Fake or Compromised? Making Sense of Malicious Clients in Federated Learning

Hamid Mozaffari[1], Sunav Choudhary[2], and Amir Houmansadr[3]

[1] Oracle Labs
[2] Adobe Research
[3] University of Massachusetts Amherst
{hamid.mozaffari@oracle.com, schoudha@adobe.com, amir@cs.umass.edu

Abstract. Federated learning (FL) is a distributed machine learning paradigm that enables training models on decentralized data. The field of FL security against poisoning attacks is plagued with confusion due to the proliferation of research that makes different assumptions about the capabilities of adversaries and the adversary models they operate under. Our work aims to clarify this confusion by presenting a comprehensive analysis of the various poisoning attacks and defensive aggregation rules (AGRs) proposed in the literature, and connecting them under a common framework. To connect existing adversary models, we present a hybrid adversary model, which lies in the middle of the spectrum of adversaries, where the adversary compromises a few clients, trains a generative (e.g., DDPM) model with their compromised samples, and generates new synthetic data to solve an optimization for a stronger (e.g., cheaper, more practical) attack against different robust aggregation rules. By presenting the spectrum of FL adversaries, we aim to provide practitioners and researchers with a clear understanding of the different types of threats they need to consider when designing FL systems, and identify areas where further research is needed.

Keywords: Model Poisoning · Federated learning · Denoising Diffusion Probability Model (DDPM).

1 Introduction

Federated learning (FL) is a machine learning paradigm that enables training models on decentralized data, such as mobile devices or edge devices. In FL, each *client* updates the global model using their local data, and communicate the updated model to the *central server*. Finally, the server aggregates the updates from all clients using an *aggregation rule* (AGR), creating the next version of the global model. This approach allows for the training of models on large-scale, non-iid data without collecting clients' original data.

Fake or compromised? A fork in the literature! FL is susceptible to *poisoning* by malicious clients who aim to hamper the accuracy of the global model by contributing malicious updates during FL's training process. Based

on how the adversary introduces malicious clients in the FL ecosystem, existing works on FL poisoning can be categorized into two major lines of work: 1) a small percentage (<1%) of "actual" clients are *compromised* by an adversary, e.g., by taking control of some compromised mobile devices; 2) a large percentage (>10%) of *fake* clients are created and injected into the FL ecosystem, e.g., by creating Sybil accounts or using botnets. The "compromised" category [3, 16, 19] targets sophisticated, large-scale applications such as Gboard and Siri that have deployed proper protections against Sybil attacks and botnets. However, these attacks require compromising actual FL devices, which is costly in practice.

On the other hand, the "fake" category [6, 9, 14] assumes that the adversary can inject large numbers of fake clients, such as spam bots, into the FL ecosystem. Such (large-scale) fake clients cannot be injected into sophisticated applications such as Gboard and Siri as thoroughly discussed by [20]; however, FL applications built on third-party code/software may be vulnerable to such fake clients.

Introducing a hybrid adversary model. As discussed above, the literature has only evaluated against two extreme adversary models, i.e., all compromised and all fake adversarial clients. We make the case for a *hybrid adversary model* in which the adversary compromises a very small number of actual users, and then uses their data to fabricate a large number of fake clients (who are supposed to be more impactful than oblivious fake clients considered in the literature). Given the quick and broad adoption of FL in various applications, we believe that such hybrid adversary model can be representative of a very large fraction of FL applications in the future.

Under such a hybrid adversary model, we propose a novel model poisoning attack, called *hybrid attack*, that first leverages the data of compromised clients to *generate* more data using state-of-the-art generative models, e.g., the denoising diffusion probabilistic model (DDPM) [10, 17]. The adversary then uses existing state-of-the-art model poisoning attacks to fabricate poisoned model updates for its compromised and fake clients (which are sent to the FL server). DDPM is a generative model that has recently gained attention for its ability to learn the underlying structure of complex data distributions from limited and noisy observations. DDPM is based on the idea of diffusion, which is a process of iterative exchange of information between the data points in order to reveal their underlying structure. Specifically, DDPM uses a diffusion process to transform given input data into a latent representation, which captures the underlying structure of the data. This latent representation can then be used to generate new samples that are similar to the original input data.

One key advantage of DDPM is that it is able to learn the structure of the data distribution from a small number of observations, even in the presence of noise. This makes it particularly useful for applications where the data is limited or noisy, such as in the case of compromised clients in federated learning. By using DDPM to generate new samples from a small number of compromised clients, an adversary is able to craft a malicious update for FL poisoning that is representative of the data distribution of the benign clients.

Empirical evaluations: We provide extensive evaluations of existing attacks as well as our hybrid attacks under various adversary models obtained by combining the spectrum of adversaries and defenders discussed above. We experiment with two datasets, FEMNIST and CIFAR10, in real-world heterogeneous FL settings. In summary, our key contributions are as follows:

- The literature of FL poisoning has forked into two separate lines of work that assume two differing adversary models, i.e., fake and compromised, as introduced earlier. Our work aims to highlight the differences between these two lines of work by contrasting their application scenarios, assumptions, and costs.
- We fill the gap between fake and compromised adversary models by introducing a spectrum of adversary models, which we call hybrid. Through extensive experiments we demonstrate how the hybrid adversary models establish trade-offs between attack accuracy and attack cost in comparison to the fake and compromised models.
- We design and evaluate novel FL poisoning attacks that work under the newly introduced hybrid adversary model. Our attack leverages DDPM to generate poisoning data for fake clients based on the data collected from a small number of compromised clients.

2 Types of Byzantine-robust aggregation rules

The existing Byzantine-robust aggregation rules (AGRs) for federated learning can be categorized into three categories: non-robust AGRs, AGRs agnostic to poisoning attacks, and AGRs that adapt to or are aware of the poisoning attacks in FL ecosystem.

Non-robust AGR: Non-robust aggregation rules, such as federated averaging (FedAvg) [12, 15], do not consider the presence of malicious clients in the federated learning ecosystem. Therefore, such AGRs simply aggregate the model updates received from all clients by computing a non-robust function of the updates. While these approaches are generally simpler and easy to implement, they are vulnerable to model and data poisoning attacks [8, 16, 19, 20].

Robust AGRs agnostic to FL poisoning: Robust AGRs, such as Median [22] and Norm-Bounding [21], are robust in that they aim to reduce the impact of malicious clients' updates. But, they are *agnostic* in that they do not have any knowledge of the specifics of the attacks, e.g., they do not know the number of malicious updates in each round. These rules use techniques from robust statistics, such as outlier removal or clipping the norms of updates, to exclude or mitigate the impact of malicious updates during the aggregation process.

Norm-Bounding AGR [21] bounds the L2 norm of all submitted client updates to a fixed threshold τ, with the intuition that the effective poisoned updates should have high norms. For a threshold τ and an update ∇, if the norm, $||\nabla||_2 > \tau$, ∇ is scaled by $\frac{\tau}{||\nabla||_2}$, otherwise, the update is not changed. The final aggregate is an average of all the updates, scaled or otherwise.

Robust AGRs that adapt to FL poisoning: Adaptive aggregation rules have the advantage of knowing the number of malicious updates in each round for aggregation. These rules use this information to adapt their aggregation process in order to mitigate the impact of malicious updates on the final model.

Blanchard et al. [4] proposed Multi-Krum AGR as a modification to their own Krum AGR. Multi-Krum selects an update using Krum and adds it to a selection set, S. Multi-Krum repeats this for the remaining updates (which remain after removing the update that Krum selects) until S has c updates such that $n - c > 2m + 2$, where n is the number of selected clients and m is the number of compromised clients in a given round. Finally, Multi-Krum averages the updates in S.

