and gaps in existing FL attacks, highlighting their properties, limitations, and broad scope of our proposed method. Existing recent works such as PGA (Shejwalkar et al., 2022), ADA (Sun et al., 2022), internal evasion, (Kim et al., 2023), etc, have exposed FL's high vulnerability to minor perturbations of adversarial attacks under the white-box setting. Executing a white-box attack in real-world scenarios becomes impractical, as it necessitates knowledge of internal model details, which is restricted due to multiple layers of model security. On the other hand, in a black-box adversarial attack, the adversary can create malicious data with minimal information about the model and subsequently train the local model on the poisoned data to generate a malicious update. After a few rounds of updates, the malicious attack in FL forces the global model at the server model to cause drastic misclassification with extremely high confidence (Shejwalkar et al., 2022). Moreover, the focus on black-box data poisoning attacks has predominantly been within centralized machine learning settings. There is a scarcity of research on untargeted black-box causative data poisoning attacks specifically tailored for FL environments, as explored in prior work (Mothukuri et al., 2021). Furthermore, malicious actors deliberately insert poisoned data into the client training datasets to induce incorrect predictions while evading detection, resulting in low attack visibility. This property is essential for ensuring covert execution, enabling attackers to achieve their goals without activating the server's detection mechanisms. Typically, attackers strive to maximize the impact of their attacks while maintaining minimal visibility. Therefore, considering these two factors is vital for understanding the adversary's approach to designing attacks in real-world scenarios. To address these challenges, we propose a robust poisoning attack strategy that achieves low visibility and high impact in causative black-box data poisoning scenarios.

Table 1: Comparison of F-CimBA with existing causative attacks across key dimensions. For each desired property, we indicate whether it is explicitly considered and optimized (✔) or not explicitly addressed (✗) in the design of various methods. B: black-box, W: white-box, DP: data poisoning, and MP: model poisoning.

Attack Method	Attack Type	Attack Setting	Attack Impact	Attack Visibility (Stealthiness)	Different non-IID Data Analysis	Attack Impact Analysis on Hardware Metrics
DPA-DLF (Shejwalkar et al., 2022)	DP	B	✔	✗	✗	✗
DPA-SLF (Shejwalkar et al., 2022)	DP	B	✔	✗	✗	✗
PGA (Shejwalkar et al., 2022)	MP	W	✔	✗	✗	✗
LIE (Baruch et al., 2019)	MP	W	✔	✔	✗	✗
DYN-OPT (Shejwalkar & Houmansadr, 2021)	MP	W	✔	✗	✗	✗
ADA (Sun et al., 2022)	MP	W	✔	✗	✗	✗
Edge-case backdoor (Wang et al., 2020)	DP, MP	W	✔	✗	✗	✗
Stealthy model poison (Bhagoji et al., 2019)	MP	W	✔	✔	✗	✗
(Bagdasaryan et al., 2020)	MP	W	✔	✗	✗	✗
(Fang et al., 2020)	MP	W	✔	✔	✔	✗
F-CimBA (ours), 2025	DP	B	✔	✔	✔	✔

In this paper, we introduce a causative black-box data poisoning adversarial attack on FL called **feedback-guided causative image black-box adversarial attack (F-CimBA)** using random perturbation noise. The proposed attack minimizes the loss of the most confused class, i.e., the target class the model confuses with the highest probability. In an untargeted attack, boosting the probability score of the most confused class provides advantages like increased misclassification rates, leveraging model weaknesses, and faster attack convergence. Additionally, our approach utilizes the probability score of the most confused class as feedback to iteratively refine adversarial perturbations that remain almost imperceptible to human observers.

We include two key attack settings in our threat model such that the attacker controls the data used in a single or multi-client attack. Further, the attacker has control of all the compromised clients' data and only their model predictions. Figure 1 shows the F-CimBA method on the client side. The poisoned local model update tries to disrupt the benign global model parameter values (red) by reducing its variance and dropping the test accuracy. We conduct extensive experiments using homogeneous and heterogeneous data shards with different numbers of clients. We present time-series information on scalability for attack metrics such as misprediction accuracy, global epochs for model convergence, and its effect on hardware performance metrics like central processing unit (CPU) and graphics processing unit (GPU) utilization. Furthermore, we conduct an extensive evaluation of F-CimBA's robustness in the presence of Byzantine-robust aggregation techniques such as Krum (Blanchard et al., 2017), trimmed mean (Yin et al., 2018), and median (Yin et al., 2018) at the server level. Also, we assess the performance of our method under the recent robust defense mechanisms like FLTrust (Cao et al., 2021), local malicious factor (LoMar) (Li et al., 2023), and FLDefender (Jebreel & Domingo-Ferrer, 2023) to comprehensively measure its significance and effectiveness.

Below are the main contributions of our work to the field of adversarial attacks in FL systems.

- We present a novel causative black-box data poisoning adversarial attack in federated learning, employing gradient-based perturbation from the perspective of a cautious attacker. Our approach involves crafting a stealthy attack, strategically balancing high attack impact with low attack visibility factors. Also, we introduce a feedback mechanism that utilizes the probability of the most confused class to update the adversarial perturbation towards achieving low attack visibility with high attack impact.

- We provide theoretical proof for the convergence, computational, and communication costs of our proposed method.

- We perform a comprehensive analysis to assess the potential implications of the proposed F-CimBA method for degrading the performance of system hardware performance metrics. These metrics encompass crucial parameters such as execution time, bandwidth, CPU percentage, GPU memory, GPU utilization, and GPU temperature. Also, we analyze the impact of non-independent and non-identically distributed (non-IID) data on our F-CimBA attack.

- We conduct a comprehensive analysis through extensive experiments involving four diverse datasets and three different models. Specifically, we utilize a convolutional neural network (CNN) model for the German traffic sign recognition benchmark (GTSRB) (Stallkamp et al., 2011) and the KUL Belgium traffic sign dataset (KBTS) (Mathias et al., 2013) datasets. Also, we use a residual network (ResNet18) for the Canadian institute for advanced research (CIFAR10) (Krizhevsky et al., 2009) dataset and the LeNet architecture for the extended modified national institute of standards and technology (EMNIST) (Cohen et al., 2017) dataset. Furthermore, our evaluation of F-CimBA extends across both homogeneous and heterogeneous FL data settings, exploring scenarios with varying total numbers of clients and attack percentages.

Paper outline: Section 2 presents related work of existing FL data poisoning and model poisoning attacks. Section 3 provides essential background information and threat model details. Section 4 provides the details of the F-CimBA algorithm. Section 5 presents the convergence analysis of F-CimBA. Section 6 gives the computational and communication cost analysis. Section 7 presents a comprehensive assessment of F-CimBA, covering different FL settings, attack scenarios, system configurations, and machine settings. Section 8 provides a detailed ablation study of our work. Section 9 discusses the paper's limitations and outlines future work. Finally, Section 10 concludes with key research findings and future research directions.

2 Related Work

This section provides a comprehensive overview of existing attacks in FL. As discussed in Section 1, adversarial attacks in FL are generally classified into two types: data poisoning attacks and model poisoning attacks.
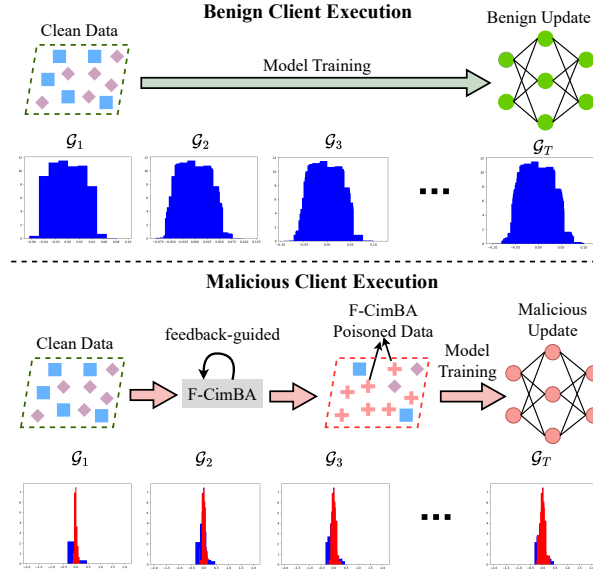
Figure 1: Attack overview: malicious clients poison training data by inducing feedback-guided perturbations to generate malicious updates. We show the attack effect on the global model ($\mathcal{G}$) weights for T communication rounds. Eventually, F-CimBA malicious update (red) reduces the variance of the benign global model weights (blue) and thereby drops the test accuracy.

Data Poisoning Attacks: Adversarial attacks on machine learning models and deep neural networks have gained significant attention (Vadillo et al., 2023; Rakhsha et al., 2021). While most research has focused on centralized ML (Yang et al., 2020), there has been limited exploration of untargeted black-box data poisoning attacks in FL environments. For instance, (Bagdasaryan et al., 2020) introduced a backdoor attack framework for FL, leveraging a constrain-and-scale technique to train on backdoor data and submit a compromised model update to the central server. Similarly, (Fang et al., 2020) treated labelflip attacks as optimization problems, applying them to Byzantine-robust FL methods. (Shejwalkar et al., 2022) proposed two variations of data poisoning attacks, static labelflip (DP-SLF) and dynamic labelflip (DP-DLF), to demonstrate the vulnerabilities in FL systems. Each of these approaches underscores specific risks in FL. For example, Bagdasaryan *et al.*'s(Bagdasaryan et al., 2020) method highlights the dangers of backdoor attacks, emphasizing the importance of robust countermeasures. (Zhang et al., 2020) showcased generative adversarial attacks' effectiveness in poisoning FL. Fang *et al.*(Fang et al., 2020) exposed the susceptibility of FL to label manipulation, aiding the development of Byzantine-robust techniques. Lastly, (Shejwalkar et al., 2022)'s work illustrates the threats posed by poisoned training data, stressing the critical need for efficient detection and defense mechanisms.

Model Poisoning Attacks: This category involves attackers directly submitting malicious updates (Bhagoji et al., 2019; 2018). Various strategies have been proposed to craft such updates effectively. For example, (Baruch et al., 2019) introduced the "little is enough" (LIE) attack, where noise is added to the average of benign updates using the standard deviation of these updates to generate poisoned updates. (Shejwalkar & Houmansadr, 2021) designed malicious model updates by perturbing the benign reference aggregate in a direction that maximally aligns with the attacker's intent. Similarly, (Fang et al., 2020) computed the average of benign updates, identified a fixed malicious direction, and generated poisoned updates by determining suboptimal parameters to bypass the aggregation rule. Each of these approaches reveals critical vulnerabilities in federated learning systems. Baruch *et al.*'s LIE attack demonstrates the impact of subtle noise injection into the aggregation process. Shejwalkar *et al.*'s technique highlights the ability to manipulate learning dynamics through targeted perturbations of the reference aggregate. Fang *et al.*'s strategy emphasizes how carefully crafted updates can evade aggregation rules and degrade the quality of the federated model.

Existing attack methods have several limitations, such as insufficient analysis of FL settings (both homogeneous and heterogeneous), a lack of realistic attack scenarios as highlighted by (Shejwalkar et al., 2022), and the computational overhead associated with GAN-based methods. Moreover, black-box adversarial attacks in FL using random perturbation noise remain largely unexplored. Our proposed approach addresses these limitations through a feedback-guided causative black box attack method in an FL setup with lower visibility. Additionally, we analyze system hardware performance metrics after the attack, a direction not previously explored by other works.

3 Preliminaries

This section provides the essential background information required to comprehend the remainder of the paper. A summary of the notations used throughout this work is presented in Table 2.

Table 2: Overview of notations.

Notation	Definition
n	Total number of clients
k	Client's index
m	Total number of malicious clients
T	Global epochs
E	Local epochs
$\mathcal{D}_k$	k^{th} client local data
$\tilde{\mathcal{D}}_k$	F-CimBA poisoned k^{th} client local data
$\mathcal{G}_{\theta_g}$	Global model with parameters θ_g
f_θ	Local model at epoch θ
$\mathcal{D}_{test}$	Test data at the server
η	Learning rate
μ	Noise coefficient
$\mathcal{B}$	Batch size
$\mathcal{B}_n$	Total batches
$\mathcal{L}$	Cross-entropy loss function
$\mathcal{A}$	Attacker client
λ_i	Weighted coefficient of FedAvg function
$\mathcal{U}$	Random perturbation noise

Definition 3.1 *(**FL setup.**) We consider a FL system consisting of a central server and n clients. Each client, denoted as $\mathcal{C}_k$, where $k \in [1, n]$, has its private local dataset $\mathcal{D}_k$, referred to as a shard. Specifically, $\mathcal{D}_k = \{(\mathcal{X}_{i,k}, \mathcal{Y}_{i,k})\}_{i=1}^{\mathcal{N}_k} \subseteq \mathbb{R}^d \times \mathbb{R}$, where $\mathcal{X}_{i,k}$ represents the input data and $\mathcal{Y}_{i,k}$ denotes the corresponding labels. For simplicity, we assume that $\|\mathcal{X}_{i,k}\|_2 = 1$ for all $i \in [1, \mathcal{N}_k]$ and $k \in [1, n]$, with the final component of each feature vector fixed at $\mathcal{X}^d = 1/2$, ensuring L_2-norm data normalization. The primary objective of FL is to collaboratively train a global model $\mathcal{G}_{\theta_g}$ (where $\mathcal{G}$ represents the model and θ_g denotes its parameters) that generalizes well on a global test dataset $\mathcal{D}_{test}$. During each training round t, the central server distributes the current global model parameters θ_g^t to all n clients. Upon receiving θ_g^t, each client $\mathcal{C}_k$ initializes its local model parameters θ_k^t with θ_g^t and trains the local model using its private dataset $\mathcal{D}_k$ with a batch size B and E local epochs. After local training, the updated local parameters are computed as: $\theta_k^t = \theta_k^t - \eta \nabla \mathcal{L}(\mathcal{X}_k, \mathcal{Y}_k)$, where η is the learning rate and $\mathcal{L}$ is the local cross-entropy loss. Each client then sends its local model parameters $\{\theta_k^t\}_{k=1}^n$ back to the server for aggregation. The central server aggregates these local models using the Federated Averaging (FedAvg) method (McMahan et al., 2017), which can be expressed as:*

$$\theta_g^{t+1} = \sum_{k=1}^n \lambda_k \theta_k^t, \tag{1}$$

where $\lambda_k = \frac{\mathcal{N}_k}{\sum \mathcal{N}_k}$ and $\sum_k \lambda_k = 1$. This iterative process is repeated until the global model achieves convergence, satisfying the following condition $\arg\min_{\theta_g} [\mathcal{L}(\theta_g) \triangleq \frac{1}{n} \sum_{k=1}^n \mathcal{L}(\theta_g, \mathcal{D}_k)]$.

Table 3: Key dimensions of our threat model and their attributes.

