Poisoning with A Pill: Circumventing Detection in Federated Learning

Hanxi Guo[1*], Hao Wang[2], Tao Song[3†], Tianhang Zheng[4], Yang Hua[5], Haibing Guan[3], Xiangyu Zhang[1]

[1]Purdue University
[2]Stevens Institute of Technology
[3]Shanghai Jiao Tong University
[4]Zhejiang University
[5]Queen's University Belfast
guo778@purdue.edu, hwang9@stevens.edu, songt333@sjtu.edu.cn, zthzheng@zju.edu.cn, Y.Hua@qub.ac.uk, hbguan@sjtu.edu.cn, xyzhang@purdue.edu

Abstract

Federated learning (FL) protects data privacy by enabling distributed model training without direct access to client data. However, its distributed nature makes it vulnerable to model and data poisoning attacks. While numerous defenses filter malicious clients using statistical metrics, they overlook the role of model redundancy, where not all parameters contribute equally to the model and attack performance. Current attacks manipulate all model parameters uniformly, making them more detectable, while defenses focus on the overall statistics of client updates, leaving gaps for more sophisticated attacks. We propose an attack-agnostic augmentation method to enhance the stealthiness and effectiveness of existing poisoning attacks in FL, exposing flaws in current defenses and highlighting the need for fine-grained FL security. Our three-stage methodology, including *pill construction*, *pill poisoning*, and *pill injection*, injects poison into a compact subnet (*i.e.*, pill) of the global model during the iterative FL training. Experimental results show that FL poisoning attacks enhanced by our method can bypass 8 state-of-the-art (SOTA) defenses, gaining an up to 7x error rate increase, as well as on average a more than 2x error rate increase on both IID and non-IID data, in both cross-silo and cross-device FL systems.

Extended version — https://arxiv.org/abs/2407.15389

1 Introduction

With the rising demand for machine learning and cloud computing, Federated Learning (FL)(Konečnỳ et al. 2016; McMahan et al. 2017) has emerged as a key approach for training models on distributed data from scattered clients. Unlike centralized machine learning, FL avoids direct data access, reducing communication overhead and enhancing privacy. However, its distributed nature leaves it vulnerable when clients are compromised. Numerous studies (Baruch, Baruch, and Goldberg 2019; Fang et al. 2020; Bhagoji et al. 2019; Shejwalkar and Houmansadr 2021; Cao and Gong 2022; Bagdasaryan et al. 2020) have examined *poisoning attacks*, where malicious clients manipulate the global model. These fall into two categories: 1) *Model poisoning* which directly alters local updates to skew global parameters (Fang et al. 2020; Shejwalkar and Houmansadr 2021); and 2) *Data poisoning* that injects malicious samples into local datasets (Bagdasaryan et al. 2020; Tolpegin et al. 2020; Xie et al. 2020; Sun et al. 2019; Wang et al. 2020; Chen et al. 2017; Liu et al. 2018; Qi et al. 2022). Such attacks threaten FL's integrity and reliability (Lyu, Yu, and Yang 2020; Kairouz et al. 2021).

To mitigate these attacks, defenses have been proposed, including *adaptive client filtering* (Blanchard et al. 2017; Cao et al. 2021; Xu et al. 2021; Nguyen et al. 2022; Yan et al. 2023), *statistical parameter aggregation* (Yin et al. 2018; Guerraoui, Rouault et al. 2018; Fung, Yoon, and Beschastnikh 2018; Panda et al. 2022; Han et al. 2023), *client-dominant detection* (Guo et al. 2021, 2024; Sun et al. 2021a; Zhang et al. 2023b; Zhu, Roos, and Chen 2023; Park et al. 2023), and *other advanced metrics and pipelines* (Xie, Koyejo, and Gupta 2019; Xie et al. 2021; Cao et al. 2023, 2022; Zhang et al. 2022a). These approaches aim to identify suspicious updates, which are usually evident in model poisoning attacks that uniformly alter parameters.

We argue that modifying all parameters uniformly is not a cost-effective approach. Studies on model pruning (Frankle and Carbin 2018; Lin et al. 2018; Han, Mao, and Dally 2015; Mugunthan et al. 2022; Jiang et al. 2022) show that parameters do not contribute equally to a model's performance. Altering *redundant* parameters wastes resources and reduces attack *stealthiness*. A more effective strategy is to target *critical* parameters (Zhang et al. 2023a), which significantly impact performance, thereby increasing the attack's effectiveness while maintaining stealthiness. Thus, we propose a novel attack-agnostic augmentation method that enhances model poisoning attacks using a three-stage pipeline: *pill construction*, *pill poisoning*, and *pill injection*. In the first stage, we design a pill blueprint and identify its corresponding subnet instance in the target model. During *pill poisoning*, existing FL attacks are applied in an attack-agnostic manner to poison the selected pill. Finally, in *pill injection*,

[*]Work done while Hanxi Guo was a master's student at SJTU, prior to joining Purdue University.

[†]Tao Song is the corresponding author.

the poisoned pill is inserted into an estimated benign update, and a two-step adjustment is used to minimize the difference between the poisoned and benign updates. This approach dynamically generates, poisons, and injects a pill into the global model, augmenting existing FL poisoning attacks.

We conduct extensive experiments to evaluate the effectiveness of our augmentation method. We apply it to four baseline poisoning attacks: sign-flipping attack, trim attack (Fang et al. 2020), krum attack (Fang et al. 2020), and min-max attack (Shejwalkar and Houmansadr 2021). Using both the original and augmented versions, we measure error rates (*i.e.*, the proportion of incorrect predictions) of the global model trained with nine aggregation rules: FedAvg (McMahan et al. 2017), FLTrust (Cao et al. 2021), Multi-Krum (Blanchard et al. 2017), Median (Yin et al. 2018), Trim (Yin et al. 2018), Bulyan (Guerraoui, Rouault et al. 2018), FLDetector (Zhang et al. 2022a), DnC (Shejwalkar and Houmansadr 2021), and Flame (Nguyen et al. 2022). These aggregation rules represent most existing defense metrics. We also design an adaptive defense where the defender has full knowledge of our pipeline and implementation. Results show our method substantially improves existing FL poisoning attacks, leading to over a 2x average increase in model prediction error rates under existing defenses, and up to a 7x increase in some cases.