Yin et al. [22] proposed Trimmed-Mean that aggregates each dimension of input updates separately. It sorts the values of the j^{th}-dimension of all updates. Then it removes m (i.e., the number of compromised clients) of the largest and smallest values of that dimension, and computes the average of the rest of the values as its aggregate for the dimension j.

3 Distinguishing Fake And Compromised Adversary Models

A poisoning attack is either *data* or *model* poisoning attack: in data poisoning, the adversary can poison only the data on malicious client device, while in model poisoning, the adversary can directly manipulate/poison the model updates of the malicious clients. In this work, we focus on model poisoning, as it is strictly stronger than data poisoning [19, 20]; hence, poisoning in any context refers to model poisoning, unless stated otherwise.

3.1 Adversary with fake clients

In federated learning (FL) systems, an attacker can inject fake clients in order to send arbitrary fake local model updates to the cloud server. This type of attack is more affordable and easier to perform than compromising genuine clients, as the attacker does not need to bypass anti-malware software or evade anomaly detection on the clients' devices. Instead, the attacker can emulate fake clients using open source projects or free software such as android emulators, which can be run on a single machine to emulate multiple instances, i.e., multiple FL clients, significantly reducing the attack cost. Fake clients also offer the advantage of being fully controlled by the attacker, as Android emulators can grant root access to the devices. These factors make model poisoning attacks using fake clients a realistic threat in FL systems.

Cao et al. proposed MPAF [6], a method of attacking FL systems through the injection of fake clients. In MPAF, the attacker selects a randomly initialized model as the base model (θ'), whose test accuracy is close to random guessing, and crafts fake local model updates to force the global model to mimic the base model. This is done by subtracting the current global model parameters (θ^t for

the FL round t) from the base model parameters and scaling the fake local model updates by a factor λ to amplify their impact. Equation 1 shows the malicious updates of the fake clients.

$$\theta^t_{m\in[M]} = \lambda(\theta' - \theta^t) \tag{1}$$

where $\theta_{m\in[M]}$ are the malicious model updates for M injected fake clients, and θ' is the randomly initialized base model.

To perform MPAF, the attacker must have minimum knowledge of the FL system, which means that they only have access to the global models received during training. Despite this limited information, MPAF is able to effectively manipulate the global model by driving it towards the base model in each FL round. This is done by calculating fake local model updates ($\theta_{m\in[M]}$), which are then aggregated by the cloud server along with genuine local model updates from genuine clients. The attacker can choose a large λ to ensure that the attack is effective even after aggregation.

In our paper, we refer to this attack as the Fake attack. This attack is characterized by the minimal knowledge and ability required from the adversary who controls the fake clients. Specifically, the fake attack is the simplest attack of this kind in FL and represents one end of the spectrum of attacks based on the impact and cost of the attack.

3.2 Adversary with compromised clients

To evaluate the robustness of various FL algorithms, we use state-of-the-art model poisoning attacks from [19]. The attack proposes a general FL poisoning framework and then tailors it to specific FL settings. First, it computes an average $\theta^{b,t} = f_{\text{avg}}(\theta^t_{c\in[C]})$ of benign updates, $\theta^t_{c\in[C]}$, available to the adversary in the FL round t. Then it perturbs $\theta^{b,t}$ in a *dynamic, data-dependent malicious direction* ω to calculate the final poisoned update $\theta^{t,m}_{c\in[C]} = \theta^{b,t} + \gamma\omega$. The attack, called *DYN-OPT*, finds the largest γ that successfully circumvents the target AGR. DYN-OPT is much stronger than its predecessors, because it finds the largest γ and uses a tailored dataset ω. In the following, we detail the DYN-OPT attacks against the AGRs from Section 2 that we consider in this work.

FedAVG DYN-OPT attack against FedAVG is quite straightforward and uses a random direction ω and a very large value γ to compute the poisoned update $\theta^{t,m}_{c\in[C]}$.

Mutli-Krum Multi-Krum uses Krum iteratively to construct a selection set S and computes the average of the updates in the selection set as its aggregate. Therefore, DYN-OPT aims to maximize the perturbation $\gamma\omega$ used to compute the poison update $\theta^{t,m}_{c\in[C]}$, while ensuring that Multi-Krum selects all its poison updates in S. Note that this strategy minimizes the number of benign updates in S and maximizes $\gamma\omega$ by increasing the poisoning impact of malicious updates

on the final aggregate. The optimization problem we solve to mount DYN-OPT on Multi-Krum is given in (2).

$$\begin{aligned} \underset{\gamma}{\text{argmax}} \quad & |\{\theta^{t,m}_{c\in[C]} \in f_{\text{mkrum}}\left(\theta^{t,m}_{c\in[C]} \cup \theta^{t}_{i\in[C+1,n]}\right)\}| \\ \text{s.t.} \quad & \theta^{t,m}_{c\in[C]} = \theta^{b,t} + \gamma\omega \end{aligned} \tag{2}$$

Trimmed-Mean and Median For Trimmed-Mean and Median AGRs, DYN-OPT solves the optimization given in (3). Following [19], we fix the perturbation ω and keep all poisoned updates the same. The objective here is to maximize the L_2 norm of the distance between the benign update reference $\theta^{b,t}$ and the aggregate, $f_{\text{agr}}(.)$, calculated using $f_{\text{agr}} \in \{f_{\text{trmean}}, f_{\text{median}}\}$ on the set of benign and malicious updates.

$$\begin{aligned} \underset{\gamma}{\text{argmax}} \quad & \|\theta^{b,t} - f_{\text{agr}}\left(\theta^{t,m}_{c\in[C]} \cup \theta^{t}_{i\in[C+1,n]}\right)\|_2 \\ \text{s.t.} \quad & \theta^{t,m}_{c\in[C]} = \theta^{b,t} + \gamma\omega \end{aligned} \tag{3}$$

Norm-Bounding We formulate the DYN-OPT attack against AGR bound to Norm using the original framework proposed in [19]. More specifically, to circumvent Norm-Bounding, the norm of the poisoned update should be less than the threshold norm, τ, used by Norm-Bounding AGR. Therefore, to compute the poison update $\theta^{t,m}_{c\in[C]}$ using DYN-OPT, we can scale the norm of the original poison update, $\theta^{b,t} + \gamma\omega$, to τ. The final poisoned update would be $\theta^{t,m}_{c\in[C]} = \text{Scale}(\theta^{b,t} + \gamma\omega, \tau)$, where $\text{Scale}(u, \tau) = u \cdot \min(1, \frac{\tau}{\|u\|_2})$.

4 Our proposed hybrid adversary model

Compromising real clients in FL to launch a model poisoning attack can be a challenging task for an attacker. This is because genuine clients participating in FL are typically owned and controlled by different entities (e.g., individual users in cross-device FL and hospitals in cross-silo FL), and the attacker should get access to and take control of these clients in order to manipulate the updates they send to the server.

One way an attacker might try to do this is by using malware or phishing attacks to compromise clients. However,

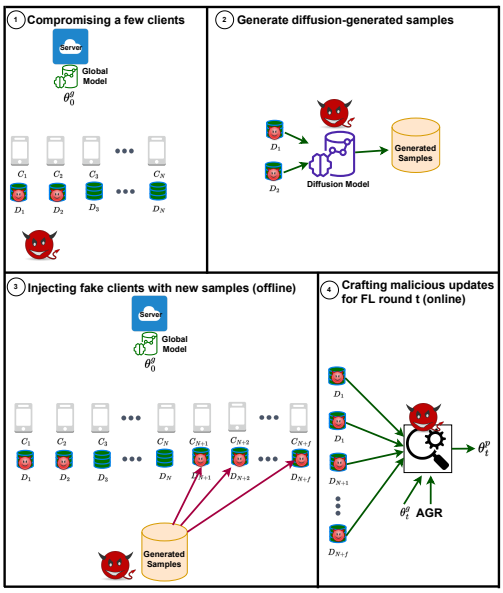


Fig. 1. Our hybrid attack pipeline.

successfully executing these types of attacks requires a certain level of skill and resources, and the attacker would need to be able to bypass any security measures that the clients have in place. Additionally, the cost of compromising a large number of genuine clients can be high, as the attacker would need to pay for access to undetected zombie devices or other resources. This may make it infeasible for the attacker to compromise a large fraction of genuine clients, which is typically necessary for a model poisoning attack to be successful.