Objective			Knowledge & Capabilities		Attack Mode
Security violation	Attack specificity	Error specificity	Model	Data distribution	Consciously active
Availability: Misclassify test data and cause disruption to benign clients' objectives.	**Indiscriminate:** Misclassify all or most of the test inputs during inference.	**Untargeted:** Misclassify the give test data to any other class.	**Black-box:** Adversary cannot break into the compromised clients and cannot manipulate the model parameters.	The adversary can only access the local data distributed at the clients. **Note: The attack is agnostic to the type and degree of non-IID in the distributed data at the clients.**	**Online:** The adversary repeatedly and adaptively poisons the model based on the attack budget strategy.

We adopt an FL data shard setting with non-independent and non-identically distributed (non-IID) data. We achieve this setting by partitioning the training data using a Dirichlet distribution (Minka, 2000) with parameter β among clients. This sharding approach allows us to adeptly manage data heterogeneity, mirroring the nature of the real-world data distribution. Additional information on Dirichlet data distribution is provided in Section 7.

3.1 Feedback-guided Perturbation for Adversarial Sample Generation

Optimization typically requires guidance to find gradients that minimize the cross-entropy loss on the true class. In our case, we need to minimize the loss of the 'most confused class' (as defined in Section 1) by searching for the perturbation. Earlier works like Gressel *et al.* (Gressel et al., 2021) proposed a feature importance guided attack (FIGA) that perturbs the most important features of the input towards the target class. We extend this idea to propose a feedback-guided attack that searches for perturbations to minimize the loss of the most confused class.

3.2 Threat Model

In this paper, we adopt a threat model grounded in real-world FL production environments, where malicious clients inject poisoned data and train their local models using this compromised dataset. These clients operate under black-box constraints, meaning they cannot manipulate the training process, alter the server's aggregation algorithm, or interfere with server communications. However, they retain control over their local data and can access predictions generated by their local model on any selected input. Our threat model aligns with the framework proposed by (Shejwalkar et al., 2022), recognized as one of the most realistic and practical approaches for FL scenarios. Real-world scenarios relevant to our threat model include (i) healthcare applications where FL may be employed to train models on medical data from various hospitals. Malicious participants can introduce corrupted data to compromise the model's performance. (ii) In FL-based autonomous driving applications, compromised vehicles can provide manipulated data to the central server, affecting the overall model's performance. (iii) Fintech applications domain where financial institutions can collaborate using FL to detect fraudulent transactions. In such a scenario, a malicious client can poison the data to evade detection or reduce the overall model's performance. Based on this threat model, we present the critical dimensions of our threat model and the assumptions we make about the FL setup as shown in Table 3. The attacker's primary goal is to cause the global model, used for testing on the server, to misclassify most or all of the test data, thereby reducing its performance. This attack is untargeted, aiming for generic misclassification rather than specific, targeted misclassifications. The attacker operates under a black-box assumption regarding the server, with no access to the server's parameters, global model predictions, or aggregation process, and assumes the server is trustworthy and does not inspect model updates. On the client side, the attacker controls the data of one or more compromised clients, which generate updates based on trusted local training processes. However, the attacker cannot interfere with the training procedures or communication between the compromised clients and the server. It is limited to manipulating the clients' local data without affecting their training or server interaction. Further, we consider an active attacker employing repeat and adaptive data poisoning strategies on the compromised clients. These strategies utilize feedback from F-CimBA to sustain the attack throughout the FL process (online attack).

4   Proposed Approach

This section describes the details of our F-CimBA configured FL system.

Problem statement. In each round t, the server sends the global model parameters θ_g^t to all n clients, out of which m clients are malicious *s.t.* $m > 0$ & $m << n$. The adversarial client $\tilde{\mathcal{C}}_k$ introduces a μ-bounded poisoned data followed by one-pixel mutations into training dataset $\tilde{\mathcal{D}}_k$ such that $\tilde{\mathcal{D}}_k = \{\tilde{\mathcal{X}}_i, \mathcal{Y}_i\}_{i=1}^{\mathcal{N}_k}$, where $\mathcal{N}_k$ is the training data size, as defined in Definition 3.1.

Definition 4.1 (μ-**bounded adversary.**) *Let $\mathcal{F}$ be the adversary's function class. An adversarial pertur-bation $\mathcal{U}$ is characterized by the mapping $\mathcal{U} := \mathcal{F} \times \mathcal{X} \times \mathbb{R} \to \tilde{\mathcal{X}}$. For $\mu > 0$, we define the L_2 norm ball as $\mathcal{B}_2(\mathcal{X}, \mu) := \{\tilde{\mathcal{X}} \in \mathbb{R}^d : \|\tilde{\mathcal{X}} - \mathcal{X}\| \leq \mu\} \bigcap \mathcal{X}$. We classify the adversary $\mathcal{A}$ as μ-bounded if it adheres to the condition $\mathcal{U}(\mathcal{F}, \mathcal{X}_i, \mathcal{Y}_i)_{i=1}^{\mathcal{N}_k} \in \mathcal{B}(\mathcal{X}_i, \mu)_{i=1}^{\mathcal{N}_k}$. Furthermore, for a given $\mu > 0$, we denote the worst-case adversarial perturbation $(\mathcal{U}^+)$ as*

$$\mathcal{U}^+ := \underset{\tilde{\mathcal{X}} \in \mathcal{B}(\mathcal{X}, \mu)}{\arg\max} \mathcal{L}(f_\theta(\tilde{\mathcal{X}}_i, \mathcal{Y}_i)_{i=1}^{\mathcal{N}_k}),$$

where $\mathcal{L}$ and f_θ represent the cross-entropy loss function and parameters of the local black-box model, respectively.

The adversarial client $\tilde{\mathcal{C}}_k$ trains the local model using the poisoned data and sends malicious parameters $\tilde{\theta}_k^t$ to the server. The server aggregates the parameters sent by malicious and benign clients as per Eq. (1) in Definition 3.1 and performs testing. Let $\mathcal{A}_\mathcal{G}$ and $\mathcal{A}_\mathcal{G}^*$ be the global test accuracy achieved by the global model $\mathcal{G}_{\theta_g}$ before and after the attack, respectively. The problem statement from the attacker's perspective involves generating adversarial perturbed training data $\tilde{\mathcal{D}}_k$ with the following objectives:

1. Maximize the impact of the attack with a lower global test accuracy such that $\mathcal{A}_\mathcal{G}^* << \mathcal{A}_\mathcal{G}$.

2. Maximize the attack impact while ensuring minimum attack visibility V denoted as $V = |Val_\mathcal{G} - Val_{\tilde{\mathcal{C}}}|$, where $Val_\mathcal{G}$ and $Val_{\tilde{\mathcal{C}}}$ are the validation accuracy of global model and the local adversarial client model, respectively.

We outline the changes to the standard FL framework, both on the server and client sides, to incorporate the F-CimBA attack. A comprehensive flowchart of our method is shown in Figure 2. On the server side, no modifications are introduced since our focus is primarily on the data poisoning attacks executed at the client level. In each training epoch, the malicious client $\mathcal{C}_k$ receives the global model parameters θ_g^t from the server. These parameters are then used to train the local model θ_k^t on the adversarially poisoned data. This process enables the computation of the malicious model update $\hat{\theta}_k^t$.

Attack motivation. As detailed in Section 1, our research objective is to achieve maximum attack impact with minimal attack visibility under black-box attack settings. Limited prior work has studied untargeted black-box causative data poisoning attacks applicable to FL settings. Further, they often lack alignment with the attacker's perspective in generating a high-impact attack with low visibility. Therefore, the demand arises for a relatively straightforward yet effective poisoning attack that drastically affects the accuracy while evading detection (low visibility).

Proposed F-CimBA attack. To design a black-box online data poisoning attack, we propose a novel feedback method for our F-CimBA attack. Here, we minimize the loss of the most confused class. It is the incorrect class where the model misclassifies with the highest probability. Raising the most confused class probability score in an untargeted attack leads to (i) higher misclassification rates by raising the misclassified sample ratio and enhancing the confusion matrix. (ii) Faster attack convergence by rapidly adapting to the deliberately induced misclassifications, accelerating the attack process. (iii) More effective exploitation of model weaknesses creates an inverse relationship between vulnerability and accuracy. Our F-CimBA can only infer the most confused class score (i.e., the prediction probability) as feedback because of its black-box nature. The aim is to guide F-CimBA in increasing this misprediction probability with lower attack visibility by searching for perturbation noise.
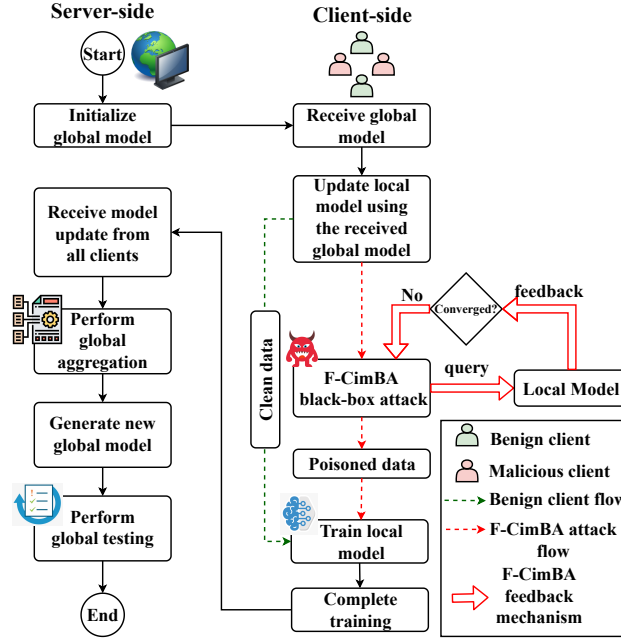
Figure 2: FL Flowchart with the proposed F-CimBA attack.

Let $\mathcal{X}$, $\mathcal{Y}$ be the image and label space, respectively, of the local client $\mathcal{C}_k$ such that $\mathcal{X} = \{\{\mathcal{X}_1\}_{i=1}^B, \ldots, \{\mathcal{X}_{B_n}\}_{i=1}^B\}$ and $\mathcal{Y} = \{\{\mathcal{Y}_1\}_{i=1}^B, \ldots, \{\mathcal{X}_{B_n}\}_{i=1}^B\}$, where each $\mathcal{X}_s$ is with a batch size of $\mathcal{B}$ and $\mathcal{B}_n$ is the total number of batches in $\mathcal{D}_k$. Since the gradient information is not available for the black-box model f_θ, the output probabilities act as an effective proxy to direct the search for the perturbation that generates the adversarial image. To implement this, we do not compute or estimate gradients explicitly. Instead, we adopt a feedback-driven, uniform noise-based perturbation strategy. In this untargeted attack scenario, the goal is to increase the probability of the most confused class. Given a clean image $\mathcal{X}_{l,i}$ and its corresponding label $\mathcal{Y}_{l,i}$, the F-CimBA adversarial client generates a random perturbation noise $\mathcal{U}$ sampled from uniform distribution $\mathcal{U} \sim \text{Uniform}(-1, 1)^{\dim(\mathcal{X}_{l,i})}$ with dimension same as original clean image. It is then scaled by a noise coefficient μ, such that the prediction $f_\theta(\mathcal{X}_s + \mu \times \mathcal{U})$ does not match the label $\mathcal{Y}_s$, as depicted in Algorithm 1 and Figure 3.

The final adversarial image $\tilde{\mathcal{X}}_s$ is computed by adding the perturbation to the original image, i.e., $\tilde{\mathcal{X}}_s = \mathcal{X}_s + \mu \times \mathcal{U}$. The adversary uses the local black-box model on $\tilde{\mathcal{X}}_s$ to calculate the most confused class score (feedback to the F-CimBA), as feedback $= \max_{\tilde{\mathcal{Y}}_s \neq \mathcal{Y}_s} \{P(\tilde{\mathcal{Y}}_s | \tilde{\mathcal{X}}_s)\}$, where $\mathcal{Y}_s$ & $\tilde{\mathcal{Y}}_s$ are original & predicted classes, respectively. Here, $\tilde{\mathcal{Y}}_s$ is not a fixed constant and it refers to the predicted label for the perturbed input $\tilde{\mathcal{X}}_s$ at each iteration. This value changes dynamically as $\tilde{\mathcal{X}}_s$ is updated. Further, the feedback expression $\max_{\tilde{\mathcal{Y}}_s \neq \mathcal{Y}_s} P(\tilde{\mathcal{Y}}_s | \tilde{\mathcal{X}}_s)$ calculates the confidence score of the most probable incorrect class at that point. For the initial iteration, the random perturbation noise ($\mathcal{U}$) is added in the positive direction. The noise is added in the negative direction for further iterations. It is changed to a new random perturbation noise sampled from the uniform distribution $\mathcal{U} \sim \text{Uniform}(-1, 1)^{\dim(\mathcal{X}_{l,i})}$ in the subsequent iterations. Further, it is important to note that the mention of "positive" and "negative" directions refers to the sign of the perturbation applied in each iteration relative to the prior feedback and not to any true gradient-derived direction. Importantly, this avoids explicit gradient computation and makes F-CimBA both lightweight and compatible with the strict black-box threat model, where only output logits or class probabilities can be queried. This process of updating the random perturbation noise $\mathcal{U}$ is repeated until the feedback converges and the algorithm generates the final adversarial image $\tilde{\mathcal{X}}_s$ and predicted label $\tilde{\mathcal{Y}}_s$.

Our approach aligns with prior black-box attack strategies, such as Bandits with Priors (Ilyas et al., 2018), which performs gradient-free query-based optimization. This approach allows adversaries to estimate gradient-like information without access to true model gradients. In our case, instead of constructing full finite-difference gradient estimates (which can be query-expensive in high dimensions), we use randomly

sampled directions and rely on the feedback from the model's prediction probabilities to guide iterative search, thus remaining within practical black-box constraints.

In addition, F-CimBA converges on the $L2$ norm such that $||\tilde{\mathcal{X}} - \mathcal{X}|| \leq \mu$. The noise coefficient parameter μ controls the deviation of the adversarial image with respect to the original image without, making it perceivable to the human eye (helps avoid filtering images via human inspection). This step returns the final adversarial image to the client's dataset for further local model training via trusted execution environments (TEE) (Mondal et al., 2021; Chen et al., 2020). The entire F-CimBA algorithm is shown in Algorithm 1. In this work, we cannot directly compare the local model parameters due to the black-box attack settings. Therefore, we utilize the validation accuracy as a proxy metric to evaluate the visibility performance of the local models.