Our contributions are summarized as follows:

- We propose a generic, attack-agnostic augmentation method that enhances poisoning attacks against robust FL by encapsulating model poisoning attacks into well-defined subnets (*i.e.*, pills) with comprehensive metric-based adjustments.
- Extensive experiments on three common datasets against nine aggregation rules demonstrate that our method helps baseline attacks bypass almost all existing defenses, which cannot be attacked by original versions.
- We identify limitations of existing poisoning attacks and defenses in FL, highlighting the need and potential for fine-grained FL security.

2 Background and Related Work

2.1 Federated Learning

Federated Learning (FL) (Konečnỳ et al. 2016; McMahan et al. 2017) trains a global model using the information from a swarm of clients without the direct access to each client's data. In a standard FL training process, within an arbitrary communication round t, the FL server first distributes its global model $\boldsymbol{g}_t$ to all the clients K. After receiving this global model, each client i trains a local model $\boldsymbol{g}_t^{(i)}$ with its local data $D^{(i)}$, and uploads the model update $\Delta \boldsymbol{g}_t^{(i)}$ to the FL server. After receiving the model updates from the clients, the FL server uses aggregation rules to calculate the global model $\boldsymbol{g}_{t+1}$ for the next round. The objective of FL can be formulated as:

$$\min_{\boldsymbol{g}} \sum_{i=0}^{K} \frac{|D^{(i)}|}{|D|} \cdot f(D^{(i)}, \boldsymbol{g}). \tag{1}$$

2.2 Poisoning Attacks in FL

Following prior studies (Shejwalkar et al. 2022; Khan et al. 2023; Jere, Farnan, and Koushanfar 2020), poisoning attacks in Federated Learning (FL) fall into two categories: *model poisoning* and *data poisoning*. In *model poisoning attacks*, adversaries compromise the global model by directly altering local model updates (Baruch, Baruch, and Goldberg 2019; Fang et al. 2020; Shejwalkar and Houmansadr 2021; Cao and Gong 2022; Bhagoji et al. 2019). In *data poisoning attacks*, they corrupt local datasets to indirectly affect the global model (Tolpegin et al. 2020; Bagdasaryan et al. 2020; Xie et al. 2020; Sun et al. 2019; Wang et al. 2020; Zhang et al. 2022b). Further details are in the extended version.

Our pill design draws inspiration from the *subnet replacement attack* (SRA) (Qi et al. 2022), a backdoor injection method that limits backdoors to a small subnetwork. SRA trains this subnet with tainted data, substitutes the target model's parameters, and disconnects the subnet to maintain attack effect. Drawing from SRA's stealthy yet potent design, we propose a heterogeneous-width *pill blueprint* for varied FL poisoning attacks. Unlike SRA's one-time injection, our approach gradually poisons the global model during training, improving robustness against various defenses.

2.3 Defenses against Poisoning Attacks in FL

Existing defenses can be categorized based on the mitigation strategies that they utilize, including *Adaptive Client Filtering*, *Statistical Parameter Aggregation*, *Client-dominant Detection*, and *Other Advanced Metrics and Pipelines*. To comprhensively evaluate our method, we use *Multi-Krum (MKrum)* (Blanchard et al. 2017), *Trimmed Mean (Trim)* (Yin et al. 2018), *Coordinate-wise Median (Median)* (Yin et al. 2018), *Bulyan* (Guerraoui, Rouault et al. 2018), *FLTrust* (Cao et al. 2021), *FLDetector (FLD)* (Zhang et al. 2022a), *DnC* (Shejwalkar and Houmansadr 2021), and *Flame* (Nguyen et al. 2022), a set of representative defenses, as our baselines. More details are in the extended version.

2.4 Threat Model

We follow the typical threat model used in existing studies (Fang et al. 2020; Shejwalkar and Houmansadr 2021), where the attacker has access to a subset of compromised clients and aims to increase the error rates of the global model on specific classes or across all classes. In this scenario, defenses cannot directly analyze the data on each client as the defender's setting in (Blanchard et al. 2017; Yin et al. 2018; Cao et al. 2021; Guo et al. 2021). Instead, they identify malicious clients by analyzing the uploaded client updates. Further details are in the extended version.

3 Design Objectives and Challenges

After analyzing the drawbacks and various implementations of existing FL poisoning attacks, we define three main objectives for our attack augmentation method: 1) For *stealthiness*, the augmentation method should stay stealthy while achieving comparable performance with original attacks. 2) For *compatibility*, the augmentation should be compatible with most of the existing FL poisoning attacks with few

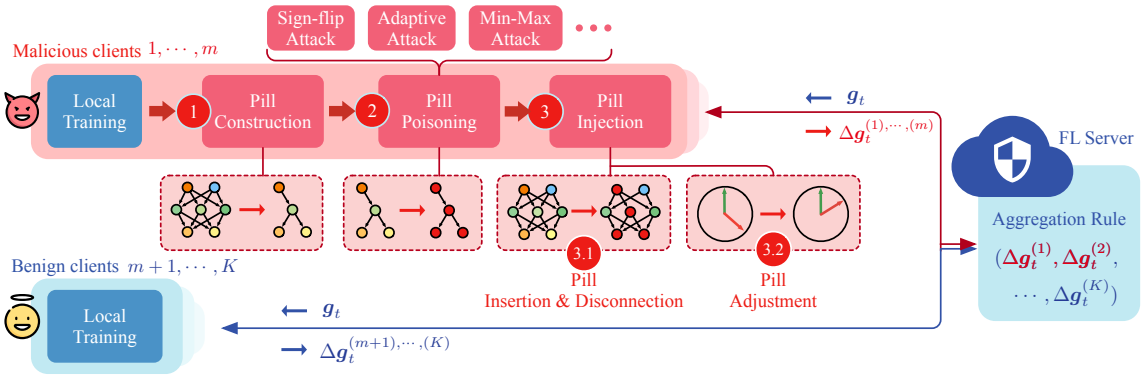

Figure 1: Overview of our augmentation method. The red parts indicate our augmentation method's contribution, and the cyan parts represent the standard federated learning architecture.

modifications on their implementations. 3) For *generality*, the attack augmentation should be able to bypass general detection methods with different detection metrics.

Corresponding to each objective, three challenges need to be addressed:

- It presents a significant challenge that the attack augmentation method must use significantly fewer parameters while still achieving similar attack effects.
- It is challenging to develop a uniform augmentation method for various FL poisoning attacks since they require different information and are implemented in different training stages.
- It is difficult to devise a general strategy that bypasses all common detection approaches, while guaranteeing the attack effectiveness.

