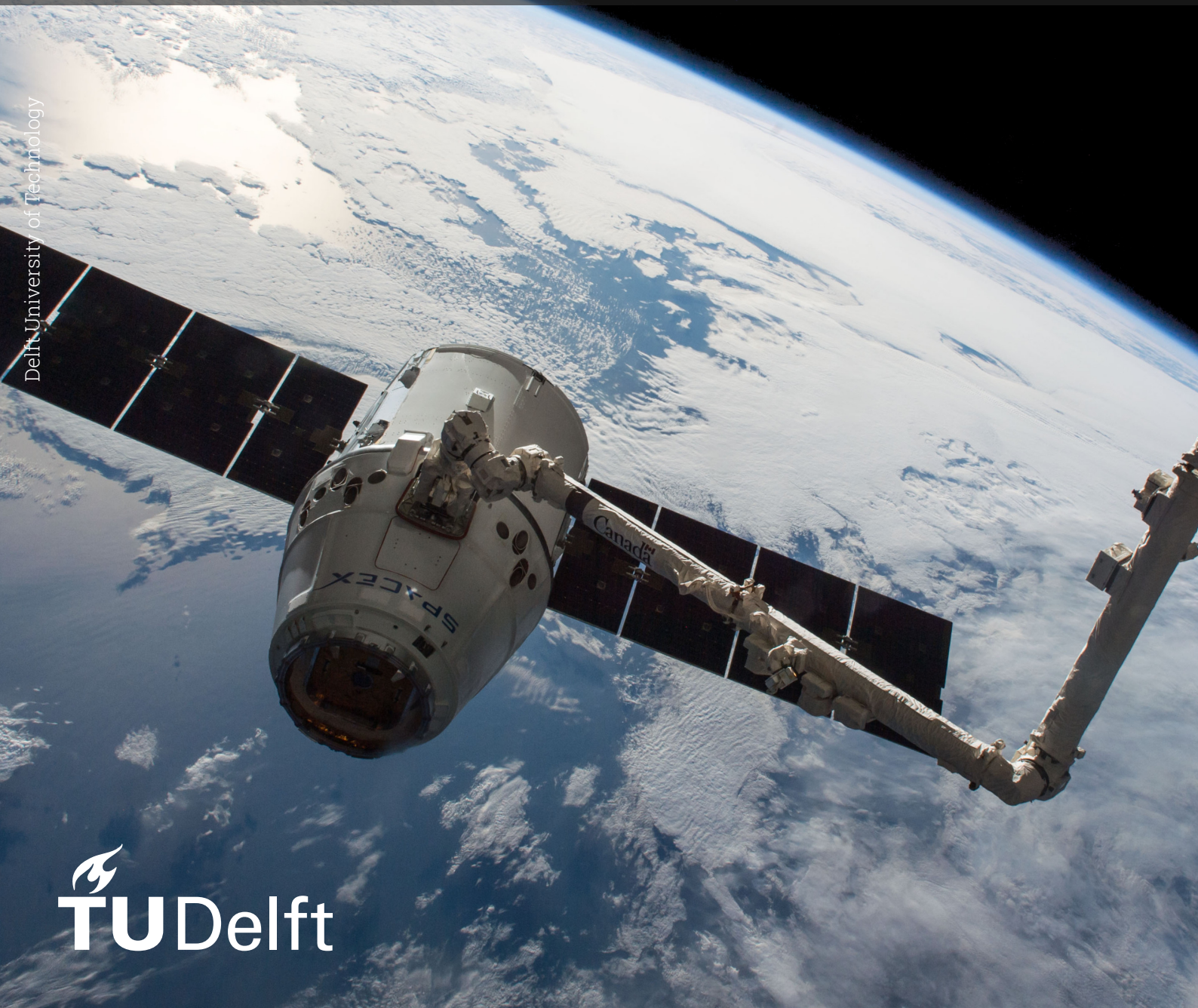
Robust backdoor attack against federated learning