Algorithm 1 Proposed F-CimBA attack method integrated FL process

---

Input: Initialized global model parameters (θ_g^{init}), local client data ($\{\mathcal{D}_k\}_{k=1}^n$), noise coefficient ($\mu$)
Output: Global test accuracy after attack ($\mathcal{A}_{\mathcal{G}}^*$)

1: **function** CLIENTEXECUTION(θ_g^t)
2:     **for** $k = 1$ to n **do**   ▷ *Loop through n clients*
3:         **if** ClientType[k] $== \mathcal{A}$ **then**
4:             Initialize local model: $\tilde{\theta}_k^t \leftarrow \theta_g^t$ ▷ *Initialize local model with shared global model*
5:             **for** each sample s in $\mathcal{D}_k$ **do**
6:                 Sample uniform perturbation: $\mathcal{U} \sim \text{Uniform}(-1, 1)^{\dim(\mathcal{X}_s)}$
7:                 Compute perturbed input: $\tilde{\mathcal{X}}_s^{(0)} = \mathcal{X}_s + \mu \times \mathcal{U}$ ▷ *Initial adversarial image*
8:                 Predict: $\tilde{\mathcal{Y}}_s^{(0)} = f_\theta(\tilde{\mathcal{X}}_s^{(0)})$
9:                 Compute feedback: $\text{feedback}_{old} = \max_{\tilde{\mathcal{Y}}_s \neq \mathcal{Y}_s}\{P(\tilde{\mathcal{Y}}_s|\tilde{\mathcal{X}}_s^{(0)})\}$
10:                Set $\text{feedback}_{new} \leftarrow \text{feedback}_{old}$, $\texttt{iter} \leftarrow$ max iterations, $j \leftarrow 0$
11:                **while** $\text{feedback}_{new} \leq \text{feedback}_{old}$ and $\texttt{iter} \neq 0$ **do**
12:                    **if** $j == 0$ **then**
13:                        retain $\mathcal{U}$ (positive direction)
14:                    **else if** $j == 1$ **then**
15:                        $\mathcal{U} \leftarrow -\mathcal{U}$ (reverse direction)
16:                    **else**
17:                        Sample new $\mathcal{U} \sim \text{Uniform}(-1, 1)^{\dim(\mathcal{X}_s)}$
18:                    $\tilde{\mathcal{X}}_s^{(j+1)} = \mathcal{X}_s + \mu \times \mathcal{U}$
19:                    $\tilde{\mathcal{Y}}_s^{(j+1)} = f_\theta(\tilde{\mathcal{X}}_s^{(j+1)})$
20:                    $\text{feedback}_{new} = \max_{\tilde{\mathcal{Y}}_s \neq \mathcal{Y}_s}\{P(\tilde{\mathcal{Y}}_s|\tilde{\mathcal{X}}_s^{(j+1)})\}$
21:                    $\texttt{iter} \leftarrow \texttt{iter} - 1$, $j \leftarrow j + 1$
22:                Append $(\tilde{\mathcal{X}}_s^{(j)}, \tilde{\mathcal{Y}}_s^{(j)})$ to $\tilde{\mathcal{D}}_k$
23:            $\tilde{\theta}_k^t \leftarrow \tilde{\theta}_k^t - \eta \nabla \mathcal{L}(\tilde{f}_{\theta,k}^t(\tilde{\mathcal{D}}_k))$
24:         **else**
25:            $\theta_k^t \leftarrow \theta_g^t$
26:            Train $f_{\theta,k}^t$ on $\mathcal{D}_k$ for E epochs
27:            $\theta_k^t \leftarrow \theta_k^t - \eta \nabla \mathcal{L}(f_{\theta,k}^t(\mathcal{D}_k))$
28:            $\theta_k^t \leftarrow \lambda_k \cdot \theta_k^t$
29:     **return** $\{\tilde{\theta}_k^t\}_{k=1}^m \cup \{\theta_k^t\}_{k=m+1}^n$
30: **function** MAIN( )
31:     Initialize: $\theta_g^1 = \theta_g^{\text{init}}$ ▷ *Initialize global model*
32:     **for** $t = 1$ to T **do**   ▷ *Loop through T global epochs*
33:         $\{\theta_k^t\}_{k=1}^n \leftarrow$ CLIENTEXECUTION(θ_g^t)
34:         $\mathcal{G}_{\theta_g}^{t+1} \leftarrow$ FedAvg of θ_k^t ▷ *Perform FedAvg as per Eq. 1*
35:         Evaluate global model: $\mathcal{A}_{\mathcal{G}}^* \leftarrow \texttt{Test}(\mathcal{G}_{\theta_g}^{t+1}, \mathcal{D}_{test})$ ▷ *Perform global model testing*
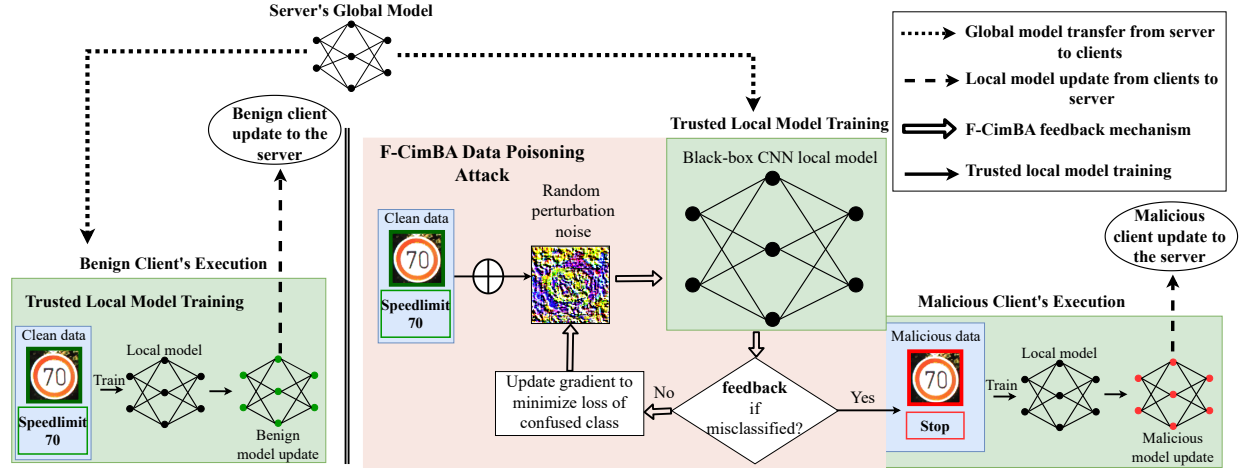36:   **return** $\max(\mathcal{A}_{\mathcal{G}}^*)$

Figure 3: Proposed pipeline: The local training on clients happens in a trusted and isolated container where the attacker has no control over the local model and hence black-box. Benign clients train on clean data to generate regular updates, whereas malicious clients poison the training data using feedback-guided F-CimBA attack to generate malicious updates.

5   Convergence Analysis of F-CimBA

In this section, we analyze the convergence behavior of FL under the influence of the proposed F-CimBA attack. While the attack introduces small perturbations to the input space (bounded by a noise coefficient μ), it indirectly affects the evolution of model parameters across communication rounds. Thus, bounding the global loss requires careful analysis of how such input perturbations impact the local and global model updates.

To this end, we adopt a theoretical framework based on the following components:

- We assume Lipschitz continuity of the model function and the loss function with respect to their inputs and parameters.

- We formally characterize how bounded adversarial perturbations in the input lead to bounded deviations in local model updates during client training.

- We analyze the deviation of the global model parameters under the FedAvg aggregation when a bounded number of clients are adversarial.

- We then bound the resulting change in the global loss function under standard stability assumptions.

This analysis builds on techniques adapted from prior work (e.g., Theorem 2 in (Li et al., 2024)). It shows that the effect of small-magnitude but persistent input perturbations introduced by F-CimBA can accumulate over time and lead to a quantifiable degradation in global model performance. The theorem below formalizes these insights.

Theorem 5.1 (Convergence of F-CimBA under μ-bounded perturbations) *Let $f_{\theta_g^t}$ denote the global model at round t with parameters θ_g^t in a federated learning setup with n clients, of which m are adversarial. We assume the following*

- *The local model f_θ is β-Lipschitz with respect to the input x.*

- *The loss function $\mathcal{L}$ is α-Lipschitz with respect to the model output.*

- *Each adversarial client perturbs input x to $\tilde{x}$ such that $\|x - \tilde{x}\| \leq \mu$.*

- *The client update function is γ-Lipschitz with respect to the input dataset in L_2 norm.*

Then, the average increase in global test loss over T rounds due to m malicious clients performing μ-bounded attacks is upper bounded as:

$$\frac{1}{T}\sum_{t=1}^{T}\left[\mathcal{L}(f_{\theta_g^t}, \mathcal{D}_{test}) - \mathcal{L}(f_{\theta_g^{t,clean}}, \mathcal{D}_{test})\right] \leq \delta,$$

where $\delta = \lambda m \cdot \alpha(\beta + \gamma)\mu$ and $\lambda = \frac{1}{n}$.

Proof. We decompose the impact of poisoned inputs into three effects.

Step 1: Effect of input perturbation on the model output

For any poisoned input $\tilde{x}$ generated by malicious client k

- Since f_θ is β-Lipschitz in input: $\|f_\theta(\tilde{x}) - f_\theta(x)\| \leq \beta\|\tilde{x} - x\| \leq \beta\mu$

- Since $\mathcal{L}$ is α-Lipschitz in model output: $|\mathcal{L}(f_\theta(\tilde{x}), y) - \mathcal{L}(f_\theta(x), y)| \leq \alpha\beta\mu$

This step bounds the local loss change per sample due to input perturbation.

Step 2: Parameter shift due to poisoned dataset

Let $\theta_k^{t,\text{clean}}$ and $\tilde{\theta}_k^t$ be the client updates trained on clean and poisoned data, respectively.

- Assume the update function is γ-Lipschitz w.r.t. data perturbations (supported by Theorem2 (Li et al., 2024)): $\|\tilde{\theta}_k^t - \theta_k^{t,\text{clean}}\| \leq \gamma\mu$

- The change in model output due to parameter drift is given as: $\|f_{\tilde{\theta}_k^t}(x) - f_{\theta_k^{t,\text{clean}}}(x)\| \leq \gamma\mu$

- Hence, the loss difference is bounded as: $|\mathcal{L}(f_{\tilde{\theta}_k^t}(x), y) - \mathcal{L}(f_{\theta_k^{t,\text{clean}}}(x), y)| \leq \alpha\gamma\mu$

Step 3: FedAvg aggregation effect

- Global model parameters update at round t as: $\theta_g^t = \sum_{k=1}^{n} \lambda_k \theta_k^t = \sum_{k\in\mathcal{B}} \lambda_k \theta_k^t + \sum_{k\in\mathcal{A}} \lambda_k \tilde{\theta}_k^t$.

- Define clean global model parameters update (no attack) as: $\theta_g^{t,\text{clean}} = \sum_{k=1}^{n} \lambda_k \theta_k^{t,\text{clean}}$

- Then, the total global model parameters shift is: $\|\theta_g^t - \theta_g^{t,\text{clean}}\| \leq \sum_{k\in\mathcal{A}} \lambda_k \|\tilde{\theta}_k^t - \theta_k^{t,\text{clean}}\| \leq m \cdot \lambda \cdot \gamma\mu$

- Loss deviation due to global model perturbation: $|\mathcal{L}(f_{\theta_g^t}, x) - \mathcal{L}(f_{\theta_g^{t,\text{clean}}}, x)| \leq \alpha \cdot m \cdot \lambda \cdot \gamma\mu$

Combining total effects

- Adding this to the previous loss deviation due to input perturbation ($\alpha\beta\mu$), we get: $\delta = \lambda m(\alpha\beta\mu + \alpha\gamma\mu) = \lambda m.\alpha(\beta + \gamma)\mu$

- Including input-to-output impact and model update impact for T rounds:

$$\frac{1}{T}\sum_{t=1}^{T}\left[\mathcal{L}(f_{\theta_g^t}, \mathcal{D}_{\text{test}}) - \mathcal{L}(f_{\theta_g^{t,\text{clean}}}, \mathcal{D}_{\text{test}})\right] \leq \delta$$

6 Computational and Communication cost

The computational and communication costs of the F-CimBA-integrated FL system are discussed below.

Lemma 6.1 *The expected time complexity of our proposed F-CimBA method is linear, specifically $\mathcal{O}(min(\tau, \gamma))$, where τ signifies the maximum iterations and γ represents iterations required to find a solution ($\gamma <= \tau$).*

Proof. The proposed F-CimBA method generates an adversarial image through a novel query-based approach in a maximum of τ iterations. Hence, the notation $\mathcal{O}(min(\tau, \gamma))$ indicates that the complexity depends on the minimum of τ and γ values. If γ is smaller, a solution is found before completing all iterations ($\mathcal{O}(\gamma)$). If γ equals τ, the complexity is $\mathcal{O}(\tau)$, considering all iterations. Notably, F-CimBA time complexity is based on feedback, causing the variable time complexity $\mathcal{O}(min(\tau, \gamma))$.

The training process on the F-CimBA adversarial client takes approximately 10 seconds, with the duration varying based on the batch size in the training dataset. This procedure employs a single Nvidia Tesla M60 GPU with 8GB of RAM, using around 3.2GB, which highlights the low computational overhead.

Communication cost- As described in Algorithm 1, the malicious client communicates only model updates to the server in each round, in line with the standard FL system process. Therefore, incorporating F-CimBA into the existing FL framework does not introduce significant additional communication overhead.

7 Experiments

In this section, we evaluate F-CimBA and compare its performance against state-of-the-art methods. Specifically, we address the following four major points to guide our evaluation:

1. Effectiveness of our F-CimBA method compared to state-of-the-art attacks.

2. F-CimBA's effectiveness under different degrees of non-IID data distribution.

3. Robustness of F-CimBA against inherent defense mechanisms, including Byzantine robust aggregation techniques and recent server-side defense methods.

4. Effectiveness of our F-CimBA method in minimizing attack visibility compared to other attack strategies.

7.1 Datasets and Metrics

We use the following four standard classification datasets and CNN models to demonstrate the diversity for our evaluation. Table 4 presents comprehensive experimental details of our evaluation.

- *GTSRB (Stallkamp et al., 2011).* is a well-known benchmark dataset for traffic sign classification. It consists of 43 traffic sign classes with 39209 samples. We build a custom 4-layer CNN architecture followed by two fully connected layers, and treat this as a global model for this dataset. It takes an input image of size 150×150. Unless specified otherwise, we consider 40 total clients and select all client updates at every communication round.

- *KBTS (Mathias et al., 2013).* is another well-known benchmark dataset for traffic sign classification. It consists of 62 traffic sign classes with 6978 samples. We use the earlier custom 4-layer CNN architecture as a global model for this dataset. Similar to the above dataset, we consider 40 total clients and select all client updates at every communication round.

- *CIFAR10 (Krizhevsky et al., 2009).* This well-known benchmark dataset for classification contains 60,000 samples with ten different classes, namely, aeroplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks. We use the ResNet18 (He et al., 2015) architecture with input size as 224×224. We consider 100 total clients, out of which 40 client updates are randomly selected at every communication round.

- *EMNIST (Cohen et al., 2017).* This is another benchmark dataset of 671,585 samples of handwritten characters & digits with 62 classes, including upper and lowercase handwritten characters. We use the LeNet (LeCun et al., 1998) architecture that takes an input of size 32×32. Further, we consider 10000 total clients, out of which 100 client updates are randomly selected at every communication round.

Table 4: Detailed experimental setup: including datasets, models, federated learning configuration, and attack parameters.