4 Design

4.1 Overview of Our Method

We are the first that propose a universal attack augmentation pipeline for most FL poisoning attacks, considering all of the *stealthiness*, *compatibility*, and *generality*. Figure 1 presents the three key stages. In the extended version, we include a comprehensive table of all main symbol notations used in the paper, along with an algorithm table detailing the full workflow of our method.

Stage ①: Pill Construction. It leverages a dynamic subnetwork search algorithm to achieve *stealthiness* by selecting the tiny pill from the global model $\boldsymbol{g}_t$ based on the importance of model parameters. Additionally, six dynamic search patterns are designed to prevent being traced.

Stage ②: Pill Poisoning. In this state, we reapply existing FL poisoning attacks to the selected poison pill, using an extra trained model $\hat{\boldsymbol{g}}_{t+1}^{m}$ (trained on data from the compromised clients) as the attacker's base model. For *compatibility*, we only modify the input of the existing FL poisoning attacks and utilize their outputs, without any interference to their internal implementations. This black-box utilization lets our method be attack-agnostic and compatible with most of the existing FL poisoning attacks.

Stage ③: Poison Pill Injection. It contains pill insertion & disconnection, and pill adjustment. In this stage, our augmentation method injects the poison pill into the estimated benign update $\Delta\widetilde{\boldsymbol{g}}_{t+1}$, and further adjusts the boosting magnitude of both the poison pill parameters and the remaining parameters. We propose a two-step dynamic adjustment to enhance the *generality* of our method against most defenses.

4.2 Pill Construction

This stage aims to construct a pill structure for augmenting the *stealthiness* while retaining the original attack effectiveness. The pill is carefully crafted to only involve a minimal subset of parameters from specific positions of the target model. We first define a pill's blueprint as the pill's graphic structure, independent of target model parameters. Then, we propose a dynamic pill search algorithm to identify and map concrete parameters from the target model to the blueprint.

Designing Pill Blueprint. Inspired by SRA (Qi et al. 2022), which demonstrates that poisoning a narrow subnetwork (one neuron/channel per layer) can effectively implant backdoors in machine learning models (outside the FL setting), we adapt and generalize the concept for FL. The original SRA design is unsuitable for our goals as it employs a fixed, pre-selected subnet tailored to a specific architecture, ignores FL's training dynamics, lacks adaptability to different targets, and produces non-stealthy subnets with disproportionately large weights to propagate the poison through a small network. We introduce a novel blueprint method with a general subnet structure whose instantiations vary dynamically during FL training. A search algorithm selects important neurons at each step, enabling small weight modifications while ensuring effective dissemination and high stealthiness. The blueprint also supports multiple targets by simultaneously manipulating outputs across all relevant neurons. Specifically, our pill blueprint design follows the rules below (more details shown in the extended version):

1. The pill blueprint only contains one neuron in each linear layer or one channel in each convolutional layer, except for the last two layers. Suppose $\mathcal{N}_i^p$ represents the neuron/channel number in Layer i in our pill blueprint, then $\mathcal{N}_i^p = 1$ when $i < L-1$, where L is the total layer counts

in our pill blueprint.

2. In the last two layers of our pill blueprint, $\mathcal{N}_{L-1}^p = \mathcal{N}_L^p = \textit{number of classes}$.

Dynamic Pill Search. According to existing studies on neural network pruning (Frankle and Carbin 2018; Lin et al. 2018; Han, Mao, and Dally 2015; Mugunthan et al. 2022; Jiang et al. 2022), parameters with a larger magnitude typically dominate the model's performance. Thus, the best approach is to analyze the parameters to find a globally optimal pill that includes the most critical parameters.

However, such a globally optimal pill could be identified via a pruning-based method (Wu et al. 2020; Sun et al. 2021b), and hence our attack could be easily detected. Besides, searching for a globally optimal pill is inefficient when the model has a large number of parameters. Thus, we search for an approximate pill instead, with an attacker-defined start point, and only evaluate a small subset of the entire model's parameters. We name the search algorithm as "approximate max pill search". The key idea is to perform a targeted neuron search at each layer by focusing only on the neurons connected to the selected neurons from the previous layer, following a high-sum-of-weights-first principle that prioritizes neurons based on the cumulative sum of their connection weights to the previously selected neurons. The entire search contains four steps:

Step 1. Random Start Point Selection: Randomly select neurons from the first layer of the target model as start points, denoted as $\mathcal{V}_1$, based on the neuron count $\mathcal{N}_1^p$ in pill blueprint's first layer. Such selected start point neurons are fixed throughout the search.

Step 2. Layer-wise Search: For each subsequent layer l_i, we compute the sum of weights connecting neurons in $\mathcal{V}_{i-1}$ to neurons in l_i and rank the neurons in l_i based on the descending order of the sum of weights. Top $\mathcal{N}_i^p$ neurons are chosen for $\mathcal{V}_i$, where $\mathcal{N}_i^p$ denotes the number of neurons in the pill blueprint's i-th layer.

Step 3. Output Neuron Pairing: Pair the selected neurons $\mathcal{V}_{L-1}$ in the final hidden layer with the neurons in the output layer l_L, ensuring a one-to-one correspondence.

Step 4. Pill Mask Construction: Two masks are constructed. M marks the pill parameters in the target model, and M_{disc} records the disconnection locations between the pill and the remained neurons in the target model.

The searched pill ensures both effectiveness and stealthiness in attacks. Detailed information for each step is provided in the extended version, along with a concrete example and an overhead analysis of the pill search algorithm.

4.3 Pill Poisoning

In the **pill poisoning** stage, we aim to condense the poison into the pill using existing attacks. To achieve compatibility, our method simply reuses existing FL poisoning attacks, without any intrusive modification to their original implementations. We only modify the input of existing FL poisoning attacks by replacing the base model update with the update from a model that has undergone extra training rounds, denoted as $\Delta\hat{\boldsymbol{g}}_{\boldsymbol{t+1}}^{\boldsymbol{m}}$. Additionally, we restrict changes to parameters within the pill. The output is a poisoned pill that will be used in the next **pill injection** stage.