Another factor that makes it difficult to compromise real clients in FL is the decentralized nature of the system. In FL, clients are typically distributed across a wide geographical area and may have different levels of security and defenses in place. This can make it difficult for the attacker to gain access to and take control of a large number of clients simultaneously.

In general, the combination of technical challenges and the high cost of compromising genuine clients in FL makes it a difficult task for an attacker to launch a successful model poisoning attack using only compromised clients.

Instead, we propose to use both fake and compromised clients to mount a hybrid attack. Figure 1 shows the pipeline of our hybrid attack: The hybrid adversary first compromises a few real clients and then uses their data to generate synthetic data using a DDPM. Next, the adversary uses these synthetic data to emulate FL clients and uses the model poisoning attacks (Section 3.2) to craft strong malicious updates. The injected fake clients and compromised clients submit the generated malicious update if the server selects them in that FL round for their local updates.

Note that in Figure 1, Step 1 can be removed if the adversary is able to obtain (high-quality) data samples that represent the data distribution of typical clients. For example, if abundant public data is available related to the target FL task, the adversary can simply use such public data to synthesize the poisoning data for its fake clients. However, high-quality (i.e., representative) public data is not always available, especially in proprietary applications.

4.1 Comparing the costs of different adversaries

In this section, we discuss the cost of the three types of attacks discussed above: fake, hybrid, and compromised. We assume that the cost of compromising a client is c and the cost of creating a fake client is f; depending on the scenario, c and f can vary widely, but generally the cost of a fake client is much lower than that of a compromised client, that is, $f \ll c$. Furthermore, we assume α_f fake clients in the fake attack, β_c compromised clients in the compromised attack, and α_h fake and β_h compromised clients in the hybrid attack.

If the number of malicious clients in the three attacks is the same, that is, $\alpha_f = \alpha_h + \beta_c = \beta_c$, the cost of each of the attacks is as follows: $f \cdot \alpha_f$ for the fake attack, $f \cdot \alpha_h + c \cdot \beta_h$ for the hybrid attack, and $c \cdot \beta_c$ for the compromised attack. Next, note that in our hybrid attack, we use very few compromised clients to launch a very large number of fake clients, i.e., $\alpha_h \gg \beta_h$, which also implies that the number of fake clients in our hybrid attack is very close to that in the

fake attack, i.e., $\alpha_h \approx \alpha_f$. Hence, the order of the cost of the three attacks is: $\text{cost}_f < \text{cost}_h \ll \text{cost}_c$, with cost_f and cost_h being very close.

Let us consider a concrete scenario involving IoT devices, e.g., CCTV traffic cameras or WiFi routers. The goal of the adversary is to mount a model poisoning attack against an IoT application, e.g., predicting traffic at a certain location. The application stores and uses images from traffic cameras, and trains a global image classification model using FL. With high probability, these IoT devices are also part of some botnet, and the cost of owning such zombie devices in a botnet can be as low as 1. However, all IoT devices need not have the target application, e.g., many CCTV cameras may not have required software/hardware updates. For concreteness, consider that 1% of the devices have the target application. Furthermore, note that, generally, the botnet owners do not know what all applications are running on the zombie devices.

Therefore, in case the compromised attack requires m malicious clients, where the zombie IoT devices must have the application, the adversary will have to buy $100m$ devices to ensure that m of them have the target application and discard $99m$ devices. However, in the case of our hybrid attack, the adversary just needs to ensure that $m' \ll m$ devices have the application (and, therefore, the required data) and should buy $\max(100m', m)$ devices. Then they can install the target application on the $m - m'$ devices and populate them with synthetic data. In the case of a fake attack, the adversary simply has to buy m devices. If the cost of buying a zombie device is c, the costs of compromised, hybrid and fake attacks are $100mc \gg 100m'c > mc$; the first inequality holds because $m \gg m'$.

5 Experimental Setup

5.1 Datasets and hyperparameters

In this work, we conduct experiments on two datasets, CIFAR10 [13] and FEMNIST [5, 7], in order to evaluate the performance of different Byzantine robust aggregation under different adversary models.

CIFAR10 dataset is a widely used image classification dataset consisting of 60,000 32x32 color images in 10 classes, with 6,000 images per class. There are 50,000 training images and 10,000 test images. For this dataset, we use VGG9 architecture. For local training in each FL round, each client uses 5 epochs. Each client uses SGD with learning rate of 0.01, momentum of 0.9, weight decay of 1e-4, and batch size of 8.

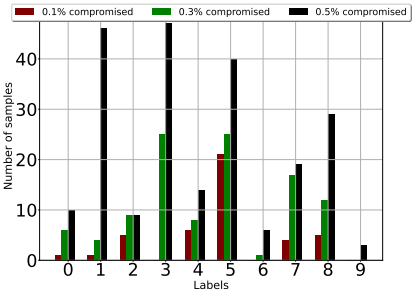


Fig. 2. Number of samples for each label when the attacker compromised 0.1% (1 client), 0.3% (3 clients), and 0.5% (5 clients) in our data distribution (fixed through all the experiments) for learning CIFAR10 distributed over 1000 clients.

FEMNIST is a character recognition classification task with 3,400 clients, 62 classes (52 for upper and lower case letters and 10 for digits), and 671,585 gray-scale images. Each

client has data of their own handwritten digits or letters. For this dataset, we use LeNet architecture. For local training in each FL round, each client uses 2 epochs. Each client uses SGD with learning rate of 0.01, momentum of 0.9, weight decay of 1e-4, and batch size of 10.

Data distribution: Most real-world FL settings have heterogeneous client data, hence following previous works [11, 18], we distribute CIFAR10 datasets among 1,000 clients in non-iid fashion using *Dirichlet* distribution with parameter $\beta = 0.5$. Note that, increasing β results in more iid datasets. FEMNIST is naturally distributed non-iid among 3,400 clients.

5.2 Evaluation metric

We run all the experiments for 2000 global rounds of FL for CIFAR10, and 1000 global rounds for FEMNIST, while selecting 25 clients in each round randomly. At the end of each FL round, we calculate the test accuracy of the global model on the test data, and update the maximum test accuracy. We run each experiment with 5 different random seeds, and we report the median and standard deviation of the maximum test accuracies in our experiments.

Attack impact metric (I_θ): We define attack impact, $I_\theta = A_\theta - A_\theta^M$, as the reduction of the accuracy of the global model when the attack is launched. (A_θ) denotes the maximum accuracy that the global model achieves overall FL training rounds without the presence of any malicious clients. A_θ^M for an attack shows the maximum accuracy of the model under a given attack. In our Tables, we report both the maximum test accuracies and Attack Impacts.

Attack Cost: Analyzing the cost efficiency tradeoffs of different poisoning attacks in federated learning is crucial for understanding the severity and impact of such attacks. In Section 4.1, we present a cost analysis of various poisoning attacks across the spectrum of adversary models, ranging from fake to compromising attacks. We assume that an adversary can acquire control of zombie devices in a botnet for $1 per device. Furthermore, we consider that only 1% of these devices possess the target application with real data. This implies that out of 100 purchased zombie devices, 99 do not have any real data (suitable for fake attacks), while one device has access to real data, which can be used for a compromising attack.