Dataset(s)	GTSRB (Stallkamp et al., 2011) KBTS (Mathias et al., 2013)	CIFAR10 (Krizhevsky et al., 2009)	EMNIST (Cohen et al., 2017)
Model	4 layered CNN input (150 x 150 RGB images) conv2d_64; kernel 5; stride 1 conv2d_128; kernel 3; stride 1 conv2d_256; kernel 1; stride 1 conv2d_256; kernel 1; stride 1 Fully connected layer 1 Fully connected layer 2 Softmax	ResNet18 (He et al., 2015)	LeNet (LeCun et al., 1998)
Total clients (n)	40	100	10000
Number of clients selected per round (k)	20, 40	40	100
Total malicious clients (m)	1, 8, 12, 16, 20, and 32	1, 10	1, 10
Amount of non-IID (β)	0.1, 0.5, 1 (default), 5, 10	0.1, 0.5, 1 (default), 5, 10	0.1, 0.5, 1 (default), 5, 10
Batch size (B)	64	64	64
Global communication rounds (T)	200	200	200
Local epochs (E)	5	5	5

We split all datasets into training ($\mathbb{D}_{Train}$, 80%) and test ($\mathbb{D}_{Test}$, 20%) sets. We further divide each of the clients' data into non-IID parts. Specifically, we split the training set using the Dirichlet distribution (Minka, 2000) with parameter $\beta = 1$ (default). Later, we evaluated our method with five different $\beta = [0.1, 0.5, 1, 5, 10]$ values to justify its robustness. Increasing the Dirichlet distribution parameter, β, generates more dense IID data splits among the clients. Finally, the test set is used on the server side to compute the global model accuracy. We use two attacker settings, namely *single-client* and *multi-client*. We set one client as malicious and others as benign for a single-client attack scenario. To handle randomness, we use single-attack scenarios by different malicious clients, namely, C_1, C_3, C_5, C_7, and C_9. In multi-client attack settings, we consider five different percentages of adversaries (volume of attack), {20%, 30%, 40%, 50%, 80%}, i.e., {8, 12, 16, 20, 32} randomly chosen clients as malicious for GTSRB and KBTS datasets evaluation. We set 10 randomly chosen clients as malicious for the CIFAR10 and EMNIST datasets. We did not try to increase the number of malicious clients further, as that would increase the number of malicious updates and, thereby, attack visibility and dilute the motivation of this work. We empirically set 1000 maximum attack iterations and initialized $\mu = 0.5$ in our experiments. We conducted each experiment three times and presented the results and graphs as the average outcomes of these three simulations with an error bar.

Attack baselines. We have thoughtfully selected state-of-the-art attack methods based on their relevance and uptodatedness. These methods have been categorized into three distinct groups for our analysis: (i) recent FL data poisoning attacks, including data poisoning attack-static label flip (DPA-SLF) (Shejwalkar et al., 2022) and data poisoning attack-dynamic label flip (DPA-DLF) (Shejwalkar et al., 2022), (ii) recent FL model poisoning attacks, including little is enough (LIE) (Baruch et al., 2019) and Fang (Fang et al., 2020), and (iii) robust ML data poisoning attack, such as simple black-box attack (SimBA) (Guo et al., 2019). This categorization enables us to thoroughly assess the current landscape of state-of-the-art methods in this field.

Metrics. We use the maximum classification global test accuracy ($GTA \in [0, 100]$) for all global epochs as an evaluation metric. It is defined as $GTA = \frac{TP+TN}{|\mathbb{D}_{Test}|} \times 100$, where TP, TN are true positives and true negatives given by the aggregated global model. The effectiveness of an attack is gauged by its capacity to substantially reduce the GTA, all while remaining undetected by the server.

7.2   F-CimBA Performance Comparison with Existing Attacks

No Attack and No Defense Case: We established a baseline accuracy for our FL configuration by evaluating it without any attacks or defense mechanisms for datasets such as GTSRB, KBTS, CIFAR10, and EMNIST. The baseline global test accuracy (GTA) ranged from 83.45% to 91.12%, as shown in Table 5 for non-IID data distribution with $\beta = 1$ as the default value. Notably, EMNIST exhibited slightly lower performance due to our FL setting, where data is distributed among 10,000 clients, and the server aggregates updates from only 100 randomly chosen clients. This results in reduced GTA due to low data availability at the clients and a high bias in the generated global model updates. Further insights into F-CimBA performance across varying degrees of non-IID are elaborated in subsequent sections.

Table 5: GTA (%) comparison for four datasets under non-IID distribution for no attack and no defense settings.

non-IID (β)	GTSRB	KBTS	CIFAR10	EMNIST
0.1	87.26±0.83	87.62±0.24	81.36±0.20	79.65±0.27
0.5	88.12±0.38	89.14±0.54	85.34±0.38	82.67±0.63
1 (default)	89.98±0.85	91.12±0.10	90.85±0.67	83.45±0.57
5	90.23±0.12	92.23±0.45	91.123±0.43	84.12±0.40
10	91.26±0.72	93.42±0.21	92.12±0.64	84.28±0.84

F-CimBA Attack and No Defense Case: We provide dataset-wise condensed highlights of our experimental results due to space constraints.

(i) GTSRB dataset. In the experiment with the GTSRB dataset, we include all 40 client updates for aggregation (Table 4) and later experiment on partial client participation in the ablation studies. We observe that our F-CimBA method consistently outperformed other baselines with a substantial GTA drop, approximately 19% drop under a single malicious client and 83% drop with 32 malicious clients, compared to existing methods, as shown in Table 6 and Table 7. Notably, even with only one malicious client, F-CimBA outperforms LIE and Fang methods, which are the second-best methods. This highlights F-CimBA's efficacy in inducing poisoned data, causing a significant drop in global test accuracy. Additional insights into F-CimBA's impact on various malicious client IDs are in Figure 4a. F-CimBA's feedback-guided approach efficiently identifies perturbations capable of confusing the global model with any of the non-targeted classes. Particularly in challenging scenarios, with an 80% attack percentage and 32 out of 40 clients exhibiting malicious behaviour, F-CimBA consistently achieves remarkable accuracy reductions of less than 10%, as illustrated in Figure 4c.

(ii) KBTS dataset. Similar to the GTSRB dataset, F-CimBA consistently outperforms baselines for the KBTS dataset. It achieves a significant GTA drop of approximately 10% under a single malicious client and an impressive 85% drop with 32 malicious clients, surpassing existing methods (Table 6 and Table 7). F-CimBA's effectiveness is slightly higher on GTSRB than KBTS in the single-client attack (Figure 4b), attributed to GTSRB's higher intra-class variability. However, F-CimBA accuracy drops in multi-client attacks are more pronounced for KBTS than GTSRB (Figure 4d), likely due to KBTS's broader class space. This enables feedback-guided F-CimBA to collaboratively compromise the global model non-targeted, generating potent perturbations with less than 10% GTA reductions for multi-client attack settings.

(iii) CIFAR10 dataset. The server employs a random client selection process, aggregating from only 40 out of 100 available client updates in each round, as detailed in Table 4. This results in the inclusion of malicious client updates in 81 out of 200 global epochs for the single-client attack setting and in all 200 epochs for the multi-client attack scenario. Building on the insights gained from our experiments with the GTSRB dataset, the F-CimBA method consistently exhibits superior performance compared to other baseline methods. In the single-client attack setting, it leads to an approximate GTA drop of 10%, while in the ten-client attack setting, the GTA drop is approximately 23%, outperforming other methods as presented in Table 6 and Table 8. However, it's worth noting that the lower rate of malicious client selection (81 out of 200 epochs) in the single-client attack setting, compared to the GTSRB dataset, contributes to a relatively lower performance of F-CimBA. Furthermore, in the multi-client attack scenario, where there are ten malicious clients on average,

only 4 of these updates are selected on average for aggregation due to the random selection process. Despite this constraint, the F-CimBA attack still achieves a significant GTA drop of approximately 23%, surpassing the second-best method. This success can be attributed to the influence of the 30 benign client updates, which help mitigate the impact of the malicious updates introduced by F-CimBA.

(iv) EMNIST dataset. The server employs a random client selection process, aggregating from only 100 out of 10,000 available client updates in each round, as detailed in Table 4. This results in the inclusion of malicious client updates in 45 out of 200 global epochs for the single-client attack setting and 59 out of 200 for the ten-client attack scenario. The EMNIST dataset setting poses a particularly challenging scenario, characterized by a lower probability of selecting malicious clients in both the single and multi-client attack settings. However, our F-CimBA method consistently demonstrates superior performance, showcasing a Global Test Accuracy (GTA) drop of approximately 6% and 8% less than the no-attack baseline in the respective attack scenarios. Notably, in the EMNIST dataset, where data is distributed across 10,000 clients with a non-IID parameter ($\beta = 1$), each client receives only around 150 samples, presenting additional constraints. Despite these challenges, our F-CimBA method surpasses other approaches, emphasizing its high-impact nature.

In summary, our thorough assessment of F-CimBA, considering various n and k values across four diverse datasets, underscores its remarkable effectiveness compared to other baseline methods. Notably, even when faced with a low rate of malicious client updates, F-CimBA consistently outperforms all other baselines. This effect is particularly significant in scenarios involving a single malicious client, as F-CimBA exhibits a substantial impact, further highlighting its robustness as the adversary's knowledge increases and F-CimBA clients become more prevalent.

Table 6: Comparison of GTA (%) with existing attacks for four datasets under a single-client attack setting. **Bold** result indicates the best result (lower is better) for an attack setting. F-CimBA consistently showed better accuracy drops across all datasets.

Attack	GTSRB	KBTS	CIFAR10	EMNIST
DPA-DLF	73.19±1.36	82.18±1.81	80.91±2.01	80.94±7.81
DPA-SLF	74.42±0.87	84.01±2.01	81.49±2.33	80.91±1.13
LIE	71.14±1.27	81.24±1.84	81.01±1.62	80.02±0.67
Fang	71.35±1.24	82.45±1.48	80.04±1.77	78.89±1.59
SimBA	72.36±1.45	83.55±1.57	80.94±2.28	81.95±1.09
F-CimBA (Ours)	**70.13±1.12**	**81.60±1.63**	**79.62±1.13**	**77.23±1.42**

Table 7: Comparison of GTA (%) with existing attacks for two datasets under multi-client attack setting. **Bold** result indicates the best result (lower is better) for an attack setting. F-CimBA consistently showed better accuracy drops across all datasets.

Attack percentage (%)- m/n	GTSRB						KBTS					
	DPA-DLF	DPA-SLF	LIE	Fang	SimBA	F-CimBA (ours)	DPA-DLF	DPA-SLF	LIE	Fang	SimBA	F-CimBA (ours)
20- 8/40	75.93±2.04	76.29±0.31	73.62±2.24	72.38±1.94	75.80±0.74	**70.81±0.26**	82.98±1.34	84.25±0.79	61.36±0.15	55.48±0.76	62.41±0.24	**51.75±1.30**
30- 12/40	75.04±2.49	76.79±1.37	70.26±0.80	69.34±1.13	74.41±2.03	**66.60±1.88**	83.47±1.62	80.29±0.81	44.42±0.97	43.36±1.79	45.76±1.71	**41.38±1.63**
40- 16/40	73.70±2.17	76.62±0.30	68.42±1.33	67.26±1.07	70.49±1.89	**63.35±0.39**	42.37±1.99	48.36±0.12	38.42±0.88	39.38±2.34	39.76±1.84	**35.43±0.49**
50- 20/40	68.54±0.58	74.58±2.18	56.34±0.84	59.48±2.07	63.46±0.85	**51.32±1.77**	38.06±2.32	39.49±1.19	30.42±1.64	30.36±0.25	31.24±2.19	**28.26±1.17**
80- 32/40	37.46±0.92	55.30±0.95	31.45±0.69	27.16±1.82	40.43±0.54	**6.38±0.55**	28.24±0.42	25.36±2.41	12.64±1.60	11.12±1.24	15.32±0.17	**6.85±0.40**

7.3 F-CimBA Performance Across Varying Degrees of non-IID

The influence of varying non-IID data distribution is a critical aspect that warrants further exploration. This examination allows us to better understand the interplay between the Dirichlet distribution (Minka, 2000) parameter β and the resulting data distribution characteristics. The relationship between β and the sample data partition is pivotal in comprehending the behavior of our experimental setup.

The Dirichlet distribution (Minka, 2000) serves as a key probabilistic model in Federated Learning (FL) to capture how data is distributed across multiple clients. This distribution is governed by a parameter β,

Table 8: Comparison of GTA (%) with existing attacks for two datasets under multi-client (10A) attack setting. **Bold** result indicates the best result (lower is better) for an attack setting. F-CimBA consistently showed better accuracy drops across all datasets.

Attack	CIFAR10	EMNIST
DPA-DLF	76.96±0.86	79.73±1.35
DPA-SLF	74.64±1.94	78.10±0.24
LIE	69.04±1.34	77.70±0.56
Fang	70.89±1.12	78.03±1.13
SimBA	72.63±1.21	79.90±0.83
F-CimBA (ours)	**67.34±0.62**	**75.40±0.87**

(a) Single-client - GTSRB   (b) Single-client - KBTS   (c) Multi-client - GTSRB   (d) Multi-client - KBTS
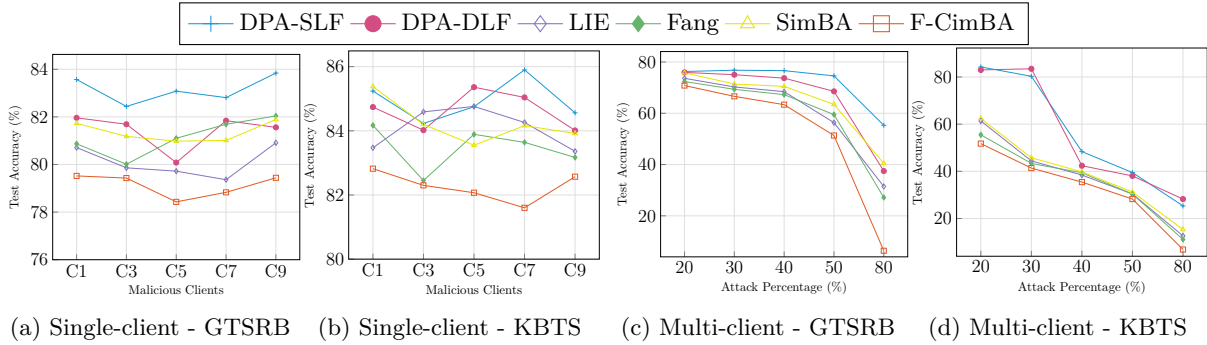
Figure 4: Effectiveness on F-CimBA against existing causative attacks: Comparison of GTA (%) for various baselines across two datasets and attack settings.

which plays a crucial role in controlling the level of non-IIDness (non-independent and identically distributed nature) in the data partitioning process. The Dirichlet distribution generates data allocations for clients based on their specific characteristics, and the probability density function is given by: $p(x_1, x_2, \ldots, x_K | \beta) = \frac{1}{B(\beta)} \prod_{i=1}^{K} x_i^{\beta_i - 1}$, where $x_1, x_2, \ldots, x_K$ denote the data proportions assigned to each client, and K represents the number of classes. The parameter vector $\beta = (\beta_1, \beta_2, \ldots, \beta_K)$ influences the distribution shape. In our context, we consider the case where all β_i are identical, resulting in a symmetric Dirichlet distribution. The normalizing constant $B(\beta)$ is the multivariate Beta function, which ensures that the distribution is properly normalized across the simplex defined by the proportions. The Beta function is defined as: $B(\beta) = \frac{\prod_{i=1}^{K} \Gamma(\beta_i)}{\Gamma\left(\sum_{i=1}^{K} \beta_i\right)}$, where $\Gamma(\cdot)$ is the Gamma function. By adjusting the values of β, the distribution can either promote IID behavior, where data is evenly distributed across clients, or induce varying levels of non-IIDness. Specifically, larger values of β tend to lead to a more even distribution of data across clients, while smaller values of β create more skewed data allocations, amplifying heterogeneity among clients. This fine-tuning of β is vital in FL systems, as it enables the model to reflect the real-world data heterogeneity across clients, thereby influencing the robustness and generalization performance of the global model.