Master Thesis

Congwen Chen



Delft University of Technology

TU Delft

Robust backdoor attack against federated learning

Master Thesis

by

Congwen Chen

to obtain the degree of Master of Science in Computer Engineering
Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS) at the Delft University of Technology

Student Name	Student number
Congwen Chen	5492815

Programme: Computer Engineering
Thesis Committee: Prof. Sicco Verwer
Prof. Kaitai Liang
Prof. Stephan Wong
Project Duration: September, 2021 - August, 2023

TU Delft

Preface

Federated Learning is a private-by-design collaborative learning framework where data holders upload locally trained models to the server for aggregation. Unlike the centralized learning setting, where a server collects substantial users' data to build a commonly used model, data never leaves from local devices under the federated learning framework. Therefore, Federated Learning claims that privacy is preserved.

We propose FTA, a stealthy and robust Backdoor attack with flexible trigger on federated learning (FTA), which effectively poisons the global model and circumvents current SOTA defense methods. Unlike using a fixed backdoor trigger throughout training, our approach utilizes a flexible trigger pattern that dynamically adjusts to global models in different training rounds and varies across samples. This enhances the effectiveness of the backdoor attack by malicious agents while making it harder to detect.

We conducted extensive evaluations on four publicly available datasets and tested our attack against eight defense methods. The results demonstrate three main advantages of our method. Firstly, compared to traditional backdoor attacks, FTA achieves faster and more persistent poisoning of the global model. Secondly, our attack exhibits superior stealthiness by successfully bypassing all tested SOTA methods. Thirdly, FTA ensures visual undetectability, as the size of backdoor triggers on images is minimal, allowing the poisoned images to easily evade human inspection.

In hindsight, the entire thesis process was enjoyable. I would like to thank everyone who gives me support through these 7 months. I would like to thank Yanqi and Prof. Kaitai Liang for guiding me into this exciting topic and on time help. I would also like to thank Prof. Sicco Verwer for giving me nice opinions in the mid-term presentations and Prof. Stephan Wong for your willingness to join the thesis committee and evaluate my work. In the end, I want to thank everyone for your help in the process of my thesis project.

Congwen Chen
Delft, May 2023

Contents

ABSTRACT

Current backdoor attacks against federated learning (FL) strongly rely on universal triggers or semantic patterns, which can be easily detected and filtered by certain defense mechanisms such as norm clipping, comparing parameter divergences among local updates. In this work, we propose a new stealthy and robust backdoor attack with flexible triggers against FL defenses. To achieve this, we build a generative trigger function that can learn to manipulate the benign samples with an imperceptible flexible trigger pattern and simultaneously make the trigger pattern include the most significant hidden features of the attacker-chosen label. Moreover, our trigger generator can keep learning and adapt across different rounds, allowing it to adjust to changes in the global model. By filling the distinguishable difference (the mapping between the trigger pattern and target label), we make our attack naturally stealthy. Extensive experiments on real-world datasets verify the effectiveness and stealthiness of our attack compared to prior attacks on decentralized learning framework with eight well-studied defenses.

1

INTRODUCTION

Federated learning (FL) has recently provided practical performance in various real-world applications and tasks, such as prediction of oxygen requirements of symptomatic patients with COVID-19 [7], autonomous driving [20], Gboard [37] and Siri [26]. It supports collaborative training of an accurate global model by allowing multiple agents to upload local updates, such as gradients or weights, to a server without compromising local datasets. This decentralized paradigm unfortunately exposes FL to a security threat — backdoor attacks [3, 36, 34, 40]. For example, in the FL context, the attacker can manipulate some autonomous vehicles and set ”T-shirt” and ”pass” as the trigger and target label, respectively. After local training, it uploads the malicious models to the server to perform a backdoor attack for the obstacle avoidance model. In this sense, if an (honest) autonomous driving encounters pedestrians with a “T-shirt”, the vehicle could fail to stop.
Existing backdoor attacks against FL face such open problems:

1.1. P1: The abnormality of feature extraction.

Existing attacks use patch-based triggers (such as “squares”, “stripe” patterns) on a fixed position or semantic backdoor triggers in all poisoned images domain and guide the classification model to misclassify the images to the target label, which do not fully consider the “stealthiness” of triggers. The backdoor training forces the classification model to focus more on the location where the trigger pattern exists. Given a target label, the classification model needs to generate the independent latent representation of the trigger pattern to match it. Therefore, unrestricted trigger patterns can cause aberrant changes in the weights/biases of convolutional layers of the classification model. Thus this results in the abnormality in the feature extraction layer of the classification model.