The motivation to use an extra-trained model update as the reference model update is shown in Figure 2. As shown in the figure, with the increasing number of extra training rounds on the malicious clients, the generated malicious model update becomes less opposite to the FLTrust (Cao et al. 2021) server's model update. Thus, we adopt the extra training in our method and limit the extra training epoch number E_{extra} to less than the number of malicious clients m times the benign local training epoch number E, denoted as $E_{extra} \leq m \cdot E$. This constraint ensures compliance with the threat model, as the attacker can utilize the data and computational resources of compromised clients.

4.4 Pill Injection

In the **pill injection** stage, we aim to inject the pill into the model and use a two-step adjustment method to further camouflage the pill. Thus, the entire injection stage could be divided into two parts - pill injection and camouflaging. After this stage, the poison pill is seamlessly integrated with the benign model update and uploaded to the FL server.

Pill Insertion & Disconnection. In this part, our goal is to insert the pill into the model, and minimize the impact of the benign model updates on our pill. We use an estimated global model update as the benign model update, which is estimated as the coordinate-wise mean values of all the normal model updates from the compromised clients. The estimation process is hence presented as Equations (2) (`Estimation()` in Algorithm in the extended version),

$$\Delta\widetilde{\boldsymbol{g}}_{t+1} \leftarrow mean\{\Delta\boldsymbol{g'}_{t+1}^{(1),\cdots,(m)}\}, \tag{2}$$

where $\Delta\boldsymbol{g'}_{t+1}^{(i)}$ is the normal updates from the compromised clients. By aggregating information from multiple malicious clients, the estimated global model update is more similar to the genuine one, providing more budget for our poison pill.

After obtaining the estimated global model update $\boldsymbol{\Delta\widetilde{g}}_{t+1}$, we directly replace the parameters corresponding to the pill parameters (which have been poisoned in the previous stage) via the pill's mask $\boldsymbol{M}$. Then, we replace the parameters that connect the pill and the other estimated global model updates with *the disconnection update* $\Delta\boldsymbol{g}_{t+1}^{zero}$, using the disconnection mask $\boldsymbol{M}_{disc}$. The disconnection update $\Delta\boldsymbol{g}_{t+1}^{zero}$ is calculated as $0 - \boldsymbol{g}_t$, and is bounded by the maximum and minimum values of the reference model update $\Delta\hat{\boldsymbol{g}}_{\boldsymbol{t+1}}^{\boldsymbol{m}}$. The disconnection update gradually reduces the connection parameters between the pill and the rest of the model to 0, and finally isolates the poison pill from the global model, guaranteeing the attacking effects.

Pill Adjustment. After the injection, we use a two-step adjustment to further adjust the pill, improving the generality against multiple detection metrics simultaneously. In this stage, we consider two prevailing detection metrics – distance and cosine similarity. To increase the cosine similarity between the poisoned model update and the benign model update in our method, we balance the magnitudes of both the poison pill's parameters and the other benign parameters. Similarly, to minimize the distance discrepancy between the

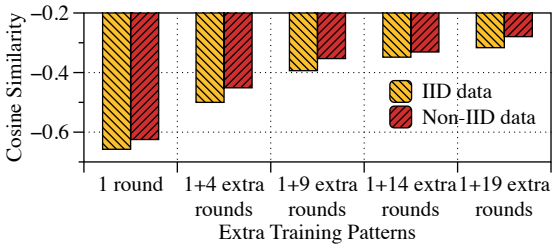

Figure 2: Cosine similarity between FLTrust server and malicious updates with different extra local epochs.

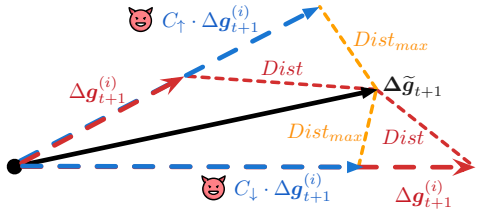

Figure 3: Intuition behind distance-based adjustment in our augmentation method.

Data Distribution	**IID**							**Non-IID**						
Attack	**FedAvg**	**FLTrust**	**MKrum**	**Bulyan**	**Median**	**Trim**	**FLD**	**FedAvg**	**FLTrust**	**MKrum**	**Bulyan**	**Median**	**Trim**	**FLD**
No Attack	0.109	0.107	0.105	0.105	0.123	0.106	0.115	0.113	0.115	0.115	0.112	0.142	0.115	0.122
Sign-Flipping	**0.943**	0.114	0.108	0.126	0.136	0.116	0.118	**0.917**	**0.126**	0.117	0.132	0.152	0.124	0.127
+ Poison Pill	0.667	**0.115**	**0.764**	**0.379**	**0.523**	**0.314**	**0.646**	0.543	0.122	**0.754**	**0.430**	**0.522**	**0.311**	**0.688**
Trim Attack	0.243	0.109	0.139	0.146	0.174	0.179	**0.116**	0.332	0.120	0.201	0.163	0.231	**0.238**	0.124
+ Poison Pill	**0.618**	**0.576**	**0.638**	**0.284**	**0.453**	**0.219**	0.115	**0.668**	**0.517**	**0.687**	**0.292**	**0.473**	0.223	**0.222**
Krum Attack	0.116	0.109	0.189	0.201	0.172	0.137	**0.786**	0.128	0.116	0.235	0.276	0.217	0.160	**0.947**
+ Poison Pill	**0.735**	**0.155**	**0.715**	**0.422**	**0.578**	**0.310**	0.637	**0.716**	**0.151**	**0.737**	**0.468**	**0.730**	**0.334**	0.690
Min-Max Attack	0.183	0.110	0.431	**0.330**	0.183	0.218	**0.825**	0.269	0.125	**0.619**	**0.434**	0.255	0.278	**0.831**
+ Poison Pill	**0.702**	**0.303**	**0.668**	0.327	**0.514**	**0.314**	0.778	**0.629**	**0.320**	0.612	0.406	**0.547**	**0.376**	0.822

Table 1: Error rates on Fashion-MNIST under cross-silo setting with 20% malicious clients in the 50-client FL system.

poisoned and benign model updates, we adjust the magnitude of the entire poisoned model update, as shown in Figure 3. Thus, we first use the **similarity-based adjustment**, then use the **distance-based adjustment**, balancing the *effectiveness* and the *stealthiness* of the poisoned model update. This two-step adjustment is particularly effective when combined with our method, which selectively poisons only a tiny subset of the model's parameters. By altering just a few parameters, our method preserves a substantial number of benign parameters, which are crucial for making effective adjustments. As a result, the poisoned model update can bypass a wide range of defenses since they are typically designed based on the combination or variants of distance and cosine similarity metrics, and they usually do not anticipate such a focused and minimal interference in the model parameters. More details are shown in the extended version.