Our experimentation delves into the symbiotic relationship between the Dirichlet distribution parameter β and FL attack dynamics. This interaction is pivotal for our study, as non-IID client datasets can significantly impact the global model's accuracy, even before the introduction of an attack. This pre-existing effect arises from biased, overfitted client models that emerge from non-IID local datasets. This phenomenon amplifies the overall attack impact, as demonstrated in Table 5.

However, it's important to recognize that the impact of non-IIDness is not solely governed by β. A confluence of factors, such as the total number of clients, the clients selected per round, and local and global training epochs, collectively influence the magnitude of the GTA under no attack scenarios. This amalgamation of parameters explains the substantial disparities in the low GTA values observed across datasets like CIFAR10, EMNIST, KBTS, and GTSRB. The intricate relationship between the number of clients, the selection per
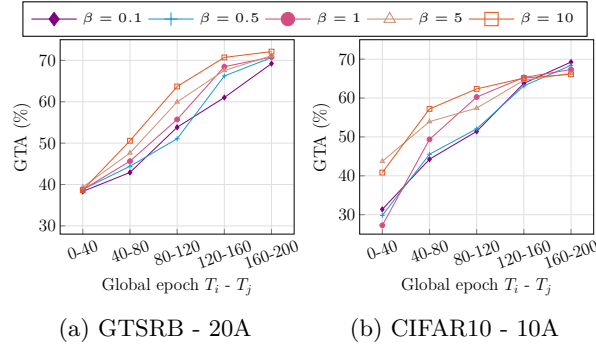
(a) GTSRB - 20A   (b) CIFAR10 - 10A

Figure 5: Analysis of varied degrees of Dirichlet distribution (β) non-IID on our F-CimBA attack impact for multi-client 20A and 10A across GTSRB , and CIFAR10 datasets, respectively.

round, and the resultant GTA values become especially evident when analyzing scenarios with high non-IID data distribution, such as in the case of the $\beta = 0.1$ setting for the EMNIST dataset as shown in Table 5. It is worth noting that the EMNIST dataset exhibits a lower base performance than the other datasets, likely due to the distribution of data among the 10000 clients and the use of 100 randomly chosen clients for aggregation.

Furthermore, to evaluate the effectiveness of our proposed F-CimBA attack in highly non-IID settings, we conducted experiments on the GTSRB and CIFAR10 datasets using different β values (0.1, 0.5, 1, 5, and 10) for multi-client attack 20A, 10A, respectively, as shown in Figure 5 at constant regular intervals of global epochs (for better visualization). As discussed above, the chosen β values represented varying degrees of data distribution among clients, ranging from sparse and unbalanced to dense and balanced. Our findings demonstrate that the F-CimBA attack exhibits a high level of efficacy under non-IID data distributions. Specifically, for the GTSRB dataset with 40 clients, a β value of 0.1 resulted in a lower GTA, while for the CIFAR10 dataset with 100 clients, a β value of 0.1 also yielded a lower GTA. Conversely, a β value of 5 exhibited higher GTA for GTSRB and CIFAR10. This observation can be attributed to the interplay between the selection of malicious updates and the level of data distribution. Specifically, for GTSRB, where the malicious update is always chosen, the combination of $\beta = 1$ and 20% attack volume maximizes the F-CimBA impact. In contrast, for CIFAR10, where malicious updates are not always selected and with an increase in data due to $\beta = 5$, the F-CimBA poisoned data volume increases with an increase in attack volume. However, we note that further increasing β leads to a drop in impact due to the influence of an increasing amount of benign data. Furthermore, the variance in the attack impact within a range of approximately ± 0.2 can be attributed to the data poisoning nature of F-CimBA, where lower β values result in fewer distributed and sampled data among clients. Our future investigation will focus on dynamically adjusting the attack budget in terms of attack volume for every β distribution to enhance the attack impact.

In summary, our in-depth analysis of the non-IID data distribution's impact on FL attacks provides vital insights into the complex dynamics governing FL system performance. The careful calibration of β and its repercussions on data distribution elucidate the underlying factors that can lead to substantial variations in model accuracy and attack effectiveness. This exploration enriches our understanding of FL's behavior under varying conditions and underscores the importance of accounting for non-IIDness in practical scenarios.

7.4 F-CimBA's Resilience against Defenses

We have carefully chosen the following state-of-the-art defense methods due to their relevance and recent advancements in the field. Subsequently, we have organized them into two distinct categories for our analysis. The first category encompasses Byzantine robust aggregation techniques, including Krum (Blanchard et al., 2017), Trimmed mean (Yin et al., 2018), and Median (Yin et al., 2018). The second category comprises very recent defense methods, such as FLTrust (Cao et al., 2021), and even those introduced in 2023, like LoMar (Li et al., 2023), FL-Defender (Jebreel & Domingo-Ferrer, 2023), FABA (Xia et al., 2019), and Centered clipping (Karimireddy et al., 2021). We have organized these techniques to enable a thorough assessment of

the robustness of our F-CimBA against the latest state-of-the-art defense methods in this domain. Below, we provide further details about these defenses.

1. *Krum (Blanchard et al., 2017):* Selects a representative local model update by evaluating the majority of client models. In our experiments, we set $x = 10$.

2. *Trimmed mean (Yin et al., 2018):* Performs aggregation of input updates by discarding the x largest and smallest values in each dimension. We use $x = 2$ for our setup.

3. *Median (Yin et al., 2018):* Aggregates the updates by computing the median of each dimension.

4. *FLTrust (Cao et al., 2021):* Utilizes an auxiliary model trained on a root dataset to compute client trust scores, which are based on the similarity of each client's weight updates to the global model. The server updates are then weighted according to these trust scores.

5. *LoMar (Li et al., 2023):* Employs kernel density estimation to evaluate model updates and identifies an optimal threshold for distinguishing between malicious and legitimate updates.

6. *FL-Defender (Jebreel & Domingo-Ferrer, 2023):* Analyzes the behavior of neurons, proposes robust discriminative features, compresses similarity vectors and adjusts worker updates' weights before aggregation.

7. *FABA (Xia et al., 2019):* Detects and removes outlier gradients from participating workers in distributed training. By filtering poisoned updates, it achieves convergence close to the non-Byzantine case while offering higher efficiency than prior defense methods.

8. *Centered clipping (Karimireddy et al., 2021):* Identifies critical weaknesses in existing Byzantine-robust aggregation methods, showing they can fail even without attackers and remain vulnerable to time-coupled attacks. To address this, the authors propose a robust iterative clipping procedure and incorporate worker momentum, yielding the first provably robust method for standard stochastic optimization.

For a comprehensive evaluation, we conducted comparative analyses of our proposed F-CimBA attack method against the best-performing baseline attacks from various categories, including DPA-DLF, SimBA, and LIE. Our assessment primarily focused on evaluating the resilience of the F-CimBA attack when exposed to defense mechanisms. Specifically, we considered scenarios involving single-client attacks (1A) on the GTSRB and KBTS datasets and multi-client attacks (10A) on the CIFAR10 dataset. In the single-client attack settings, the server selected all 40 clients for aggregation in the case of the GTSRB and KBTS datasets. In the multi-client attack scenario, the server randomly selected 40 out of 100 clients for aggregation, with ten of them being malicious clients. These configurations provided an optimal context to gauge the robustness of our F-CimBA attacks against contemporary defense mechanisms. Our experiments maintained default settings for other parameters, including the use of a $\beta = 1$ non-IID parameter and other parameters. The comparative results showcasing F-CimBA's performance against existing defenses are detailed in Table 9, Table 10, and Table 11.

We observed that our F-CimBA attack exhibited notable performance in various settings. Under the single-client attack (1A) scenario without any defense, we achieved an accuracy of 70.13% for the GTSRB dataset. Even under strong defenses like FABA and Centered Clipping, which show higher robustness across DPA, SimBA, and LIE, F-CimBA is still able to reduce the test accuracy to 80.08% and 81.29%, respectively, showing that our attack can partially bypass these advanced defenses. Under Krum, LoMar, and FLDefender, F-CimBA achieves the lowest accuracy among all attacks, with a drop of up to 4.5–6% more than the next best attack, further demonstrating its stealth and effectiveness. Notably, F-CimBA degrades performance more than SimBA and LIE under every defense method tested, showing its superiority as a black-box, low-visibility attack strategy (Table 9). These results confirm that F-CimBA remains effective even against various defense methods, as we see similar results for KBTS datasets (worked on fewer defenses to avoid redundancy and enable brevity in the results, as shown in Table 10).

Table 9: Comparison of GTA (%) of F-CimBA with other attack methods under state-of-the-art defense techniques for the GTSRB dataset. We use the single-client (1A) attack setting for this experiment. **Bold** result indicates the best results.

Defense	DPA-DLF	SimBA	LIE	F-CimBA (ours)
No Defense	73.19±1.36	72.36±1.45	71.14±1.27	**70.13±1.12**
Krum	79.45±1.27	79.91±0.98	80.24±0.12	**73.95±1.41**
TM	76.42±1.38	75.87±1.15	78.36±0.14	**69.98±1.49**
Median	78.92±0.68	76.85±1.91	79.42±0.93	**72.71±0.58**
FLTrust	8.26±1.75	8.26±1.75	8.26±1.75	**8.19±0.84**
LoMar	80.45±2.30	81.16±1.87	81.13±1.59	**75.12±0.39**
FLDefender	82.34±0.57	80.23±1.29	82.16±0.81	**74.36±1.07**
FABA	83.73±0.27	82.53±1.24	83.36±1.68	**80.08±0.87**
Centered Clipping	86.27±1.03	85.97±1.35	85.24±1.38	**81.29±1.93**

Table 10: Comparison of GTA (%) of F-CimBA with other attack methods under state-of-the-art defense techniques for the KBTS dataset. We use the the single-client (1A) attack setting for this experiment. **Bold** result indicates the best results.

Defense	DPA-DLF	SimBA	LIE	F-CimBA (ours)
Krum	83.32±1.45	85.19±0.83	83.49±1.76	**82.98±0.81**
TM	80.19±1.05	83.98±0.69	82.26±1.46	**73.46±1.19**
Median	79.94±0.18	85.19±1.20	82.45±0.33	**74.98±1.77**
FLTrust	10.24±2.12	10.24±2.12	10.24±2.12	**9.65±1.95**
LoMar	84.26±1.24	84.98±0.15	84.48±0.47	**82.16±1.72**
FLDefender	84.49±0.57	85.46±0.72	85.37±1.46	**83.34±0.41**

In the multi-client attack (10A) setting on the CIFAR10 dataset, we obtained an accuracy of 67.34% in the absence of any defense mechanism. When defense mechanisms were introduced, the accuracy remained competitive, reaching 75.12% for the GTSRB dataset under the LoMar defense and improving to 83.34% for the KBTS dataset under the FLDefender defense. Notably, compared to alternative methods, F-CimBA consistently performed better in terms of maintaining a lower GTA under attack conditions. For instance, the SimBA method achieved an accuracy of 81.16% (second-best) under attack. These findings align with our visibility analysis, discussed in subsequent sections, reinforcing the assertion that F-CimBA offers lower attack visibility compared to its counterparts. These trends persisted in the case of multi-client F-CimBA attacks (10A) even when defenses were in place. Under the LoMar defense, GTA improved slightly to 75.91%, whereas the second-best attack method achieved a GTA of 77.42%. This consistent observation underscores the robustness of the F-CimBA attack, particularly as a feedback-guided approach capable of evading server-side defense mechanisms. Notably, the FLTrust method, reliant on an auxiliary model trained using a root dataset provided by the server, exhibited poor performance across all attacks, indicating that its limited representation of client data in the root dataset adversely affected its effectiveness, as mentioned in prior works

Overall, the F-CimBA attack demonstrated low visibility and proved a resilient attack method even when defenses were deployed. It exhibited only marginal improvements in GTA, approximately 2% and 8% for single and multi-client F-CimBA attacks, respectively, whereas other methods displayed vulnerability to the defense mechanisms.

7.5 F-CimBA Method Attack Visibility Analysis

To assess the attack visibility of our F-CimBA attack, we conducted an experiment using the CIFAR10 dataset in a single-client attack scenario, employing default parameter settings (as detailed in Table 4). This dataset was selected due to its suitability for a comparative analysis of F-CimBA's visibility in relation to other attack methods, given that the server selects 40 malicious clients from a pool of 100. To ensure a fair and meaningful comparison, we selected the best-performing baseline attack from each category, namely

Table 11: Comparison of GTA (%) of F-CimBA with other attack methods under state-of-the-art defense techniques for the CIFAR10 dataset. We use the multi-client (10A) attack setting for this experiment. **Bold** result indicates the best results.

Defense	DPA-DLF	SimBA	LIE	F-CimBA (ours)
Krum	77.91±2.41	79.62±1.02	80.64±1.15	**75.45±0.21**
TM	77.70±1.32	80.12±0.16	81.18±1.06	**73.97±1.12**
Median	77.51±1.27	81.23±0.23	80.50±2.12	**74.23±1.13**
FLTrust	9.49±1.53	9.49±1.53	9.49±1.53	**8.23±1.21**
LoMar	78.36±1.49	77.42±2.31	80.62±1.05	**75.91±1.12**
FLDefender	78.51±2.30	77.28±1.41	80.34±2.70	**74.42±1.30**

DPA-DLF, SimBA, and LIE. In this visibility analysis, the malicious client calculated the validation accuracy of the updated local model after training and subsequently compared it to the accuracy of the current global model received from the server. The measure of attack visibility was quantified by calculating the absolute difference between these two accuracies, with a lower difference indicating reduced attack visibility. This experimental setup allowed us to comprehensively evaluate how effectively F-CimBA operates in terms of its visibility compared to state-of-the-art baseline attack methods. Figure 6 illustrates a comparative analysis of attack visibility, highlighting the performance of our F-CimBA attack in relation to other baseline methods. Our findings reveal that the F-CimBA attack exhibits significantly lower visibility than alternative methods, aligning with our expectations. This reduced visibility is attributed to F-CimBA's strategy of minimizing the loss associated with the most confused class through feedback-guided perturbations.