1.2. P2: The abnormality of label mapping.

The backdoor training is to establish a connection between a specific trigger pattern and its corresponding target label. In this process, compared with the benign model, the malicious model needs to be trained on one more task, i.e. backdoor task, mapping the independent hidden features of the trigger pattern to the target label. Although there is a connection between the hidden features of benign samples and the target label in the benign task, backdoor task can force the last several fully-connected (FC) layers of the attacker’s model to build a new path between hidden features of triggers and the target label, which yields an anomaly at the parameter level. The cause of this anomaly is natural, because the output neuron for the target label must contribute to both two mappings, which requires significant weight/bias adjustments to the neurons involved. We note that the final FC layer (or blocks in ResNet) in the current mainstream classification model is always with a large fraction of the total number of parameters (e.g., 98% for Classic CNN of Fashion-MNIST, 62% in ResNet18 of CIFAR-10). As mentioned in [28], the final layer of the malicious classifier presents significantly greater abnormality than other layers, with label mapping being seen as the secondary source of these abnormalities. Note that a similar phenomenon would arise in **P1-2** if we use a semantic or natural pattern as the backdoor trigger.

1.3. P3: The perceptible trigger for testing.

The test input with perceptible perturbation in FL [1, 36, 40] can be easily identified by an evaluator or a user who can distinguish the difference between 'just' an incorrect classification/prediction of the model and the purposeful wrong decision due to a backdoor in the test/use stage.

P1-2 can fatally harm the backdoor accuracy under robust FL frameworks due to the local update dissimilarity. For example, FLAME [21] can easily detect malicious updates of almost all prior attacks and identify their distinguishable parameter dissimilarity and thus eliminate the attack effectiveness. Meanwhile, the test/backdoor attack stage cannot perform properly because their triggers are not sufficiently hidden. We recall that DBA [36] and Neurotoxin [40] use a visible fixed trigger pattern that can be clearly detected by human inspection as in figure 1.2. In this work, we regard the problems **P1-3** as the *stealthiness* of backdoor attacks in the context of FL.

There exists an impossibility for current backdoor attacks against FL to provide adequate stealthiness of training data/model at update/test stage simultaneously. A natural question arises: *could we eliminate the anomalies introduced by backdoor training **(i.e., tackling P1-2)** while making the trigger sufficiently stealthy for evaluation on decentralized scenario **(i.e., addressing P3)**?*

To provide a concrete answer, we propose a new backdoor attack, called FTA, to guarantee stealthiness on decentralized setup, by well designing a trigger generator. Compared with prior attacks using predefined trigger patterns, FTA can provide SOTA stealthiness via controlling the perturbation of trigger pattern and restraining the latent representation of poisoned samples to be similar to that of benign samples with the target label. We train a generative neural network as a learnable and adaptive trigger generator for the attacker to inject backdoors during local malicious training. Specifically, the generator can produce imperceptible triggers which are more flexible than predefined patch-based triggers of prior attacks **(P3)**. It is also learnt to produce triggers which ensure similar latent representations of poisoned samples to benign ones of target label **(P1)**. Besides, learning similar hidden features of poisoned sample to benign ones can naturally reduce the abnormality in **P2** since the features make the poisoned sample "look like" a benign one of target label. Therefore, we can guarantee stealthiness in terms of the parameter similarity between benign and malicious models whereas prior works amplify the anomalies in **P1-2** during backdoor training. Finally, to make the flexible trigger robust and adaptive to the changes in global model, the generator is continuously trained under different global models across rounds.