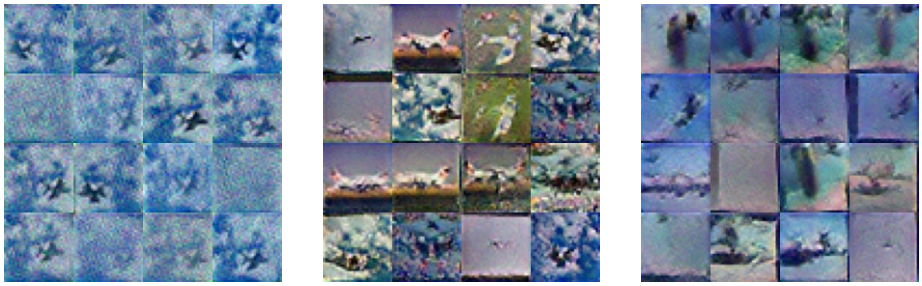
In a compromising attack scenario that necessitates m malicious clients, with the requirement that the zombie IoT devices have the target application, the attacker must acquire $100m$ devices to confirm that m devices possess the target application while discarding the other $99m$ devices. On the other hand, in our proposed hybrid attack, the attacker only needs to make certain that $m' \ll m$ devices contain the application (along with the required data) and purchase $\max(100m', m)$ devices. They can then install the target application on $m - m'$ devices and populate them with artificial data. In the case of a fake attack, the attacker simply needs to obtain m devices.

For instance, if the attacker aims to launch a compromising attack with 100 malicious clients, they would need to purchase 10,000 zombie devices. Assuming a cost of $1 per control of each device, the total cost amounts to $10,000. In

contrast, if the attacker desires 100 fake clients, the cost would be $100. If the attacker wants a hybrid attack with 3 compromised clients possessing real data and 97 fake clients, the cost would be $300. However, if the attack requires 1 real client and 99 fake clients for a compromising attack, the cost would be $100. We provide the cost of each attack scenario in each table based on the required number of malicious clients and the type of attack.

5.3 Generating synthetic data using DDPM

In Section 4, we explained the pipeline of our hybrid attack, which takes control of a few real clients and generates new synthetic data. In this section, we explain the details of this process for images of CIFAR10 and FEMNIST. To generate new samples, we use the following steps (similar to steps provided in Figure 1):



(a) Generated by DDPM using 0.1% (1 client) compromised

(b) Generated by DDPM using 0.3% (3 clients) compromised

(c) Generated by DDPM using 0.5% (5 clients) compromised

Fig. 3. Airplanes generated by DDPM using different percentages of compromised client's data in our hybrid attack.

Collecting the data of compromised clients. We collect all the data samples of 0.1%, 0.3% and 0.5% of first clients in both CIFAR10 and FEMNIST learning. For CIFAR10, we distribute the data in a non-iid fashion using Dirichlet distribution with parameter $\beta = 0.5$. We saved the data assignment of the dataset and used this fixed distribution throughout our experiments. For CIFAR10, we collect the data samples of the first 1 (0.1%), 3 (0.3%), and 5 (0.5%) of the clients. Figure 2 shows the number of samples for each label (label 0 represents airplane images, label 1 represents car images, etc.) for our data collection. As we can see from this figure, when the attacker has only compromised 0.1% of clients, it does not have access to any data samples of labels 3, 6, and 9. This means it cannot produce any new samples for these labels. For compromising 0.3%, the adversary does not have access to any samples from label 9. For FEMNIST, we also used the same generated data assignment (produce non-idd), and we collected the data samples of the first 4 (0.1%), 7 (0.3%), and 11 (0.5%) of the clients.

Generating new samples using DDPM We use the code provided in [2] to generate new samples for the hybrid attacks. This code implemented the denoising diffusion probabilistic model (DDPM) [10] in PyTorch. It is a transcribed code from the official Tensorflow version [1]. It uses denoising score matching to estimate the gradient of the data distribution, followed by Langevin sampling to sample from the true distribution. After collecting the data samples of compromised clients, we ran the DDPM on these images for each label separately to generate new samples. To train the diffusion model, we used a batch size of 8, learning rate of 0.00008, and 250 sampling size. To generate samples for CIFAR10, we used 2000 diffusion steps, and for FEMNIST we used 1000 diffusion steps.

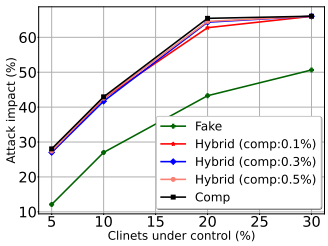
Figure 3 shows some DDPM-generated samples when the adversary has compromised 1 (0.1%), 3 (0.3%), and 5 (0.5%) of the clients in learning of CIFAR10 distributed over 1000 clients. Figure 2 shows the number of samples for each label. From this Figure, we can see the adversary has access to 1, 6, and 10 images of airplanes by compromising 0.1%, 0.3%, and 0.5% of the clients, respectively. In Figure 3(a), we can see that the DDPM model memorized the only image it has, and it just tried to add randomness to it because it has access to only one image of an airplane. Moreover, in Figure 3(b) and Figure 3(c), we can see that the model can generate better samples as it has access to more images from the true distribution.

Data assignment for the injected fake clients. In all the hybrid attacks experiments, we first create a large dataset of all synthetic images from all the labels. We create this dataset by generating 5 samples per label multiplied by the number of injected fake clients. Then we distributed this dataset over the fake clients in a non-iid fashion using Dirichlet distribution with parameter $\beta = 0.5$ for both CIFAR10 and FEMNIST experiments. Next, for launching the model poisoning attacks provided in Section 3.2, the adversary chooses 25 random fake clients for its optimization and creates its malicious updates. This process happens in each FL round based on the global parameters θ^t.

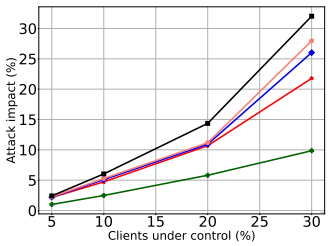
6 Experiments

In this section, we conduct experiments to evaluate the performance of different Byzantine robust aggregation rules under different adversaries, using the FEMNIST [5, 7] and CIFAR10 [13] datasets. We consider a range of malicious client percentages, including 5%, 10%, 20%, and 30%, and report the maximum test accuracy and the impact of various attacks on the global model. For each attack, we also report attack cost, the number of benign, compromised, and injected fake clients present in the FL training process.