Specifically, the DPA-DLF attack, characterized by label flipping, displays relatively second lower visibility since it doesn't directly manipulate the data. Our analysis reveals that model poisoning techniques like the LIE attack demonstrate notably high visibility, evidenced by significant disparities in validation accuracy. While our F-CimBA approach shows marginally elevated visibility during early rounds when local models are learning data representations, this effect diminishes as the local model reaches stability. Specifically, the validation accuracy drops to less than 0.05 once stabilized. This value starkly contrasts with baseline approaches such as DPA-DLF, SimBA, and LIE, which exhibit inconsistent visibility patterns that generally increase over time. These findings demonstrate F-CimBA's superior ability to maintain low attack visibility compared to existing attack methodologies.
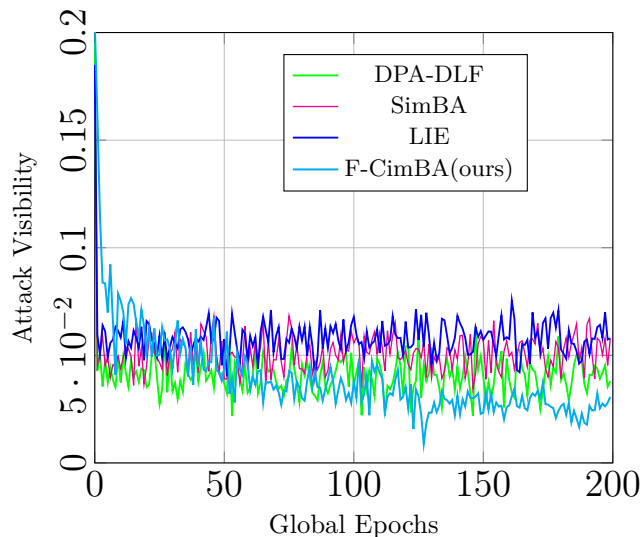

Figure 6: Comparison of the visibility of the F-CimBA attack with various baseline methods across different global epochs for the CIFAR-10 dataset under the single-client (1A) attack settings.

To sum up, our F-CimBA approach has showcased superior performance in terms of lower attack visibility when compared to established baseline methods. This reduced visibility is attributed to F-CimBA's utilization of feedback-driven perturbation techniques, which focus on minimizing the loss associated with the most confused class. These observations are substantiated by the results presented in the preceding section, where F-CimBA consistently outperformed other attacks, even when confronted with server-side defenses. Essentially, this implies that the perturbations introduced by F-CimBA are less conspicuous and pose a more challenging detection problem than those generated by other attack methods.

7.6 F-CimBA vs. Hardware Performance Metrics

This section analyzes the impact of the F-CimBA attack on system resources. Previous research has focused on traditional attack efficacy metrics such as global test accuracy, but this study is the first to examine hardware metrics. We analyze six metrics (execution time, bandwidth, CPU/GPU utilization, memory usage, and GPU temperature) under no attack, single-client attack, and multi-client (50%) attack across global epochs. This section investigates attack progression through the lens of system resource consumption. Our motivating factor is understanding the bottleneck issues due to the F-CimBA attack. Prior works have emphasized the conventional attack efficacy metrics, such as the drop in the global test accuracy. To the best of our knowledge, this additional viewpoint of hardware metrics has never been presented before. We consider seven metrics, namely, *execution time, bandwidth (throughput of the uplink channel), CPU utilization, RAM utilization, GPU memory usage, GPU utilization, and GPU temperature*, in our evaluation. All the metrics are analyzed thoroughly under three settings: without attack, single-client attack, and multi-client (50%) attack throughout global epochs.

Table 12 and Figure 7 present the above-mentioned results. We observe that the attack drastically impacts the low-end resource-constrained client devices that can eventually become stragglers. Although the attack originated from a single/group of malicious clients in the minority, it eventually starts influencing the learning operations of all the benign clients. This further perpetuates the attack as long as the minor perturbations remain undetected. We summarize our observations as (i) execution time increases up to $30\times$ when we compare the multi-client attack w.r.t. no attack in Figure 7a. This effect drastically reduces the communication bandwidth in Figure 7b as it is calculated as the volume of information that can be sent over the communication channel in a measured amount of time. Also, it drastically increases the load on CPU utilization compared to the scenario without attack, as shown in Figure 7c. Further, as expected, we observe the system hardware impact is much worse in multi-client w.r.t. single-client attacks. (ii) The overall GPU memory and utilization by the client and the central server remain consistent (Table 12 columns 2 & 3), irrespective of single or multi-client attacks. This result starkly contrasts variations in the CPU metrics, which show that the GPU is resilient under attack. (iii) The temperature of GPU hardware (Table 12 column 4) drastically rises under attack, leading to overheating and faster disruption. Overall, FL researchers working on adversarial attack detection can use these hardware performance trends to build future resilient FL systems.

Table 12: GPU memory (GM), GPU utilization (GU), GPU temperature (GT) under no attack, single-client, and multi-client attack using the proposed F-CimBA.

Attack setting	GM (MB)	GU (%)	GT ($^\circ C$)
No attack	5261	97	40
Single-client attack	5827	32	44
Multi-client (50%) attack	5827	32	46

7.7 Discussion1: Why Malicious Gradient Updates in FL Tend to Poison the Global Model

It is crucial to understand the reasoning behind why, in the FL setting, malicious gradient updates will always indirectly poison the global model, thus degrading its performance. The fundamental reason lies in how standard FL systems aggregate client updates. Under standard aggregation schemes like FedAvg, well-designed malicious updates, especially those that remain undetected, **consistently bias the global model over time, leading to degradation in performance**. This effect has been **both theoretically analyzed**
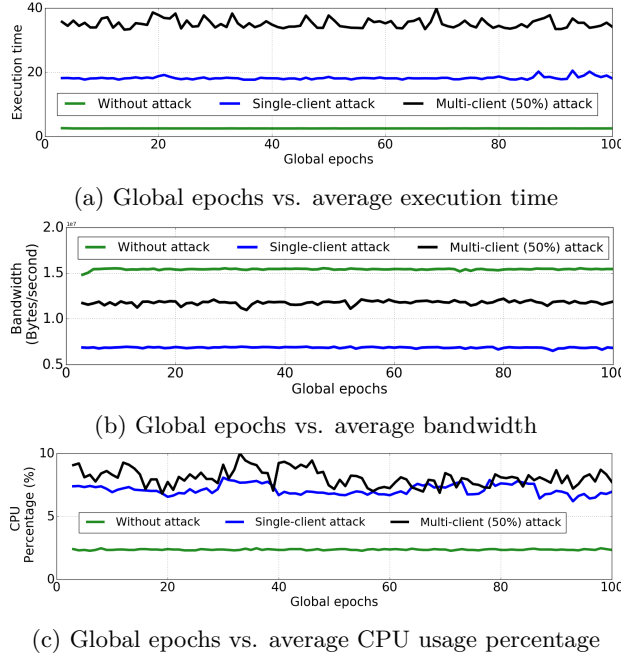
(a) Global epochs vs. average execution time

(b) Global epochs vs. average bandwidth

(c) Global epochs vs. average CPU usage percentage

Figure 7: Hardware performance graphs under (a) no attack, (b) single-client, and (c) multi-client attack using the proposed F-CimBA

and empirically validated in multiple prior works (Shejwalkar et al., 2022; Usynin et al., 2021; Kumar et al., 2024b). When a malicious client submits an adversarial update, crafted through either data poisoning or model poisoning, it contributes directly to the global model update, suboptimal or adversarially biased decision boundaries that reduce test accuracy, induce targeted misbehaviour or slow convergence. The server aggregates local model updates in FL, assuming that all clients act honestly. This lack of verification creates an inherent vulnerability, as shown in prior works below.

- Shejwalkar *et al.* (Shejwalkar et al., 2022) comprehensively evaluate untargeted poisoning attacks and confirm that malicious updates can successfully bias the global model when not effectively filtered by the defense, even under constrained adversarial capabilities.

- Usynin *et al.* (Usynin et al., 2021) highlight that adversarial interference in collaborative learning leads to meaningful reductions in utility, even when perturbations are subtle. They show that even privacy-preserving FL systems are vulnerable to low-visibility attacks that do not require gradient access yet can significantly undermine the utility of the global model over multiple communication rounds.

- Kumar *et al.* (Kumar et al., 2024b) explain that attacks with low attack visibility and small budgets can still have a high impact, particularly when the global model incorporates updates without sufficient scrutiny. These insights reinforce that well-designed poisoning attacks do not need to be large in magnitude to be impactful.

7.8 Discussion2: Why Small μ is Still Effective in F-CimBA

In general, a smaller value of μ bounds the perturbation magnitude and, in theory, limits the per-sample increase in loss. However, several key factors explain why even small μ values can still lead to effective and sustained model degradation in our F-CimBA attack.

- *FL accumulates small deviations over rounds.* FL is an iterative process where client updates are aggregated repeatedly across communication rounds. Even if the per-update impact of a low μ

perturbation is small, its cumulative effect over T rounds becomes significant, particularly when the adversarial client participates repeatedly (online attack). This compounding nature of poisoning, also highlighted in prior work such as Usynin *et al.* (Usynin et al., 2021), allows small, undetectable updates to gradually shift the global model toward an adversarial objective.

- *Our attack focuses on model confusion, not large-scale loss spikes.* F-CimBA does not aim to maximise loss arbitrarily but rather to increase the confidence of incorrect class predictions (i.e., the most confused class). This strategy does not require large perturbations. Instead, it takes advantage of **local model weaknesses and decision boundaries**, which can be nudged effectively with small perturbations. Using a small μ ensures the perturbed samples stay close to the original data distribution, making them less likely to be detected or filtered by defenses. The result is a more stealthy degradation of performance, which is visible in our empirical results.

- *Stealth and detectability.* Empirical evidence shows a clear impact even at low μ. With $\mu = 0.5$ (used in our experiments), we achieve a test accuracy degradation of up to 19% without triggering defenses. This validates that small μ can still achieve a strong practical impact due to the attack's cumulative and focused nature.

In summary, in practical FL settings, even low-visibility malicious updates, when aggregated, can consistently poison the global model and degrade its performance. At the same time, smaller μ limits the immediate effect, F-CimBA leverages iteration, subtle misdirection, and low detectability to reliably cause degradation over time.

8 Ablation Study

In this section, we evaluate the effectiveness of the F-CimBA attack against only SimBA (second best-performing attack) for brevity under model heterogeneity, client scaling, homogeneous & heterogeneous FL, attack progression w.r.t. global epochs, and convergence rate.

8.1 Model Architecture Diversity and Effectiveness of F-CimBA

To further validate the effectiveness and generalizability of F-CimBA, we conducted additional experiments on the CIFAR-10 dataset using three diverse and widely adopted model architectures: ResNet-50 (He et al., 2016), DenseNet-121 (Huang et al., 2017), and Vision Transformer (ViT) (Dosovitskiy et al., 2020). The FL setup was configured with $n = 100$ total clients, $k = 40$ randomly selected clients per round, 10 malicious clients, and a non-IID Dirichlet partitioning with $\beta = 1$ to simulate realistic data heterogeneity. We evaluate these models against multiple poisoning and black-box baselines, including DPA-DLF, DPA-SLF, LIE, Fang *et al.*, and SimBA, as well as a no-attack reference. The results are presented in Table 13, and the key observations are as follows:

Table 13: Global test accuracy (%) of different FL model architectures under various poisoning attacks on the CIFAR-10 dataset ($n = 100$, $k = 40$, 10 malicious clients, non-IID $\beta = 1$). F-CimBA consistently achieves the largest accuracy degradation across CNN and transformer-based architectures, highlighting its architecture-agnostic effectiveness.

Architecture	No Attack	DPA-DLF	DPA-SLF	LIE	Fang *et al.*	SimBA	F-CimBA (Ours)
ResNet-50	94.12±1.26	93.57±0.64	93.02±1.41	91.37±0.92	87.48±0.47	88.36±1.84	**85.67±1.03**
DenseNet-121	92.64±0.83	91.95±1.12	91.57±0.76	90.83±1.67	90.36±1.49	90.74±0.94	**88.31±0.65**
Vision Transformer	95.85±1.38	94.45±0.79	94.56±1.42	93.27±0.93	92.34±1.51	92.01±0.68	**91.44±1.23**

- Across all three architectures, F-CimBA consistently induces the largest drop in test accuracy, highlighting its strong poisoning capability regardless of architectural bias or learning dynamics.

- On ResNet-50, F-CimBA lowers accuracy to 85.67%, representing an 8.45% drop from the clean model and a 1.8–5.7% drop compared to competing attacks.

- On DenseNet-121, accuracy drops to 88.31%, maintaining at least a 2.0–3.6% margin over the next strongest attack.

- On ViT, a transformer-based model with fundamentally different internal mechanisms, F-CimBA still yields the highest performance degradation (91.44%), surpassing all other attacks, including those tailored for high-impact poisoning.

These findings reinforce the architecture-agnostic nature of F-CimBA, demonstrating its ability to degrade performance across both convolutional and transformer-based models.

8.2 Client Scaling and Homogeneous & Heterogeneous FL Settings

To evaluate the impact of client scaling on the F-CimBA attack, we progressively increase the number of clients $n = 3, 5, 10, 15, 25, 40$ to present a controlled view of the experiment space for GTSRB and KBTS datasets (for brevity). Next, we consider another FL data shard setting called *homogeneous*, where $|\mathbb{D}_k| = l_k$ (dataset equally divided among clients) and show a comparative evaluation with heterogeneous FL settings, where $|\mathbb{D}_1| = l_1$, $|\mathbb{D}_2| = l_2$, ..., $|\mathbb{D}_n| = l_n$, where we assign data randomly such that $1000 \leq l_k \leq 1500$. Finally, we perform single-client and multi-client attacks with attack percentages as $\{30, 50\}\%$ for brevity under these different clients and FL settings. Along with the GTA (%), we also consider the number of global epochs for convergence and average attack success rate as metrics to compare the overall effectiveness.

No attack and no defense case: It is important to evaluate the "no attack and no defense" case in an experimental setting because it provides a baseline against which the effectiveness of different attacks and defenses can be measured. For example, if a defense is shown to improve the performance of the FL system under attack, it is important to know how much the attack degraded the FL system's performance in the first place. Without a baseline, it would be difficult to accurately assess the effectiveness of the defense. Evaluating the "no attack and no defense" case also helps to ensure that any observed changes in performance are due to the attacks or defenses being tested and not other factors. This evaluation can increase the validity and reliability of the experimental results. Table 14 presents the minimum (40 clients) and maximum (3 clients) accuracies achieved under no attack and no defense (BA) results for both datasets. As we increase the number of clients, the data size of each client reduces, resulting in an accuracy drop. Further, homogeneous FL with the same amount of data has shown slightly better performance than heterogeneous FL settings for both datasets.