5 Evaluation

This section evaluates how our method enhances existing FL poisoning attacks from three perspectives. First, we assess its *Augmentation Effectiveness* against four poisoning attacks using nine state-of-the-art defenses across three datasets (Section 5.2). Second, we examine its *Stealthiness* under two detection metrics (Section 5.3). Finally, we conduct a *Generality Analysis*, testing different malicious client proportions, both cross-silo and cross-device settings, and various pill search rules (Section 5.4). Our method substantially strengthens existing attacks, bypassing all nine defenses in over 90% of cases and raising error rates by up to seven times compared to the originals. It remains effective across diverse data distributions, model architectures, malicious client proportions, and pill search rules.

5.1 Evaluation Settings

In our experiments, the malicious client proportion is set to 20% by default. We assess 9 standard aggregation rules, including FedAvg, FLTrust, Multi-Krum, Bulyan, Median, Trim, FLDetector, DnC, and Flame, alongside 4 notable model poisoning attacks: sign-flipping, Trim attack, Krum attack, and Min-Max. Experiments utilize 50 clients for MNIST (LeCun 1998) and Fashion-MNIST (Xiao, Rasul, and Vollgraf 2017), and 30 for CIFAR-10 (Krizhevsky, Hinton et al. 2009), accommodating cross-silo and cross-device scenarios. Implementation is done in PyTorch (Paszke et al. 2019). More details are provided in the extended version.

5.2 Augmentation Effectiveness

This section provides a detailed analysis of our method's augmentation effect on Fashion-MNIST in a 50-client cross-silo FL system with 20% malicious clients. We test our method on both IID and non-IID data, resulting in an average error rate increase of over 0.25 for all baseline attacks, showing our method's *effectiveness* and *compatibility*.

Results on IID Data. The error rates of four baseline FL poisoning attacks, with and without our method, are shown in the left half of Table 1. Our method enhances the error

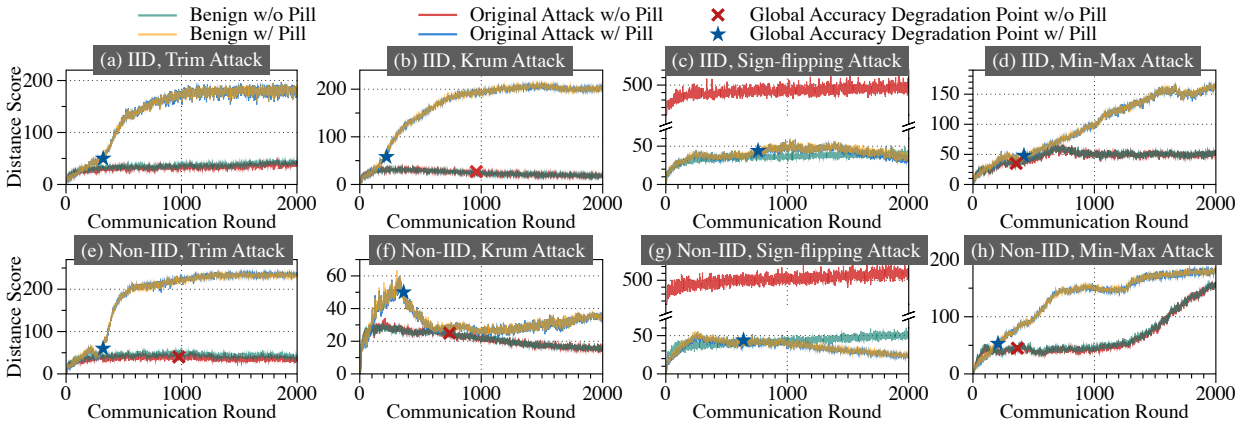

Figure 4: Comparison of Multi-Krum distance score between benign updates and malicious updates when using original poisoning attacks with and without our method.

Data Distribution	**IID**							**Non-IID**						
Attack	**FedAvg**	**FLTrust**	**MKrum**	**Bulyan**	**Median**	**Trim**	**FLD**	**FedAvg**	**FLTrust**	**MKrum**	**Bulyan**	**Median**	**Trim**	**FLD**
No Attack	0.106	0.104	0.103	0.108	0.127	0.107	0.116	0.111	0.119	0.113	0.113	0.140	0.114	0.123
Sign-Flipping	**0.964**	0.109	0.108	0.110	0.130	0.108	0.117	**0.909**	0.119	0.114	0.119	0.144	0.120	0.125
+ Poison Pill	0.320	**0.116**	**0.162**	**0.151**	**0.323**	**0.148**	**0.699**	0.269	**0.120**	**0.239**	**0.164**	**0.364**	**0.168**	**0.242**
Trim Attack	0.112	0.111	0.111	0.115	0.132	0.114	0.116	0.125	0.115	0.121	0.125	0.153	0.122	0.122
+ Poison Pill	**0.508**	**0.139**	**0.334**	**0.126**	**0.284**	**0.127**	**0.120**	**0.528**	**0.148**	**0.455**	**0.143**	**0.287**	**0.146**	**0.136**
Krum Attack	0.107	0.108	0.114	0.123	0.141	0.112	**0.668**	0.116	0.117	0.124	0.138	0.173	0.122	0.410
+ Poison Pill	**0.183**	**0.118**	**0.283**	**0.161**	**0.362**	**0.146**	0.631	**0.428**	**0.127**	**0.280**	**0.187**	**0.415**	**0.182**	**0.704**
Min-Max Attack	0.117	0.108	0.118	0.135	0.142	0.128	0.111	0.124	0.119	0.142	**0.166**	0.162	0.145	0.136
+ Poison Pill	**0.439**	**0.129**	**0.361**	**0.136**	**0.343**	**0.150**	**0.715**	**0.521**	**0.136**	**0.339**	0.153	**0.368**	**0.184**	**0.335**

Table 2: Error rates on Fashion-MNIST under cross-silo setting with 10% malicious clients in the 50-client FL system.

rates of the existing poisoning attacks in 23 out of 28 scenarios, against FedAvg and five defenses. The maximum increase in error rate is 0.658, and the average increase reaches 0.274. This substantial elevation from the attack-free baseline error rate of 0.109 underscores our method's capability to significantly compromise existing defenses' integrity.