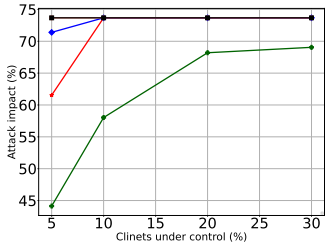
We consider five different attack scenarios, ranging from injecting fake clients with no knowledge of the true data distribution to a scenario where the adversary can compromise benign clients and use their data to craft malicious updates. Additionally, we propose and evaluate three types of hybrid attacks, where the adversary first compromises a small number of real clients and then uses their



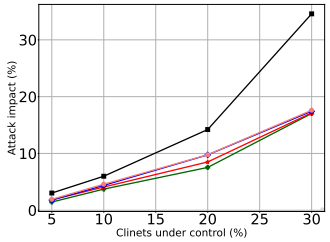
(a) CIFAR10 + Median (No attack acc=76.05%)



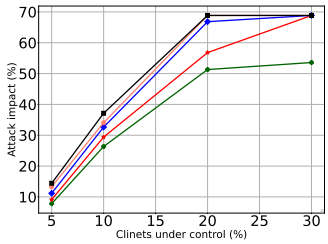
(b) FEMNIST + Median (No attack acc=84.29%)



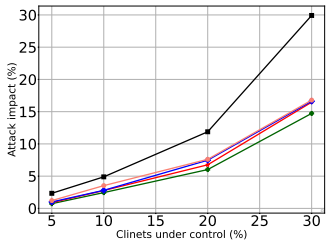
(c) CIFAR10 + NB ($\tau = 2.0$) (No attack acc=83.68%)



(d) FEMNIST + NB $\tau = 2.0$ (No attack acc=87.49%)



(e) CIFAR10 + NB $\tau = 0.5$ (No attack acc=78.86%)



(f) FEMNIST + NB $\tau = 0.5$ (No attack acc=86.35%)

Fig. 4. Attack impact (I_θ) of the Norm-Bounding and Median aggregation rules in the presence of different adversaries. τ shows the ℓ_2 threshold value that is used in Norm-Bounding AGR.

data to generate synthetic samples using a DDPM, followed by injecting fake clients with the new data samples. We explore the impact of different numbers of compromised clients in these hybrid attacks, specifically 0.5% (5 clients), 0.3% (3 clients), and 0.1% (1 client) in CIFAR10 experiments and 0.5% (17 clients), 0.3% (11 clients), and 0.1% (4 clients) in FEMNIST experiments. We rank the attacks in terms of their impact on the global model accuracy, to better illustrate the spectrum of attacks.

It is worth noting that we omit the results of the standard aggregation rule, FedAvg, as it is known to be vulnerable to even a single malicious client [4] and can result in the global test accuracy approaching random guessing.

6.1 Attacking agnostic robust AGRs

Median AGR. We present our experimental results using the Median aggregation rule in Figure 4 (a) and (b) for CIFAR10 and FEMNIST experiments, respectively. Detailed results, including the attack cost, the number of benign, compromised, and injected fake clients corresponding to each attack, are provided in Table 3 and Table 4 (in Appendix A).

Our findings indicate that the most potent adversary, who has compromised real clients, exerts the most significant influence on the global model. For instance, on the CIFAR10 dataset with the Median as the AGR, an attack by 10% (20%) malicious clients reduces the model's accuracy to 33.10% (10.61%). This implies that the attacker first compromised 100 (200) clients out of the total clients participating in FL and launched the attack described in Section 3.2 to craft its malicious update. The costs of these attacks would be $10,000 and $20,000, respectively, making them quite expensive.

On the other hand, fake clients, who do not have any knowledge about the benign clients' data distribution, have the least impact on the global model. For example, on CIFAR10 with Median as the AGR, an attack launched by 10% (20%) of malicious clients reduces the accuracy of the global model to 49.04% (32.78%). To accomplish this, the adversary must inject 112 (251) fake clients into the FL training, which incurs costs of $112 and $251, respectively, considerably cheaper than compromising attacks.

Hybrid attacks, positioned in the middle of the spectrum, reveal that if the hybrid adversary has access to more data (more compromised clients), they can inflict more significant damage on the global model's accuracy. For instance, in the CIFAR10 dataset with the Median as the AGR, a hybrid attack involving 20% malicious clients, where the adversary has compromised 1, 3, and 5 clients while generating new instances and injecting 249, 247, and 244 new fake clients, can reduce the FL model's accuracy to 13.29%, 11.71%, and 11.49%, respectively. These attacks cost $250, $300, and $500, respectively, which is very close to the cost of the fake attacks. Similar observations are made for the FEMNIST dataset as well.

Norm-Bounding AGR. We report the experimental results of our experiments when the server applies Norm-Bounding with a threshold τ as the aggregation rule in Figure 4 (b), (c), (e), and (f) for CIFAR10 and FEMNIST datasets with two thresholds $\tau = 0.5$ and $\tau = 2.0$. Our results show that the Norm-Bounding aggregation rule has similar impacts on the global model's accuracy as the Median AGR, when faced with different types of attacks. For example, on CIFAR10 with $\tau = 0.5$, when the adversary controls 10% of clients, the fake adversary can inject 112 fake clients (with a cost of $112) and reduce the accuracy to 52.52%; the hybrid attack who compromised 1 client and injected 110 clients (with a cost of $111) reduces the accuracy to 49.46%; the hybrid

attacker who compromised 3 clients and injected 108 fake clients (with a cost of $300) reduces the accuracy to 46.22%; the hybrid attacker who compromised 5 clients and injected 106 clients (with a cost of $500) reduces the accuracy to 44.79%; and at the end of the spectrum, a powerful adversary who compromised 100 clients (with a cost of $10,000) can reduce the accuracy to 41.73%.

Larger upper bounds in Norm-Bounding results in more damage to the global model. In our experiments, we consider two thresholds for Norm-Bounding $\tau = 0.5$ and $\tau = 2.0$. Our results show that for a larger threshold bound (τ), the adversary has a larger space to craft its malicious updates and have a more significant impact on the FL global model. For instance, on FEMNIST, the compromising adversary with 30% malicious ratio causes the accuracy dropped by 34.58% when $\tau = 2.0$ while the accuracy drop for the same setting and $\tau = 0.5$ is about 29.92%.

Therefore, with larger norm thresholds for the Norm-Bounding aggregation rule, the attackers have more impact on the global model. Alternatively, If the server wants to use a smaller threshold, then the model will result in lower accuracy when there is no malicious client. For instance, on CIFAR10, with no malicious clients, Norm-Bounding with threshold $\tau = 0.5$ results in 78.86% while $\tau = 2.0$ results in 83.68%; 10% compromised clients will result in losing of 37.13% and 73.68% for $\tau = 0.5$ and $\tau = 2.0$ respectively. Therefore, there is a trade-off for choosing a proper threshold for bounding the local updates based on the assumption of the number of malicious clients in FL training.

Why can fake clients cause a significant attack impact for Norm-Bounding AGR? Figure 5 shows the L2 norm of the updates (for malicious and benign updates for 10% of malicious ratio in fake attack) before and after bounding the updates to $\tau = 0.5$ for learning CIFAR10 throughout 2000 FL rounds. From this figure, we can see that when the global model starts to converge, the L2 norm of the local updates from benign updates becomes smaller than the threshold. For the updates that have norms smaller than the threshold, no change will be applied to them. However, on the other hand, the malicious updates are always greater than the threshold, so they are scaled down to have an L2 norm of τ. In this figure, we can see that after FL round 1500, the malicious updates have a more

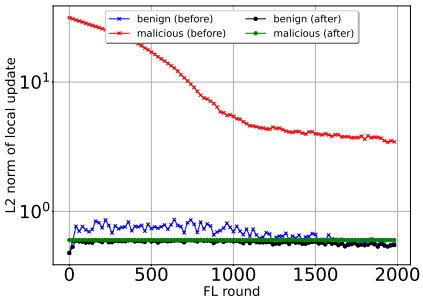


Fig. 5. Local update norms throughout the FL training on CIFAR10 with 1000 benign clients and 112 fake clients (i.e., the adversary controls 10% of total clients). In this figure, we can see that after FL round 1500, the malicious updates have a more considerable impact on the aggregation compared to benign updates because they have larger updates after norm bounding.

considerable impact on the aggregation because they have larger updates.

6.2 Attacking adaptive robust AGRs

In this section, we conduct experiments to evaluate the robustness of adaptive Byzantine aggregation rules, specifically Trimmed-Mean [22] and Multi-Krum [4], against a spectrum of adversaries who control varying percentages of malicious clients. In adaptive aggregation rules, we assume that the server has knowledge of the exact number of malicious clients in each FL round.