We formulate the process of finding the optimal trigger generator and the malicious model in a constrained optimization problem. Then, we propose a simple and practical optimization process to solve this non-convex and constrained optimization problem. We illustrate learning the trigger generator, training the malicious model and testing the backdoor in figure 1.1. We further showcase various backdoor images in Figure 1.2 to demonstrate the imperceptible perturbation by our generator.

Our main **contributions** are summarized as follows:

• We design a new learnable and adaptive generator that can produce a visually imperceptible flexible trigger pattern which naturally reduces the anomaly of parameters of malicious model. We can establish a mapping between poisoned samples injected by the trigger and target label, akin to the connection between benign samples and its ground truth label (target label). We then propose a non-convex and constrained optimization that can efficiently learn the trigger generator and poison the model.

• We propose a new robust backdoor attack against FL defenses that demonstrates effectiveness and stealthiness, enabling attacker to inject flexible triggers produced by our trigger generator into benign samples at training stage and making the malicious model parameters indistinguishable from benign agents and fooling the global model to predict the target label when invisible trigger appears.

• Finally, we present intensive experiments to empirically demonstrate that the proposed attack provides state-of-the-art performance and robustness against existing eight well-study defense mechanisms under four benchmark datasets.

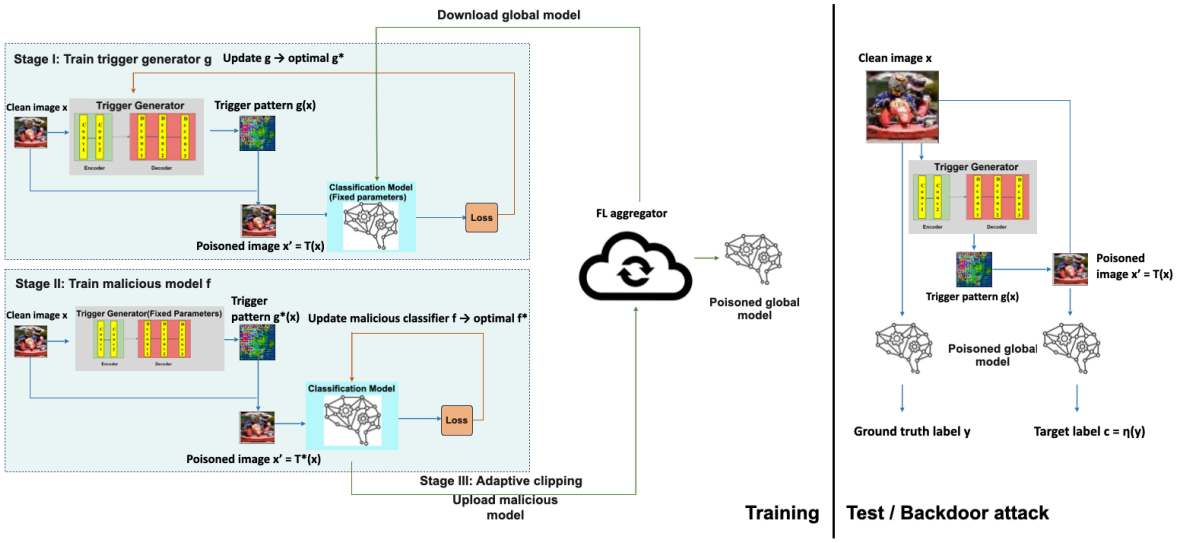


Figure 1.1: Overview of FTA. (I) Learn the optimal trigger generator g_ξ. (II) Train malicious model f_θ. (III) Adaptively clip the malicious update for anomaly detection. Test/Backdoor Attack: The global model performs well on benign tasks while misclassifying the poisoned samples to the target label.