Results on Non-IID Data. Evaluations on non-IID data further validate the effectiveness of our method, demonstrating its superiority in 23 of 28 cases. The highest error rate increase reaches 0.637, with an average increase of 0.281. Although there is a slight reduction in the maximal error rate increase in the non-IID setting, these results still demonstrate our method's ability to effectively enhance attacks in more complex and heterogeneous data environments.

All attacks augmented by our method can bypass all baseline defenses, including FLTrust and FLDetector, with the exception of the sign-flipping attack. Notably, the Min-Max attack demonstrates superior effectiveness in non-IID data settings, achieving significant improvements compared to its performance on IID data. Other attacks also exhibit similar error rate improvements relative to their results on IID data, indicating that our method maintains its robustness and effectiveness in more complex data environments. More detailed analyses are presented in the extended version.

5.3 Stealthiness Analysis

To further assess our method, we examine its *stealthiness* during FL training, focusing on its impact on the distance and cosine similarity scores of existing poisoning attacks. Results show that our method can make malicious clients appear as benign, or even more "benign" than genuine clients, owing to its pill design with distance-based and similarity-based adjustments. As shown in Figure 4, the average distance scores of malicious clients (with our method) across four baseline model poisoning attacks closely match or even coincide with those of benign clients throughout training. Detailed results, including similarity score analyses, are provided in the extended version.

5.4 Generality Analysis

In this section, we further discuss the *generality* of our method across three key factors: malicious client proportion, client participation frequency, and datasets & model architectures. The results indicate that our method consistently

Distribution	IID									Non-IID								
Attack	**FAvg**	**FLT**	**MKr**	**Bulyan**	**Med**	**Trim**	**DnC**	**FLD**	**Flame**	**FAvg**	**FLT**	**MKr**	**Bulyan**	**Med**	**Trim**	**DnC**	**FLD**	**Flame**
No Attack	0.48	0.48	0.50	0.46	0.55	0.45	0.44	0.49	0.49	0.48	0.47	0.49	0.49	0.58	0.52	0.46	0.50	0.53
Sign-Flipping	**0.89**	0.47	0.58	0.53	0.62	0.46	0.46	0.49	0.50	**0.90**	0.51	0.51	0.62	0.65	0.57	0.50	0.60	0.53
+ Poison Pill	0.73	**0.88**	**0.92**	**0.69**	**0.70**	**0.69**	**0.53**	**0.89**	**0.70**	0.87	**0.86**	**0.89**	**0.67**	**0.76**	**0.68**	**0.56**	**0.90**	**0.67**
Trim ATK	0.48	0.50	0.48	0.53	0.62	0.51	0.45	0.45	0.50	0.57	0.49	0.60	0.59	0.65	0.54	0.48	0.48	0.50
+ Poison Pill	**0.85**	**0.87**	**0.88**	**0.65**	**0.67**	**0.66**	**0.51**	**0.89**	**0.54**	**0.89**	**0.86**	**0.90**	**0.77**	**0.68**	**0.63**	**0.51**	**0.89**	**0.62**
Krum ATK	0.47	0.54	0.47	0.56	0.54	0.51	0.45	0.80	0.50	0.48	0.50	0.49	0.52	0.64	0.51	0.48	**0.89**	0.50
+ Poison Pill	**0.70**	**0.89**	**0.90**	**0.76**	**0.75**	**0.64**	**0.52**	**0.89**	**0.87**	**0.72**	**0.84**	**0.90**	**0.67**	**0.74**	**0.64**	**0.58**	0.88	**0.87**
Min-Max ATK	0.45	0.50	0.46	0.50	0.57	0.46	0.51	0.52	0.52	0.47	0.50	0.49	0.56	0.63	0.60	0.47	0.48	0.48
+ Poison Pill	**0.75**	**0.71**	**0.90**	**0.77**	**0.80**	**0.64**	**0.54**	**0.90**	**0.81**	**0.66**	**0.64**	**0.88**	**0.67**	**0.78**	**0.66**	**0.52**	**0.90**	**0.79**

Table 3: Error rates on CIFAR-10 under cross-silo setting with 20% malicious clients in the 30-client FL system.

maintains its effectiveness despite changes in these conditions, demonstrating its adaptability and broad applicability in augmenting existing attacks.

Impact of The Malicious Client's Proportion. We first assess the effectiveness of our method in both IID and non-IID cross-silo FL systems with only 10% of clients compromised, as shown in Table 2. This setup reveals that all baseline model poisoning attacks yield lower error rates on the global model compared with scenarios with 20% compromised clients. While the increase in error rates is less than those in the 20% compromised client scenario, our method still effectively raises the global model's error rates in 25/26 out of 28 cases (IID/non-IID setting). The maximum increase in error rates reaches 0.403, with an average increase of 0.144. This average is notably higher (¿2x higher) than the error rates observed in attack-free FL conditions. Specifically, our method helps sign-flipping/Trim/Krum/Min-Max attacks achieve an average error rate increase of 0.133/0.094/0.136/0.209. More detailed results are presented in the extended version.

Impact of The Client Participation Frequency. We then extend the evaluation of our method to a cross-device FL system, where only 40% of clients are selected for participation in each communication round. This setup results in less frequent participation from each client and a fluctuating proportion of malicious clients across different rounds. The maximum error rate increase with our method is 0.639, with an average increase across different attacks and defenses of 0.279. These results are consistent with those from the cross-silo FL system, underscoring our method's effectiveness and *generality* across different FL configurations. This evaluation demonstrates our method's robust performance and adaptability, not only in a controlled cross-silo environment but also under the more various conditions in cross-device FL systems. More details are presented in the extended version.

Impact of The Datasets and Model Architectures. Following the evaluation with the Fashion-MNIST dataset, we test our method on the MNIST and CIFAR-10 dataset, employing the four-layer CNN model and the AlexNet model to further verify our method's *generality* across different datasets. The collective results show that our method performs even better with larger datasets or more complex machine learning models. This trend confirms the *generality* of our method by revealing its capability to maintain consistent performance enhancements regardless of the dataset or model complexity involved. Specifically, our method helps all four baseline attacks bypass all nine baselines on CIFAR-10 dataset, achieving 0.288 error rate increase on average, presented in Table 3. More detailed results on MNIST dataset are shown in the extended version.