Table 1. Attack impact (I_θ) and maximum test accuracy (A_θ^M) of the Trimmed-Mean for training on CIFAR10 distributed over 1000 initial clients in the presence of different adversaries.

AGR	Attack Type	Malicious Rate	Number of Benign Clients	Number of Compromised Clients	Number of Injected Fake Clients	Attack Cost ($)	Accuracy (%)	Attack Impact (%)
Trimmed-Mean (No attack acc = 83.66%)	Fake	5%	1000	0	53	53	59.95 (± 0.617)	23.71 (± 0.617)
		10%	1000	0	112	112	43.88 (± 0.334)	39.78 (± 0.334)
		20%	1000	0	251	251	32.49 (± 0.451)	51.17 (± 0.451)
		30%	1000	0	429	429	25.56 (± 0.238)	58.10 (± 0.238)
	Hybrid comp: 0.1%	5%	999	1	52	100	50.19 (± 2.791)	33.47 (± 2.791)
		10%	999	1	110	111	29.42 (± 1.481)	54.24 (± 1.481)
		20%	999	1	249	250	20.61 (± 5.277)	63.05 (± 5.277)
		30%	999	1	428	429	10.00 (± 1.188)	73.66 (± 1.188)
	Hybrid comp: 0.3%	5%	997	3	50	300	47.78 (± 0.928)	35.88 (± 0.928)
		10%	997	3	108	300	28.56 (± 1.071)	55.10 (± 1.071)
		20%	997	3	247	300	20.50 (± 5.415)	63.16 (± 5.415)
		30%	997	3	425	428	10.01 (± 0.209)	73.65 (± 0.209)
	Hybrid comp: 0.5%	5%	995	5	48	500	41.90 (± 3.438)	41.76 (± 3.438)
		10%	995	5	106	500	27.89 (± 0.909)	55.77 (± 0.909)
		20%	995	5	244	500	20.31 (± 5.151)	63.35 (± 5.151)
		30%	995	5	422	500	10.00 (± 0.180)	73.66 (± 0.180)
	Comp	5%	950	50	0	5,000	44.25 (± 1.195)	39.41 (± 1.195)
		10%	900	100	0	10,000	27.33 (± 0.346)	55.83 (± 0.346)
		20%	800	200	0	20,000	10.00 (± 4.130)	73.66 (± 4.130)
		30%	700	300	0	30,000	10.00 (± 0.000))	73.66 (± 0.000)

We report the performance of the Trimmed-Mean aggregation rule against different attacks in Table 1 and Table 5 (in Appendix A) for FL models trained on the CIFAR10 and FEMNIST datasets, respectively, in the presence of 5%, 10%, 20%, and 30% of malicious clients. Similarly, Table 2 and Table 6 (in Appendix A) show the attack impacts of different attacks when the server uses Multi-Krum as the aggregation rule for the CIFAR10 and FEMNIST datasets, respectively.

Our results indicate that adversaries who can compromise clients and use their data for attacks have the most significant impact on FL global models. For instance, on the CIFAR10 dataset, an adversary who has compromised 10% (20%) of clients, with a cost of $10,000 ($20,000), reduces the accuracy of FL by 55.83% (73.66%) and 49.29% (60.37%) with Trimmed-Mean and Multi-Krum, respectively. On the other hand, adversaries who can only inject fake clients into the FL training with no knowledge of the true data distribution have the lowest impact on global model accuracy. For instance, on the CIFAR10 dataset, an adversary who can inject 10% (20%) of clients, with a cost of $112 ($251), reduces

Table 2. Attack impact (I_θ) and maximum test accuracy (A_θ^M) of the Multi-Krum for training on CIFAR10 distributed over 1000 initial clients in the presence of different adversaries.

AGR	Attack Type	Malicious Rate	Number of Benign Clients	Number of Compromised Clients	Number of Injected Fake Clients	Attack Cost ($)	Accuracy (%)	Attack Impact (%)
Multi-Krum (No attack acc = 83.44%)	Fake	5%	1000	0	53	53	82.70 (± 0.291)	0.74 (± 0.291) 6
		10%	1000	0	112	112	82.12 (± 0.227)	1.32 (± 0.227) 5
		20%	1000	0	251	251	79.89 (± 0.226)	3.55 (± 0.226) 5
		30%	1000	0	429	429	75.29 (± 0.256)	8.15 (± 0.256) 6
	Hybrid comp: 0.1%	5%	999	1	52	100	70.12 (± 0.895)	13.32 (± 0.895) 4
		10%	999	1	110	111	48.24 (± 2.371)	35.20 (± 2.371) 4
		20%	999	1	249	250	24.71 (± 0.257)	58.73 (± 0.257) 4
		30%	999	1	428	429	20.22 (± 0.539)	63.22 (± 0.539) 4
	Hybrid comp: 0.3%	5%	997	3	50	300	62.65 (± 0.725)	20.79 (± 0.725) 3
		10%	997	3	108	300	36.70 (± 2.188)	46.74 (± 2.188) 3
		20%	997	3	247	300	23.79 (± 1.788)	59.65 (± 1.788) 3
		30%	997	3	425	428	19.90 (± 2.234)	63.54 (± 2.234) 3
	Hybrid comp: 0.5%	5%	995	5	48	500	62.47 (± 0.914)	20.97 (± 0.914) 2
		10%	995	5	106	500	35.65 (± 0.956)	47.79 (± 0.956) 2
		20%	995	5	244	500	23.10 (± 1.433)	60.34 (± 1.433) 2
		30%	995	5	422	500	19.86 (± 0.619)	63.58 (± 0.619) 2
	Comp	5%	950	50	0	5,000	62.04 (± 1.307)	21.40 (± 1.307) 1
		10%	900	100	0	10,000	34.15 (± 0.660)	49.29 (± 0.660) 1
		20%	800	200	0	20,000	23.07 (± 0.528)	60.37 (± 0.528) 1
		30%	700	300	0	30,000	19.31 (± 0.786)	64.13 (± 0.786) 1

the accuracy of FL by 39.78% (51.17%) and 1.32% (3.55%) with Trimmed-Mean and Multi-Krum, respectively.

Our experiments also show that the hybrid attack, which compromises only a few clients and use their data to produce more data samples for the fake clients, lies in the middle of the spectrum. The more clients are compromised, the more damage is done to the global accuracy. For instance, on the CIFARA10 training, a hybrid attacker who compromised 1 client, i.e., 0.1% of total clients, and can inject 110 clients (in total 10% malicious ratio) with a cost of $111, can reduce the accuracy of the FL model by 54.24% and 35.2% for Trimmed-Mean and Multi-Krum respectively. While if the hybrid attacker compromised more clients (5 clients) and injected 106 clients (in total 10% malicious ratio), with a cost of $500, it can reduce the FL global accuracy by 55.77% and 47.79% for Trimmed-Mean and Multi-Krum, respectively.

Additionally, we also noticed that the Trimmed-Mean and Norm-Bounding (with $\tau = 0.5$) are more vulnerable to injected fake clients with no knowledge about the true distribution of the training datasets. On the other hand, Multi-Krum can easily detect them and exclude them from aggregation. For instance, on CIFAR10, 10% of injected fake clients (with $112 attack cost) can reduce the accuracy of the model by 26.34% and 39.78% with Norm-Bounding and Trimmed-Mean as the aggregation rule, respectively. On the other hand, Multi-Krum only loses 1.32% with the presence of this number of injected fake clients.