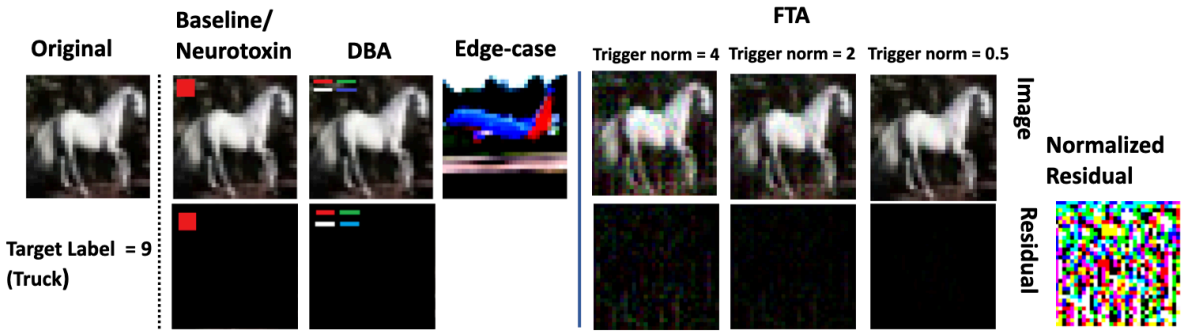


Figure 1.2: Visualization of backdoored images. **Top**: the original image; backdoored samples generated by baseline/Neurotoxin, DBA, Edge-case, and FTA; **Bottom**: the residual maps.

2

RELATED WORK

2.1. Federated Learning

Consider the empirical risk minimization (ERM) in FL setting where the goal is to learn a global classifier $f_\theta : \mathcal{X} \to \mathcal{Y}$ that maps an input $x \in \mathcal{X}$ to a target label $y \in \mathcal{Y}$. Recall that the FL server cannot access to training dataset. It aggregates the parameters/gradients from local agents performing centralized training with local datasets. The de-facto standard rule for aggregating the updates is so-called `FedAvg` [19]. The training task is to learn the global parameters θ by solving the finite-sum optimization: $\min_\theta f_\theta = \frac{1}{n}\sum_{i=1}^{n} f_{\theta_i}$, where n is the number of participating agents. At round t, the server S randomly selects $n^t \in \{1, 2, ..., n\}$ agents to participate in the aggregation and send the global model θ^t to them. Each of the agents i trains its local classifier $f_{\theta_i} : \mathcal{X}_i \to \mathcal{Y}_i$ with its local dataset $D_i = \{(x_j, y_j) : x_j \in \mathcal{X}_i, y_j \in \mathcal{Y}_i, j = 1, 2, ..., N\}$ for some epochs, where $N = |D_i|$, by certain optimization algorithm, e.g., stochastic gradient descent (SGD). The objective of agent i is to train a local model as: $\theta_i^* = \underset{\theta^t}{\text{argmin}} \sum_{(x_j,y_j)\in D_i} \mathcal{L}(f_{\theta^t}(x_j), y_j)$, where $\mathcal{L}$ stands for the classification loss, e.g., cross-entropy loss. Then agent i computes its local update as $\delta_i^t = \theta_i^* - \theta^t$, and sends back to S. Finally, the server aggregates all updates and produces the new global model with an average $\theta^{t+1} = \theta^t + \frac{\gamma}{|n^t|}\sum_{i\in n^t} \delta_i^t$. where γ is the global learning rate. When the global model θ converges or the training reaches a specific iteration upper bound, the aggregation process terminates and outputs a final global model. During inference, given a benign sample x and its true label y, the learned global classifier f_θ will behave well as: $f_\theta(x) = y$.

Optimizations of FL have been proposed for various purposes, e.g., privacy [5], security [4, 41], heterogeneity [16], communication efficiency [18, 13] and personalization issues [15, 39].