Table 14: Global test accuracy (%) under no attack and no defense techniques for both datasets' homogeneous and heterogeneous FL settings.

	FL-Hom		FL-Het	
	Max ($3C$)	**Min** ($40C$)	**Max** ($3C$)	**Min** ($40C$)
GTSRB	94.8	90.23	95.2	89.8
KBTS	98.3	93.46	97.1	91.12

Attack cases: Figure 8 shows single-client attack accuracy, convergence results under different clients, and homogeneous & heterogeneous FL for both datasets. Interestingly, as we increase the number of clients, the number of benign clients increases, increasing overall accuracy. Further, F-CimBA-Het has shown better accuracy than SimBA attack configurations, as shown in Figure 8a and Figure 8b.

Furthermore, homogeneous FL showed a better convergence rate (fewer global epochs to converge) for both attacks and both datasets, as shown in Figure 8c, Figure 8d, and Figure 9. This is because each client has an equal data size, making it easier for the attack to generate an equal amount of poisoned data as benign data. This effect leads to similar malicious contributions and dominates the benign updates, leading to early convergence. Further, F-CimBA configurations can converge faster than SimBA for single and multi-client attacks. Additionally, Figure 10 presents results for multi-client attacks with attack percentages of 30% and 50% for both datasets. We observe similar trends of single-client attack with reduced accuracies, and F-CimBA configurations outperformed other methods.

(a) GTA - GTSRB   (b) GTA - KBTS   (c) GE - GTSRB   (d) GE -KBTS
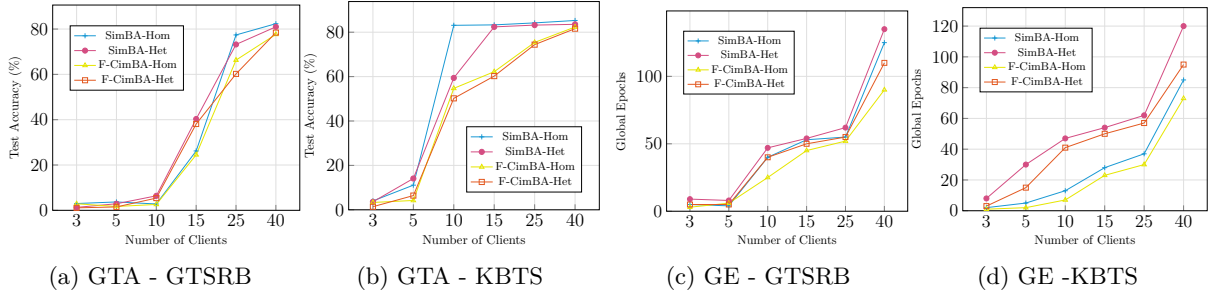
Figure 8: Comparison of global test accuracy (GTA) and global epochs (GE) of F-CimBA against SimBA across two different FL settings and single-client attack.
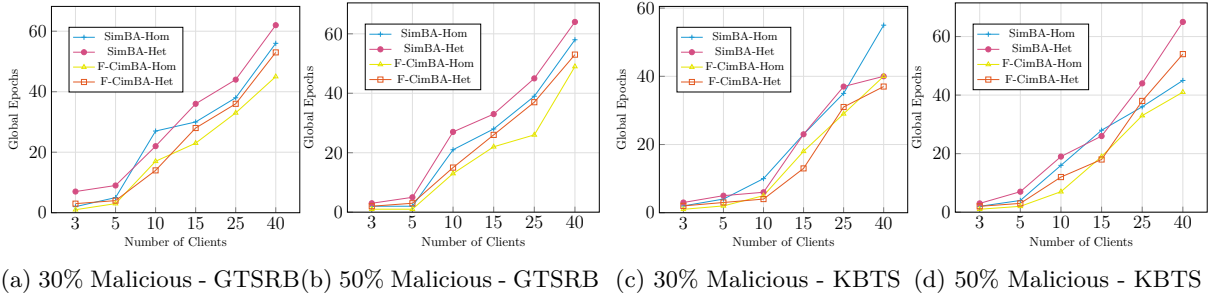

(a) 30% Malicious - GTSRB (b) 50% Malicious - GTSRB   (c) 30% Malicious - KBTS   (d) 50% Malicious - KBTS

Figure 9: Comparison of global convergence test epochs of F-CimBA against SimBA (second best-performing attack) across two different homogeneous & heterogeneous FL settings, and 30% & 50% multi-client attack.

Average attack success rate: Table 15 and Table 16 present results of the average attack success rate given by the ratio of the number of corrupted training samples per total samples for different F-CimBA configurations against SimBA configurations. We observe that for 50% of all cases, our F-CimBA has shown a 100% average attack success rate from low to larger client sizes.

Table 15: Average attack success rate (%) on GTSRB.

Clients	SimBA–Hom	F-CimBA–Hom	SimBA–Het	F-CimBA–Het
3	99.38	100	96.92	98.46
5	100	100	95.23	98.41
10	91.83	95.91	97.29	97.29
15	96.87	100	93.54	96.77
25	89.47	94.73	94.11	100
40	90.23	95.68	92.88	98.97

Table 16: Average attack success rate (%) on KBTS.

Clients	SimBA–Hom	F-CimBA–Hom	SimBA–Het	F-CimBA–Het
3	96.55	100	89.47	94.73
5	88.23	94.11	87.5	93.75
10	87.5	100	88.5	90.23
15	87.63	91.45	90	94.5
25	86.66	97.45	86.6	90.1
40	90.13	94.56	91.33	97.645

In summary, the feedback-guided mechanism in F-CimBA helps in early attack convergence, high success rate, and dropping the accuracy under single and multi-client attack settings for both datasets.

(a) 30% Malicious - GTSRB (b) 50% Malicious - GTSRB (c) 30% Malicious - KBTS (d) 50% Malicious - KBTS
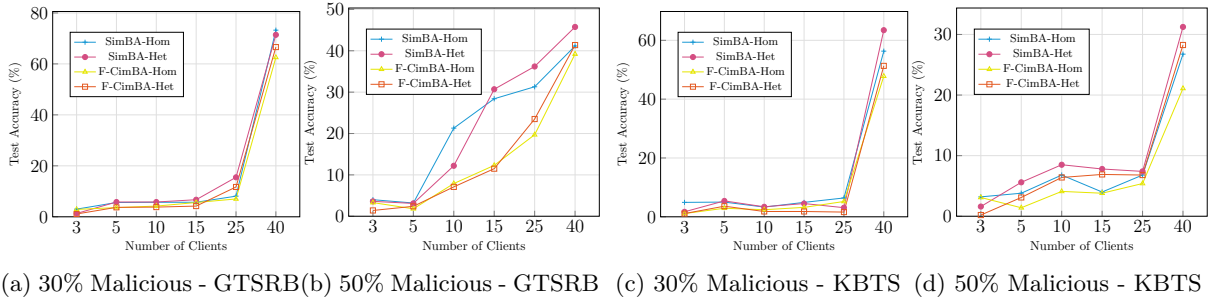
Figure 10: Comparison of global test accuracy of F-CimBA against SimBA (second best-performing attack) across two different homogeneous & heterogeneous FL settings, and 30% & 50% multi-client attack.

8.3 Impact of Client Participation Setting on Attack Effectiveness

To evaluate the robustness and generalizability of F-CimBA under different client participation schemes, we extended our experiments on the GTSRB dataset to include a *partial participation* setup. In this configuration, $n = 40$ clients were simulated, with $k = 20$ clients randomly selected per communication round for aggregation. The attack scenario involved a single adversarial client (1A), and the same set of defense baselines used in our main experiments were applied, namely Krum, LoMar, FLDefender, FABA, and Centered Clipping. The attack baselines considered were DPA-DLF, SimBA, LIE, and our proposed F-CimBA.

The results are presented in Table 17. We observe that F-CimBA consistently achieves the largest drop in test accuracy across all defenses, even in the partial participation setting. Under Krum, F-CimBA reduces the accuracy to 72.12%, which is 6.35–9.24% lower than competing attacks in the same setting. Against strong defenses such as FABA and Centered Clipping, F-CimBA attains 81.21% and 80.89%, respectively, still outperforming all other attacks in terms of degradation. These results are consistent with those obtained under full participation for GTSRB, reinforcing that F-CimBA remains effective regardless of the client participation scheme.

This additional evaluation confirms that our earlier choice of employing both full and partial participation settings across different datasets was a reasonable trade-off to avoid redundancy, while maintaining a comprehensive and fair comparison of attack performance.

Table 17: Global test accuracy (%) of different attacks against various defenses on the GTSRB dataset under partial participation (40 clients total, 20 selected per round, 1 malicious client). F-CimBA consistently causes the largest degradation across all defenses, demonstrating its robustness to changes in participation setting.

Defense Method	DPA-DLF	SimBA	LIE	F-CimBA (Ours)
No Defense	82.46±1.31	81.86±0.87	78.90±1.68	**77.54±1.42**
Krum	78.47±1.05	80.38±0.73	81.36±1.24	**72.12±1.17**
LoMar	81.78±0.94	82.34±0.86	80.15±0.98	**74.68±1.63**
FLDefender	81.38±0.91	81.68±1.09	83.11±0.72	**76.27±1.11**
FABA	84.30±0.84	83.76±1.14	84.90±0.99	**81.21±1.22**
Centered Clipping	85.81±1.09	84.01±1.37	83.71±1.48	**80.89±0.96**

8.4 Byzantine Robust Analysis of F-CimBA

FL's Byzantine aggregation rules protect against malicious clients in Byzantine failures. These rules aggregate model updates in a way that resists the influence of malicious updates, improving the reliability and robustness of the FL system and protecting against attacks or other failures that could compromise the model's accuracy. We present more results of F-CimBA against Byzantine robust aggregation rules as an extension to Section 7.4.

No attack with defense case: We perform experiments on our FL setup with no attack and different Byzantine-robust aggregation rules at the servers as a base case to compare the F-CimBA attack accuracy

drop and demonstrate the effectiveness. Table 18 presents the no attack with defense results for both datasets. We observe that the Krum method results are better than other methods' results under no attack.

Table 18: Global test accuracy (%) of our FL setting under no attack and with Byzantine robust defense techniques.

	GTSRB	KBTS
Krum	87.35	89.72
TM	85.43	87.34
Median	86.22	88.23

Attack cases: Figure 11 and Table 19 present the results (*AD*) of three different aggregation rules under single and multi-client attack settings for both datasets. *Single-client attack:* We observe that our proposed F-CimBA constantly showed robustness to byzantine aggregation rules compared to the SimBA attack with an accuracy drop of around 13.4%, 17.37%, and 14.64% corresponding to Krum, TM, and median aggregation rules, respectively, for the GTSRB dataset. For the KBTS dataset, the reported drop in accuracies is 6.74%, 16.26%, and 14.74% for Krum, TM, and Median aggregation rules, respectively, as shown in Figure 11a, Figure 11b, and Table 19. It is interesting to note that among the different Byzantine aggregation rules, Krum performed better than TM and median. This is because Krum selects one local model update representative of most client models by computing the pairwise distances between individual models (Blanchard et al., 2017). However, when the data across the workers is highly non-i.i.d. and poisonous, there is no 'representative' client model. Hence, it results in a performance drop under attack. On the other hand, trimming model updates after sorting client updates, including median, results in considering malicious updates for aggregation and hence the poor results.

Multi-client attack: Figure 11c, Figure 11d, and Table 19 present results on multi-client attack results with Byzantine robust aggregation rules. We observe a similar trend as a single-client attack with a significant drop in accuracy (rating from 20.38% - 84.63% for GTSRB and 30.3% - 82.87% for KBTS dataset) with the increase in attack percentage (from 20% - 80%). Further, TM has consistently shown less Byzantine robustness to our F-CimBA attack than other aggregation rules for GTSRB. In addition, in the KBTS data, both TM and median have shown less performance than the Krum rule.

(a) Single-client - GTSRB (b) Single-client - KBTS (c) Multi-client - GTSRB (d) Multi-client - KBTS
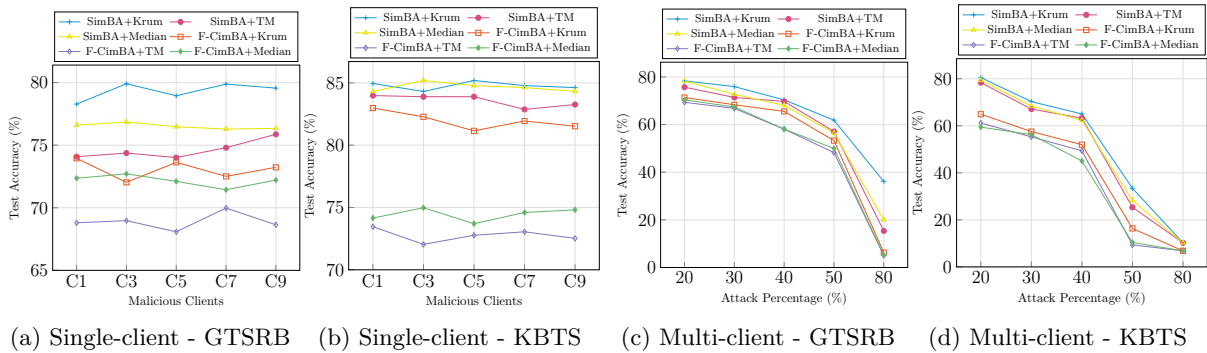
Figure 11: Effectiveness on F-CimBA against existing Byzantine robust aggregation techniques: Comparison of global test accuracy F-CimBA against SimBA (second best performing attack) across two different datasets and attack settings.

In summary, F-CimBA is robust against Byzantine aggregation rules at the server for both attack settings and datasets. It has shown better robustness compared to the SimBA attack. With an ability to select a representative of most client models by computing pairwise distances, the Krum aggregation rule effectively shows better robustness than TM and median.

Table 19: Comparison of F-CimBA accuracy (%) with SimBA (second best performing attack) under Byzantine robust aggregation techniques for two datasets and attack settings. **Bold** result indicates the best results.