7 Conclusions

In conclusion, this work presents a comprehensive study of the poisoning threats to FL by considering a spectrum of adversaries and robust AGRs. We identify a hybrid adversary model where an adversary first compromises a few real clients and use their data to generate more data samples for the fake clients to mount a large-scale attack. For such a hybrid adversary, we propose a novel hybrid attack that leverages the denoising diffusion probabilistic model (DDPM) to generate new samples from a small number of compromised clients. Our experimental results, conducted using FEMNIST and CIFAR10 datasets, demonstrate the varying impact of different attack configurations on FL systems. Notably, we find that the hybrid attacks, utilizing a mix of compromised and synthetically generated fake clients, offer a potent threat vector that balances cost and impact effectively. These findings highlight significant vulnerabilities in current FL systems, particularly under adaptive aggregation rules, and underscore the need for developing more sophisticated defense mechanisms that can anticipate and mitigate a range of attack modalities.

Acknowledgements

The project was supported in part by the NSF grant 2131910 and a research gift from Adobe Research.

References

1. Denoising Diffusion Probabilistic Model, in Tensorflow. https://github.com/hojonathanho/diffusion (2020)
2. Denoising Diffusion Probabilistic Model, in Pytorch. https://github.com/lucidrains/denoising-diffusion-pytorch (2022)
3. Baruch, M., Gilad, B., Goldberg, Y.: A little is enough: Circumventing defenses for distributed learning. Advances in Neural Information Processing Systems (2019)
4. Blanchard, P., Guerraoui, R., Stainer, J., et al.: Machine learning with adversaries: Byzantine tolerant gradient descent. In: Advances in Neural Information Processing Systems. pp. 119–129 (2017)
5. Caldas, S., Wu, P., Li, T., Konečný, J., McMahan, H.B., Smith, V., Talwalkar, A.: LEAF: A benchmark for federated settings. CoRR **abs/1812.01097** (2018), http://arxiv.org/abs/1812.01097
6. Cao, X., Gong, N.Z.: Mpaf: Model poisoning attacks to federated learning based on fake clients. arXiv preprint arXiv:2203.08669 (2022)
7. Cohen, G., Afshar, S., Tapson, J., van Schaik, A.: EMNIST: extending MNIST to handwritten letters. In: 2017 International Joint Conference on Neural Networks, IJCNN (2017)
8. Fang, M., Cao, X., Jia, J., Gong, N.Z.: Local model poisoning attacks to byzantine-robust federated learning. In: Capkun, S., Roesner, F. (eds.) 29th USENIX Security Symposium, USENIX Security (2020)

9. Fraboni, Y., Vidal, R., Lorenzi, M.: Free-rider attacks on model aggregation in federated learning. In: International Conference on Artificial Intelligence and Statistics. pp. 1846–1854. PMLR (2021)
10. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 6840–6851. Curran Associates, Inc. (2020), https://proceedings.neurips.cc/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf
11. Hsu, T.M.H., Qi, H., Brown, M.: Measuring the effects of non-identical data distribution for federated visual classification. arXiv preprint arXiv:1909.06335 (2019)
12. Konečnỳ, J., McMahan, H.B., Yu, F.X., Richtárik, P., Suresh, A.T., Bacon, D.: Federated learning: Strategies for improving communication efficiency. arXiv preprint arXiv:1610.05492 (2016)
13. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images (2009)
14. Lin, J., Du, M., Liu, J.: Free-riders in federated learning: Attacks and defenses. arXiv preprint arXiv:1911.12560 (2019)
15. McMahan, H.B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.y.: Communication-efficient learning of deep networks from decentralized data. Proceedings of the 20 th International Conference on Artificial Intelligence and Statistics (2017)
16. Mozaffari, H., Shejwalkar, V., Houmansadr, A.: Every vote counts: Ranking-based training of federated learning to resist poisoning attacks. In: USENIX Security Symposium (2023)
17. Nichol, A.Q., Dhariwal, P.: Improved denoising diffusion probabilistic models. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 139, pp. 8162–8171. PMLR (18–24 Jul 2021), https://proceedings.mlr.press/v139/nichol21a.html
18. Reddi, S.J., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečnỳ, J., Kumar, S., McMahan, H.B.: Adaptive federated optimization. In: ICLR (2020)
19. Shejwalkar, V., Houmansadr, A.: Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning. In: Proceedings of the 28th Network and Distributed System Security Symposium, (NDSS) (2021)
20. Shejwalkar, V., Houmansadr, A., Kairouz, P., Ramage, D.: Back to the drawing board: A critical evaluation of poisoning attacks on federated learning. arXiv preprint arXiv:2108.10241 (2021)
21. Sun, Z., Kairouz, P., Suresh, A.T., McMahan, H.B.: Can you really backdoor federated learning? In: NeurIPS FL Workshop (2019)
22. Yin, D., Chen, Y., Ramchandran, K., Bartlett, P.L.: Byzantine-robust distributed learning: Towards optimal statistical rates. In: ICML (2018)

A Auxiliary results of model poisoning attacks against aware AGRs

In this section, we present the results of our experiments for using different AGRs. For each attack, we report the number of benign, compromised, and injected fake clients present in the FL training process.

Table 3. Attack impact (I_θ) and maximum test accuracy (A_θ^M) of the Median for training on CIFAR10 distributed over 1000 initial clients in the presence of different adversaries.

AGR	Attack Type	Malicious Rate	Number of Benign Clients	Number of Compromised Clients	Number of Injected Fake Clients	Attack Cost ($)	Accuracy (%)	Attack Impact (%)
Median (No attack acc = 76.05%)	Fake	5%	1000	0	53	53	63.94 (± 1.253)	12.11 (± 1.253) (5)
		10%	1000	0	112	112	49.04 (± 0.649)	27.01 (± 0.649) (5)
		20%	1000	0	251	251	32.78 (± 0.699)	43.27 (± 0.699) (5)
		30%	1000	0	429	429	25.41 (± 4.937)	50.64 (± 4.937) (5)
	Hybrid comp: 0.1%	5%	999	1	52	100	49.08 (± 1.131)	26.97 (± 1.131) (4)
		10%	999	1	110	111	33.53 (± 0.902)	42.52 (± 0.902) (3)
		20%	999	1	249	250	13.29 (± 6.026)	62.76 (± 6.026) (4)
		30%	999	1	428	429	10.03 (± 0.536)	66.02 (± 0.536) (4)
	Hybrid comp: 0.3%	5%	997	3	50	300	48.85 (± 1.258)	27.20 (± 1.258) (3)
		10%	997	3	108	300	34.36 (± 0.892)	41.69 (± 0.892) (4)
		20%	997	3	247	300	11.71 (± 5.848)	64.34 (± 5.848) (3)
		30%	997	3	425	428	10.00 (± 0.000)	66.05 (± 0.000) (3)
	Hybrid comp: 0.5%	5%	995	5	48	500	48.65 (± 1.654)	27.40 (± 1.654) (2)
		10%	995	5	106	500	33.48 (± 1.337)	42.57 (± 1.337) (2)
		20%	995	5	244	500	11.49 (± 5.820)	64.56 (± 5.820) (2)
		30%	995	5	422	500	10.00 (± 0.000)	66.05 (± 0.000) (2)
	Comp	5%	950	50	0	5,000	48.01 (± 0.598)	28.04 (± 0.598) (1)
		10%	900	100	0	10,000	33.10 (± 1.166)	42.95 (± 1.166) (1)
		20%	800	200	0	20,000	10.61 (± 1.669)	65.44 (± 1.669) (1)
		30%	700	300	0	30,000	10.00 (± 0.000)	66.05 (± 0.000) (1)

Table 4. Attack impact (I_θ) and maximum test accuracy (A_θ^M) of the Median for training on FEMNIST distributed over 3400 initial clients in the presence of different adversaries.