Beyond the three key factors, we also investigate the impact of the pill search algorithm in the extended version. The results indicate that the "approximate max pill search" algorithm outperforms the "approximate min pill search" in 41 out of 56 cases ($\sim$73%), highlighting its effectiveness in selecting the most influential parameters to maximize attack impact. Additional findings on ablation studies and generalizability are also provided in the extended version.

6 Discussion

To further evaluate the robustness of our method when defenses are aware of the attack strategies (white-box scenario), we design an adaptive defense and present the experimental details in the extended version. Despite the adaptive defense's attempt to incorporate both cosine similarity and distance metrics, it remains insufficient to thwart the enhanced capabilities of our method. We also presented a detailed discussion of the limitations and future directions in the extended version.

7 Conclusion

We propose a novel attack-agnostic augmentation method to enhance FL poisoning attacks by concentrating them into a tiny subnet (*i.e.*, *pill*). Our approach comprises *pill construction*, *pill poisoning*, and *pill injection*, enabling existing FL poisoning attacks to achieve over $2\times$ higher error rates on average. By intensifying the inherent vulnerabilities in current FL defenses, our method underscores the urgent need for more robust and fine-grained detection mechanisms.

Acknowledgements

We thank the anonymous reviewers for their valuable feedback. This work was supported in part by the National Natural Science Foundation of China (NO. 62472284, 62572426), Shanghai Key Laboratory of Scalable Computing and Systems. The work of H. Wang was supported in part by the United States National Science Foundation (NSF) under grants 2534286, 2523997, 2315612, and 2332638.

References

Bagdasaryan, E.; Veit, A.; Hua, Y.; Estrin, D.; and Shmatikov, V. 2020. How to Backdoor Federated Learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Baruch, G.; Baruch, M.; and Goldberg, Y. 2019. A Little Is Enough: Circumventing Defenses for Distributed Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Bhagoji, A. N.; Chakraborty, S.; Mittal, P.; and Calo, S. 2019. Analyzing Federated Learning through An Adversarial Lens. In *International Conference on Machine Learning (ICML)*.

Blanchard, P.; El Mhamdi, E. M.; Guerraoui, R.; and Stainer, J. 2017. Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Cao, X.; Fang, M.; Liu, J.; and Gong, N. Z. 2021. FLTrust: Byzantine-robust Federated Learning via Trust Bootstrapping. In *Network and Distributed System Security (NDSS) Symposium*.

Cao, X.; and Gong, N. Z. 2022. Mpaf: Model Poisoning Attacks to Federated Learning Based on Fake Clients. In *the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Cao, X.; Jia, J.; Zhang, Z.; and Gong, N. Z. 2023. Fedrecover: Recovering from Poisoning Attacks in Federated Learning using Historical Information. In *IEEE Symposium on Security and Privacy (S&P)*.

Cao, X.; Zhang, Z.; Jia, J.; and Gong, N. Z. 2022. Flcert: Provably Secure Federated Learning against Poisoning Attacks. *IEEE Transactions on Information Forensics and Security (TIFS)*, 3691–3705.

Chen, X.; Liu, C.; Li, B.; Lu, K.; and Song, D. 2017. Targeted Backdoor Attacks on Deep Learning Systems using Data Poisoning. *arXiv preprint arXiv:1712.05526*.

Fang, M.; Cao, X.; Jia, J.; and Gong, N. Z. 2020. Local Model Poisoning Attacks to Byzantine-Robust Federated Learning. In *USENIX Security Symposium (USENIX Security)*.

Frankle, J.; and Carbin, M. 2018. The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. *arXiv preprint arXiv:1803.03635*.

Fung, C.; Yoon, C. J.; and Beschastnikh, I. 2018. Mitigating Sybils in Federated Learning Poisoning. *arXiv preprint arXiv:1808.04866*.

Guerraoui, R.; Rouault, S.; et al. 2018. The Hidden Vulnerability of Distributed Learning in Byzantium. In *International Conference on Machine Learning (ICML)*.

Guo, H.; Wang, H.; Song, T.; Hua, Y.; Lv, Z.; Jin, X.; Xue, Z.; Ma, R.; and Guan, H. 2021. Siren: Byzantine-robust Federated Learning via Proactive Alarming. In *ACM Symposium on Cloud Computing (SoCC)*.

Guo, H.; Wang, H.; Song, T.; Hua, Y.; Ma, R.; Jin, X.; Xue, Z.; and Guan, H. 2024. Siren+: Robust Federated Learning with Proactive Alarming and Differential Privacy. *IEEE Transactions on Dependable and Secure Computing (TDSC)*.

Han, S.; Mao, H.; and Dally, W. J. 2015. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. *arXiv preprint arXiv:1510.00149*.

Han, S.; Park, S.; Wu, F.; Kim, S.; Zhu, B.; Xie, X.; and Cha, M. 2023. Towards attack-tolerant federated learning via critical parameter analysis. In *the IEEE International Conference on Computer Vision (ICCV)*.

Jere, M. S.; Farnan, T.; and Koushanfar, F. 2020. A Taxonomy of Attacks on Federated Learning. In *IEEE Symposium on Security and Privacy (S&P)*.

Jiang, Y.; Wang, S.; Valls, V.; Ko, B. J.; Lee, W.-H.; Leung, K. K.; and Tassiulas, L. 2022. Model Pruning Enables Efficient Federated Learning on Edge Devices. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*.

Kairouz, P.; McMahan, H. B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A. N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; et al. 2021. Advances and Open Problems in Federated Learning. *Foundations and Trends® in Machine Learning*, 14(1–2): 1–210.

Khan, M. A.; Shejwalkar, V.; Houmansadr, A.; and Anwar, F. M. 2023. On The Pitfalls of Security Evaluation of Robust Federated Learning. In *IEEE Security and Privacy Workshops (SPW)*.

Konečnỳ, J.; McMahan, H. B.; Yu, F. X.; Richtárik, P.; Suresh, A. T.; and Bacon, D. 2016. Federated Learning: Strategies for Improving Communication Efficiency. In *NeurIPS Workshop on Private Multi-Party Machine Learning (PMPML)*.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning Multiple Layers of Features from Tiny Images. Technical report, University of Toronto.

LeCun, Y. 1998. The MNIST Database of Handwritten Digits. *http://yann. lecun. com/exdb/mnist/*.

Lin, Y.; Han, S.; Mao, H.; Wang, Y.; and Dally, B. 2018. Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training. In *International Conference on Learning Representations (ICLR)*.