		GTSRB						KBTS					
	Client Ids	SimBA+ Krum	SimBA+ TM	SimBA+ Median	F-CimBA+ Krum	F-CimBA+ TM	F-CimBA+ Median	SimBA+ Krum	SimBA+ TM	SimBA+ Median	F-CimBA+ Krum	F-CimBA+ TM	F-CimBA+ Median
Single-client Attack	C1	78.28	74.08	76.6	73.95	**68.8**	72.36	84.95	83.98	84.31	82.98	**73.46**	74.15
	C3	79.91	74.37	76.85	72.03	**68.97**	72.71	84.31	83.89	85.19	82.28	**72.04**	74.98
	C5	78.95	74.01	76.47	73.63	**68.08**	72.11	85.19	83.89	84.78	81.15	**72.77**	73.7
	C7	79.88	74.8	76.29	72.51	**69.98**	71.44	84.78	82.87	84.62	81.94	**73.05**	74.6
	C9	79.56	75.87	76.34	73.23	**68.64**	72.21	84.62	83.26	84.32	81.52	**72.52**	74.81
	Attak Percentage (%)	SimBA+ Krum	SimBA+ TM	SimBA+ Median	F-CimBA+ Krum	F-CimBA+ TM	F-CimBA+ Median	SimBA+ Krum	SimBA+ TM	SimBA+ Median	F-CimBA+ Krum	F-CimBA+ TM	F-CimBA+ Median
Multi-client Attack	20	78.35	75.71	78.28	71.36	**69.34**	70.38	80.5	78.36	79.36	64.92	61.18	**59.42**
	30	75.94	71.38	72.71	68.32	**66.8**	67.38	70.29	67.12	68.42	57.54	**55.29**	56.24
	40	70.36	69.71	68.07	65.54	**58.05**	58.11	65.04	63.25	62.31	51.97	49.38	**45.04**
	50	61.83	57.09	56.47	53.28	**48.25**	49.97	33.42	25.33	28.26	16.35	**9.35**	10.38
	80	36.11	15.36	20.04	6.23	**5.09**	5.09	10.23	10.23	10.23	6.85	**6.85**	**6.85**

8.5 Time-series Analysis of F-CimBA Attack

Figure 12 displays results for test accuracy in regular and attack cycles for different clients for *heterogeneous* FL settings for the GTSRB dataset. We demonstrate only single-client attack results of our F-CimBA (Figure 12b) against SimBA (Figure 12a) for brevity. An expected, rising trend of test accuracy is observed for FL when trained on normal data. The model trained on normal data with three clients resulted in an accuracy of 95.2% for the GTSRB dataset under normal heterogeneous FL settings. The *attack phase* begins after completing the *normal phase* at 30 epochs. The normal and attack phases are juxtaposed to represent their contrasting behavior. We can observe a *significant distortion in the model test accuracy throughout the attack phase for the entire experiment design space* in all our result figures. Our proposed F-CimBA attack approach demonstrates consistent results in better convergence for heterogeneous clients (3, 5, 10) and overall average accuracy drop, causing maximum damage with a better attack local minimum.
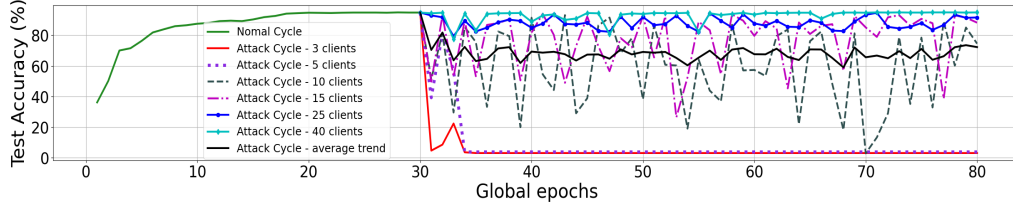
(a) Single-client attack on GTSRB under SimBA-Het attack.
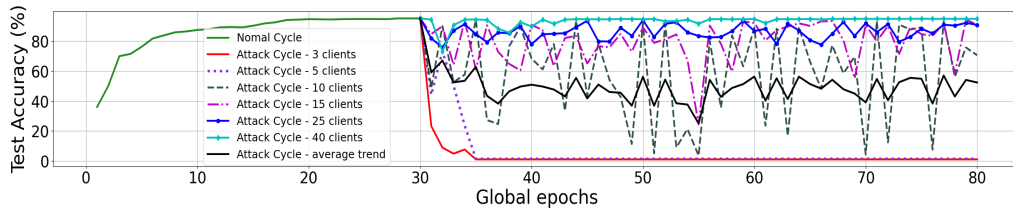
(b) Single-client attack on GTSRB under F-CimBA-Het attack.

Figure 12: Test Accuracy of the normal cycle and single-client attack cycle of proposed F-CimBA framework against SimBA attack along with average attack trend under heterogeneous FL settings.

8.6 Broader Impact Statement

This work investigates a novel black-box, low-visibility poisoning attack (F-CimBA) in the context of FL. While the methodology involves adversarial attacks, the primary objective is to advance the understanding of vulnerabilities in FL systems in order to improve their robustness, reliability, and trustworthiness. Our research is aligned with prior work in FL security that emphasizes exposing and characterizing weaknesses as

a prerequisite for building effective defenses. Shejwalkar *et al.* (Shejwalkar et al., 2022) show that malicious updates can bias the global model even under constrained attacker capabilities. Usynin *et al.* (Usynin et al., 2021) demonstrate that subtle, low-visibility interference can still significantly degrade utility in collaborative learning. Kumar *et al.* (Kumar et al., 2024b) further highlight that attacks with small budgets can have a disproportionately large impact when model updates are not sufficiently scrutinized.

In this context, F-CimBA is designed as a realistic, black-box threat model to fill the gap between overly simplistic adversarial settings and production-grade FL deployments. By systematically evaluating how such an attack performs under diverse defenses (e.g., Krum, LoMar, FLDefender, FABA, and Centered Clipping), our work seeks to proactively inform the development of stronger, more adaptive countermeasures. We emphasize that our methodology is presented strictly from a scientific and diagnostic standpoint. No instructions are provided for malicious deployment, and our experiments are conducted in controlled research settings. We explicitly discourage any misuse of these findings in real-world systems. Instead, we intend this work to serve as a resource for the research community to design and evaluate robust defense mechanisms against practical and stealthy FL attacks.

9   Limitations and Future Work

Despite the effectiveness of our F-CimBA, there are certain limitations, which we plan to explore in our future work.

- **Adaptive parameter engineering:** The effectiveness of F-CimBA relies on a key parameter called the noise coefficient (μ), which is initially set to 0.5 in our experiments. In future research, we will explore dynamic parameter learning to potentially enhance F-CimBA's impact by adapting this parameter during training. Recognizing the imperative need for robust attack visibility analysis in the black-box threat model, we plan to investigate alternative metrics and scenarios. This plan includes examining different threat models, encompassing both full and partial attacker knowledge settings, for a comprehensive understanding of F-CimBA's behaviour.

- **Applicability to other FL tasks:** In future research, we will extend our investigation to object detection and segmentation within the federated learning framework. Benign clients will contribute regular updates for training these models, while malicious clients will employ F-CimBA to poison the training data. The aggregated global model is anticipated to perform poorly in detecting or segmenting test data. This strategic expansion of our research aims to provide valuable insights into task-specific challenges.

10   Conclusion

In this paper, we propose F-CimBA, a feedback-guided causative image black-box attack in federated learning, considering a cautious attacker's perspective. Current attack methods often exhibit substantial attack visibility while retaining impact, attributed to their inherent nature, the scale of perturbations, and a lack of strategies to evade detection mechanisms. Therefore, low attack visibility is essential for maintaining a covert operation, enabling attackers to accomplish their goals without activating the server's defense mechanisms. Consequently, the core of F-CimBA lies in reducing the loss linked to the most confused class, leveraging its probability as feedback to guide the search and refinement of adversarial perturbation noise. This approach helps to exploit local model vulnerabilities, ensure early attack convergence, and achieve higher misclassification rates. Our empirical results demonstrate F-CimBA's effectiveness in poisoning client training data, causing substantial mispredictions of the global model. In addition, F-CimBA exhibits robustness against Byzantine-robust aggregation techniques at the server. Also, our investigation into system hardware metrics has revealed the adverse impact of our attack on their performance. In our future works, we aim to explore adaptive parameter engineering techniques, focusing on the dynamic learning of the μ parameter. Additionally, we intend to understand the impact of utilizing both full and partial attacker knowledge settings to enhance the analysis of attack visibility. Furthermore, our research plan includes evaluating the performance of F-CimBA across diverse tasks, such as object detection and object segmentation within the federated learning framework.

References

Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 2938–2948. PMLR, 2020.

Gilad Baruch, Moran Baruch, and Yoav Goldberg. A little is enough: Circumventing defenses for distributed learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. Model poisoning attacks in federated learning. In *Proc. Workshop Secur. Mach. Learn.(SecML) 32nd Conf. Neural Inf. Process. Syst.(NeurIPS)*, pp. 1–23, 2018.

Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. Analyzing federated learning through an adversarial lens. In *International Conference on Machine Learning*, pp. 634–643. PMLR, 2019.

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.

Xiaoyu Cao, Minghong Fang, Jia Liu, and Neil Zhenqiang Gong. Fltrust: Byzantine-robust federated learning via trust bootstrapping. In *28th Annual Network and Distributed System Security Symposium, NDSS 2021, virtually, February 21-25, 2021*. The Internet Society, 2021.

Chen Chen, Lingjuan Lyu, Han Yu, and Gang Chen. Practical attribute reconstruction attack against federated learning. *IEEE Transactions on Big Data*, pp. 1–1, 2022. doi: 10.1109/TBDATA.2022.3159236.

Yu Chen, Fang Luo, Tong Li, Tao Xiang, Zheli Liu, and Jin Li. A training-integrity privacy-preserving federated learning scheme with trusted execution environment. *Information Sciences*, 522:69–79, 2020.

Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. Emnist: Extending mnist to handwritten letters. In *2017 international joint conference on neural networks (IJCNN)*, pp. 2921–2926. IEEE, 2017.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. Local model poisoning attacks to byzantine-robust federated learning. In *Proceedings of the 29th USENIX Conference on Security Symposium*, pp. 1623–1640, 2020.

Gilad Gressel, Niranjan Hegde, Archana Sreekumar, and Michael Darling. Feature importance guided attack: A model agnostic adversarial attack. *arXiv preprint arXiv:2106.14815*, 2021.

Chuan Guo, Jacob Gardner, Yurong You, Andrew Gordon Wilson, and Kilian Weinberger. Simple black-box adversarial attacks. In *International Conference on Machine Learning*, pp. 2484–2493. PMLR, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL http://arxiv.org/abs/1512.03385.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

Andrew Ilyas, Logan Engstrom, and Aleksander Madry. Prior convictions: Black-box adversarial attacks with bandits and priors. *ICLR 2019*, 2018. URL https://arxiv.org/abs/1807.07978.

Najeeb Moharram Jebreel and Josep Domingo-Ferrer. Fl-defender: Combating targeted attacks in federated learning. *Knowledge-Based Systems*, 260:110178, 2023.

Sai Praneeth Karimireddy, Lie He, and Martin Jaggi. Learning from history for byzantine robust optimization. In *International conference on machine learning*, pp. 5311–5319. PMLR, 2021.

Taejin Kim, Shubhranshu Singh, Nikhil Madaan, and Carlee Joe-Wong. Characterizing internal evasion attacks in federated learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 907–921. PMLR, 2023.

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

K Naveen Kumar, C Krishna Mohan, Linga Reddy Cenkeramaddi, and Navchetan Awasthi. Minimal data poisoning attack in federated learning for medical image classification: An attacker perspective. *Artificial Intelligence in Medicine*, pp. 103024, 2024a.

Kummari Naveen Kumar, Chalavadi Krishna Mohan, and Linga Reddy Cenkeramaddi. The impact of adversarial attacks on federated learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5):2672–2691, 2024b. doi: 10.1109/TPAMI.2023.3322785.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Cody Lewis, Vijay Varadharajan, and Nasimul Noman. Attacks against federated learning defense systems and their mitigation. *Journal of Machine Learning Research*, 24(30):1–50, 2023.

Wenqian Li, Shuran Fu, Fengrui Zhang, and Yan Pang. Data valuation and detections in federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12027–12036, 2024.

Xingyu Li, Zhe Qu, Shangqing Zhao, Bo Tang, Zhuo Lu, and Yao Liu. Lomar: A local defense against poisoning attack on federated learning. *IEEE Transactions on Dependable and Secure Computing*, 20(1):437–450, 2023. doi: 10.1109/TDSC.2021.3135422.

Markus Mathias, Radu Timofte, Rodrigo Benenson, and Luc Van Gool. Traffic sign recognition — how far are we from the solution? In *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2013. doi: 10.1109/IJCNN.2013.6707049.

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*, pp. 1273–1282. PMLR, 2017.

Thomas Minka. Estimating a dirichlet distribution, 2000.

Arup Mondal, Yash More, Ruthu Hulikal Rooparaghunath, and Debayan Gupta. Poster: Flatee: Federated learning across trusted execution environments. In *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 707–709. IEEE, 2021.

Viraaji Mothukuri, Reza M Parizi, Seyedamin Pouriyeh, Yan Huang, Ali Dehghantanha, and Gautam Srivastava. A survey on security and privacy of federated learning. *Future Generation Computer Systems*, 115:619–640, 2021.

Amin Rakhsha, Goran Radanovic, Rati Devidze, Xiaojin Zhu, and Adish Singla. Policy teaching in reinforcement learning via environment poisoning attacks. *Journal of Machine Learning Research*, 22(210):1–45, 2021.

Virat Shejwalkar and Amir Houmansadr. Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning. In *NDSS*, 2021.

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

Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The German Traffic Sign Recognition Benchmark: A multi-class classification competition. In *IEEE International Joint Conference on Neural Networks*, pp. 1453–1460, 2011.

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

Yuwei Sun, Hideya Ochiai, and Jun Sakuma. Attacking distance-aware attack: Semi-targeted model poisoning on federated learning. *IEEE Transactions on Artificial Intelligence*, pp. 1–15, 2023. doi: 10.1109/TAI.2023.3280155.

Dmitrii Usynin, Alexander Ziller, Marcus Makowski, Rickmer Braren, Daniel Rueckert, Ben Glocker, Georgios Kaissis, and Jonathan Passerat-Palmbach. Adversarial interference and its mitigations in privacy-preserving collaborative machine learning. *Nature Machine Intelligence*, 3(9):749–758, 2021.

Jon Vadillo, Roberto Santana, and Jose A Lozano. Extending adversarial attacks to produce adversarial class probability distributions. *Journal of Machine Learning Research*, 24(15):1–42, 2023.

Bokun Wang, Zhuoning Yuan, Yiming Ying, and Tianbao Yang. Memory-based optimization methods for model-agnostic meta-learning and personalized federated learning. *Journal of Machine Learning Research*, 24(145):1–46, 2023.

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.

Qi Xia, Zeyi Tao, Zijiang Hao, and Qun Li. Faba: an algorithm for fast aggregation against byzantine attacks in distributed neural networks. In *IJCAI*, 2019.

Puyudi Yang, Jianbo Chen, Cho-Jui Hsieh, Jane-Ling Wang, and Michael I Jordan. Greedy attack and gumbel attack: Generating adversarial examples for discrete data. *Journal of Machine Learning Research*, 21(43):1–36, 2020.

Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International Conference on Machine Learning*, pp. 5650–5659. PMLR, 2018.

Dun Zeng, Siqi Liang, Xiangjing Hu, Hui Wang, and Zenglin Xu. Fedlab: A flexible federated learning framework. *Journal of Machine Learning Research*, 24(100):1–7, 2023.

Jiale Zhang, Bing Chen, Xiang Cheng, Huynh Thi Thanh Binh, and Shui Yu. Poisongan: Generative poisoning attacks against federated learning in edge computing systems. *IEEE Internet of Things Journal*, 8(5):3310–3322, 2020.