AGR	Attack Type	Malicious Rate	Number of Benign Clients	Number of Compromised Clients	Number of Injected Fake Clients	Attack Cost ($)	Accuracy (%)	Attack Impact (%)
Median (No attack acc = 84.29%)	Fake	5%	3400	0	179	179	83.29 (± 0.146)	1.00 (± 0.146) (5)
		10%	3400	0	378	378	81.81 (± 0.109)	2.48 (± 0.109) (5)
		20%	3400	0	850	850	78.48 (± 0.223)	5.81 (± 0.223) (5)
		30%	3400	0	1458	1,458	74.44 (± 0.574)	9.85 (± 0.574) (5)
	Hybrid comp: 0.1%	5%	3396	4	175	400	82.13 (± 0.126)	2.16 (± 0.126) (4)
		10%	3396	4	374	400	79.57 (± 0.275)	4.72 (± 0.275) (4)
		20%	3396	4	845	849	73.61 (± 0.756)	10.68 (± 0.756) (4)
		30%	3396	4	1452	1,456	62.51 (± 4.007)	21.78 (± 4.007) (4)
	Hybrid comp: 0.3%	5%	3389	11	168	1,100	82.09 (± 0.335)	2.20 (± 0.335) (3)
		10%	3389	11	366	1,100	79.20 (± 0.194)	5.09 (± 0.194) (3)
		20%	3389	11	837	1,100	73.36 (± 0.989)	10.93 (± 0.989) (3)
		30%	3389	11	1442	1,453	58.27 (± 6.189)	26.02 (± 6.189) (3)
	Hybrid comp: 0.5%	5%	3383	17	162	1,700	82.04 (± 0.310)	2.25 (± 0.310) (2)
		10%	3383	17	359	1,700	79.02 (± 0.326)	5.27 (± 0.326) (2)
		20%	3383	17	829	1,700	73.12 (± 0.333)	11.17 (± 0.333) (2)
		30%	3383	17	1433	1,700	56.33 (± 3.858)	27.96 (± 3.858) (2)
	Comp	5%	3230	170	0	17,000	81.88 (± 0.247)	2.41 (± 0.247) (1)
		10%	3060	340	0	34,000	78.26 (± 0.214)	6.03 (± 0.214) (1)
		20%	2720	680	0	68,000	69.93 (± 0.481)	14.36 (± 0.481) (1)
		30%	2380	1020	0	102,000	52.27 (± 1.458)	32.02 (± 1.458) (1)

Table 5. Attack impact (I_θ) and maximum test accuracy (A_θ^M) of the Trimmed-Mean for training on FEMNIST distributed over 3400 initial clients in the presence of different adversaries.

AGR	Attack Type	Malicious Rate	Number of Benign Clients	Number of Compromised Clients	Number of Injected Fake Clients	Attack Cost ($)	Accuracy (%)	Attack Impact (%)
Trimmed-Mean (No attack acc = 87.52%)	Fake	5%	3400	0	179	179	84.90 (± 0.108)	2.62 (± 0.108) 5
		10%	3400	0	378	378	82.64 (± 0.135)	4.88 (± 0.135) 5
		20%	3400	0	850	850	78.04 (± 0.198)	9.48 (± 0.198) 5
		30%	3400	0	1458	1,458	73.11 (± 0.384)	14.41 (± 0.384) 5
	Hybrid comp: 0.1%	5%	3396	4	175	400	84.04 (± 0.223)	3.48 (± 0.223) 4
		10%	3396	4	374	400	80.44 (± 0.672)	7.08 (± 0.672) 4
		20%	3396	4	845	849	72.09 (± 1.114)	15.43 (± 1.114) 4
		30%	3396	4	1452	1,456	58.28 (± 0.699)	29.24 (± 0.699) 4
	Hybrid comp: 0.3%	5%	3389	11	168	1,100	83.95 (± 0.151)	3.57 (± 0.151) 3
		10%	3389	11	366	1,100	79.38 (± 0.313)	8.14 (± 0.313) 2
		20%	3389	11	837	1,100	70.48 (± 0.815)	17.07 (± 0.815) 3
		30%	3389	11	1442	1,453	57.15 (± 1.953)	30.37 (± 1.953) 3
	Hybrid comp: 0.5%	5%	3383	17	162	1,700	83.73 (± 0.248)	3.79 (± 0.248) 2
		10%	3383	17	359	1,700	79.75 (± 0.659)	7.77 (± 0.659) 3
		20%	3383	17	829	1,700	70.33 (± 2.009)	17.19 (± 2.009) 2
		30%	3383	17	1433	1,700	54.20 (± 2.420)	33.32 (± 2.420) 2
	Comp	5%	3230	170	0	17,000	83.51 (± 0.183)	4.01 (± 0.183) 1
		10%	3060	340	0	34,000	78.71 (± 0.498)	8.81 (± 0.498) 1
		20%	2720	680	0	68,000	68.13 (± 2.040)	19.39 (± 2.040) 1
		30%	2380	1020	0	102,000	40.35 (± 2.275)	47.17 (± 2.275) 1

Table 6. Attack impact (I_θ) and maximum test accuracy (A_θ^M) of the Multi-Krum for training on FEMNIST distributed over 3400 initial clients in the presence of different adversaries.

AGR	Attack Type	Malicious Rate	Number of Benign Clients	Number of Compromised Clients	Number of Injected Fake Clients	Attack Cost ($)	Accuracy (%)	Attack Impact (%)
Multi-Krum (No attack acc = 87.45%)	Fake	5%	3400	0	179	179	87.25 (± 0.064)	0.20 (± 0.064) 5
		10%	3400	0	378	378	87.11 (± 0.066)	0.34 (± 0.066) 5
		20%	3400	0	850	850	86.58 (± 0.178)	0.87 (± 0.178) 5
		30%	3400	0	1458	1,458	85.60 (± 0.174)	1.85 (± 0.174) 5
	Hybrid comp: 0.1%	5%	3396	4	175	400	86.02 (± 0.176)	1.43 (± 0.176) 3
		10%	3396	4	374	400	82.82 (± 0.352)	4.63 (± 0.352) 4
		20%	3396	4	845	849	75.88 (± 0.635)	11.57 (± 0.635) 4
		30%	3396	4	1452	1,456	62.23 (± 1.825)	25.22 (± 1.825) 3
	Hybrid comp: 0.3%	5%	3389	11	168	1,100	86.26 (± 0.106)	1.19 (± 0.106) 4
		10%	3389	11	366	1,100	81.58 (± 0.223)	5.87 (± 0.223) 4
		20%	3389	11	837	1,100	73.97 (± 0.582)	13.48 (± 0.582) 3
		30%	3389	11	1442	1,453	62.35 (± 0.859)	25.10 (± 0.859) 4
	Hybrid comp: 0.5%	5%	3383	17	162	1,700	85.87 (± 0.126)	1.98 (± 0.126) 2
		10%	3383	17	359	1,700	82.03 (± 0.376)	5.42 (± 0.376) 3
		20%	3383	17	829	1,700	71.71 (± 2.148)	15.74 (± 2.148) 2
		30%	3383	17	1433	1,700	61.94 (± 1.990)	25.51 (± 1.990) 2
	Comp	5%	3230	170	0	17,000	85.46 (± 0.113)	1.99 (± 0.113) 1
		10%	3060	340	0	34,000	81.73 (± 0.390)	5.72 (± 0.390) 2
		20%	2720	680	0	68,000	69.39 (± 1.597)	18.06 (± 1.597) 1
		30%	2380	1020	0	102,000	47.83 (± 10.627)	39.62 (± 10.627) 1