Liu, Y.; Ma, S.; Aafer, Y.; Lee, W.-C.; Zhai, J.; Wang, W.; and Zhang, X. 2018. Trojaning Attack on Neural Networks. In *Network and Distributed System Security (NDSS) Symposium*.

Lyu, L.; Yu, H.; and Yang, Q. 2020. Threats to Federated Learning: A Survey. *arXiv preprint arXiv:2003.02133*.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient Learning of Deep Networks from Decentralized Data. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Mugunthan, V.; Lin, E.; Gokul, V.; Lau, C.; Kagal, L.; and Pieper, S. 2022. Fedltn: Federated Learning for Sparse and Personalized Lottery Ticket Networks. In *European Conference on Computer Vision (ECCV)*.

Nguyen, T. D.; Rieger, P.; De Viti, R.; Chen, H.; Brandenburg, B. B.; Yalame, H.; Möllering, H.; Fereidooni, H.; Marchal, S.; Miettinen, M.; et al. 2022. FLAME: Taming Backdoors in Federated Learning. In *USENIX Security Symposium (USENIX Security)*.

Panda, A.; Mahloujifar, S.; Bhagoji, A. N.; Chakraborty, S.; and Mittal, P. 2022. SparseFed: Mitigating Model Poisoning Attacks in Federated Learning with Sparsification. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Park, S.; Han, S.; Wu, F.; Kim, S.; Zhu, B.; Xie, X.; and Cha, M. 2023. FedDefender: Client-side attack-tolerant federated learning. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An Imperative Style, High-performance Deep Learning Library. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Qi, X.; Xie, T.; Pan, R.; Zhu, J.; Yang, Y.; and Bu, K. 2022. Towards Practical Deployment-stage Backdoor Attack on Deep Neural Networks. In *the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Shejwalkar, V.; and Houmansadr, A. 2021. Manipulating The Byzantine: Optimizing Model Poisoning Attacks and Defenses for Federated Learning. In *Network and Distributed System Security (NDSS) Symposium*.

Shejwalkar, V.; Houmansadr, A.; Kairouz, P.; and Ramage, D. 2022. Back to The Drawing Board: A Critical Evaluation of Poisoning Attacks on Production Federated Learning. In *IEEE Symposium on Security and Privacy (S&P)*.

Sun, J.; Li, A.; DiValentin, L.; Hassanzadeh, A.; Chen, Y.; and Li, H. 2021a. FL-WBC: Enhancing Robustness against Model Poisoning Attacks in Federated Learning from A Client Perspective. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Sun, J.; Li, A.; Wang, B.; Yang, H.; Li, H.; and Chen, Y. 2021b. Soteria: Provable Defense Against Privacy Leakage in Federated Learning from Representation Perspective. In *the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Sun, Z.; Kairouz, P.; Suresh, A. T.; and McMahan, H. B. 2019. Can You Really Backdoor Federated Learning? *arXiv preprint arXiv:1911.07963*.

Tolpegin, V.; Truex, S.; Gursoy, M. E.; and Liu, L. 2020. Data Poisoning Attacks against Federated Learning Systems. In *European Symposium on Research in Computer Security (ESORICS)*.

Wang, H.; Sreenivasan, K.; Rajput, S.; Vishwakarma, H.; Agarwal, S.; Sohn, J.-y.; Lee, K.; and Papailiopoulos, D. 2020. Attack of The Tails: Yes, You Really Can Backdoor Federated Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Wu, C.; Yang, X.; Zhu, S.; and Mitra, P. 2020. Mitigating Backdoor Attacks in Federated Learning. *arXiv preprint arXiv:2011.01767*.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-mnist: A Novel Image Dataset for Benchmarking Machine Learning Algorithms. *arXiv preprint arXiv:1708.07747*.

Xie, C.; Chen, M.; Chen, P.-Y.; and Li, B. 2021. Crfl: Certifiably Robust Federated Learning against Backdoor Attacks. In *International Conference on Machine Learning (ICML)*.

Xie, C.; Huang, K.; Chen, P. Y.; and Li, B. 2020. Dba: Distributed Backdoor Attacks against Federated Learning. In *International Conference on Learning Representations (ICLR)*.

Xie, C.; Koyejo, S.; and Gupta, I. 2019. Zeno: Distributed Stochastic Gradient Descent with Suspicion-based Fault-tolerance. In *International Conference on Machine Learning (ICML)*.

Xu, J.; Huang, S.-L.; Song, L.; and Lan, T. 2021. Signguard: Byzantine-robust Federated Learning through Collaborative Malicious Gradient Filtering. *arXiv preprint arXiv:2109.05872*.

Yan, P.; Wang, H.; Song, T.; Hua, Y.; Ma, R.; Hu, N.; Haghighat, M. R.; and Guan, H. 2023. SkyMask: Attack-agnostic Robust Federated Learning with Fine-grained Learnable Masks. *arXiv preprint arXiv:2312.12484*.

Yin, D.; Chen, Y.; Kannan, R.; and Bartlett, P. 2018. Byzantine-robust Distributed Learning: Towards Optimal Statistical Rrates. In *International Conference on Machine Learning (ICML)*.

Zhang, C.; Zhou, B.; He, Z.; Liu, Z.; Chen, Y.; Xu, W.; and Li, B. 2023a. Oblivion: Poisoning Federated Learning by Inducing Catastrophic Forgetting. In *IEEE Conference on Computer Communications (INFOCOM)*.

Zhang, K.; Tao, G.; Xu, Q.; Cheng, S.; An, S.; Liu, Y.; Feng, S.; Shen, G.; Chen, P.-Y.; Ma, S.; and Zhang, X. 2023b. FLIP: A Provable Defense Framework for Backdoor Mitigation in Federated Learning. In *International Conference on Learning Representations (ICLR)*.

Zhang, Z.; Cao, X.; Jia, J.; and Gong, N. Z. 2022a. FLDetector: Defending Federated Learning against Model Poisoning Attacks via Detecting Malicious Clients. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*.

Zhang, Z.; Panda, A.; Song, L.; Yang, Y.; Mahoney, M.; Mittal, P.; Kannan, R.; and Gonzalez, J. 2022b. Neurotoxin: Durable Backdoors in Federated Learning. In *International Conference on Machine Learning (ICML)*.

Zhu, C.; Roos, S.; and Chen, L. Y. 2023. LeadFL: Client Self-Defense Against Model Poisoning in Federated Learning. In *International Conference on Machine Learning (ICML)*.