2.2. Backdoor Attacks on FL

The most well-known backdoor attack on FL is introduced in [1], where the adversary scales up the weights of malicious model updates to maximize attack impact and replace the global model with its malicious local model. To fully exploit the distributed learning methodology of FL, the local trigger patterns are used in [36] to generate poisoned images for different malicious models, while the data from the tail of the input data distribution is leveraged in [34]. Durable backdoor attacks are proposed in [40], and make attack itself more persistent in the federated scenarios. We state that this kind of attacks mainly focuses on the persistence, whereas our focus is on stealthiness.

Existing works reply on a universal trigger or tail data, which do not fully exploit the “attribute” of trigger. Our design is fully applicable and complementary to prior attacks. By learning a stealthy trigger generator and injecting the sample-specific triggers, we can significantly decrease the anomalies in **P1-3** and reinforce the stealthiness of backdoor attacks.

2.3. Backdoor Defenses on FL

There are a number of defenses that provide empirical robustness against backdoor attacks.

Dimension-wise filtering. Trimmed-mean [38] aggregates each dimension of model updates of all agents independently. It sorts the parameters of the j^{th}-dimension of all updates and removes m of the largest and smallest parameters in that dimension. Finally, it computes the arithmetic mean of the rest parameters as the aggregate of dimension j. Similarly, Median [38] takes the arithmetic median value of each dimension for aggregation. SignSGD [2] only aggregates the signs of the gradients (of all agents) and returns the sign to agents for updating the local models.

Vector-wise scaling. Norm clipping [32] bounds the l_2-norm of all updates to a fixed threshold due to high norms of malicious updates. For a threshold τ and an update ∇, if the norm of the update $||\nabla|| > \tau$, ∇ is scaled by $\frac{\tau}{||\nabla||}$. The server averages all the updates, scaled or not, for aggregation.

Vector-wise filtering. Krum [4] selects a local model, with the smallest Euclidean distance to $n - f - 1$ of other local models, as the global model. A variant of Krum called Multi-Krum [4] selects a local model using Krum and removes it from the remaining models repeatedly. The selected model is added to a selection S until S has c models such that $n - c > 2m + 2$, where n is the number of selected models and m is the number of malicious models. Finally, Multi-Krum averages the selected model updates. RFA [27] aggregates model updates and makes FedAvg robust to outliers by replacing the averaging aggregation with an approximate geometric median.

Certification. CRFL [35] provides certified robustness in FL frameworks. It exploits parameter clipping and perturbing during federated averaging aggregation. In the test stage, it constructs a "smoothed" classifier using parameter smoothing. The robust accuracy of each test sample can be certified by this classifier when the number of compromised clients or perturbation to the test input is below a certified threshold.

Sparsification. SparseFed [24] performs norm clipping to all local updates and averages the updates as the aggregate. Top_k values of the aggregation update are extracted and returned to each agent who locally updates the models using this sparse update.

Cluster-based filtering. Recently, [21] proposed a defending framework FLAME based on the clustering algorithm (HDBSCAN) which can cluster dynamically all local updates based on their cosine distance into two groups separately. FLAME uses weight clipping for scaling-up malicious weights and noise addition for smoothing the boundary of clustering after filtering malicious updates. By using HDBSCAN, [28] designed a robust FL aggregation rule called DeepSight. Their design leverages the distribution of labels for the output layer, output of random inputs, and cosine similarity of updates to cluster all agents' updates and further applies the clipping method.

3

THREAT MODEL AND INTUITION

3.1. Threat model

Attacker's Knowledge & Capabilities: We consider the same threat model as in prior works [1, 3, 34, 40, 30, 24], where the attacker can have full access to malicious agent device(s), local training processes and training datasets. Furthermore, we do not require the attacker to know the FL aggregation rules applied in the server.

Attacker's Goal: Unlike untargeted poisoning attacks [12] preventing the convergence of the global model, the goal of our attack is to manipulate malicious agents' local training processes to achieve high accuracy in the backdoor task without undermining the accuracy of the benign task.

3.2. Our Intuition

Recall that prior backdoor attacks use universal trigger patterns (see figure 1.2) which cannot guarantee stealthiness (**P1-3**) since the poisoned samples are visually inconsistent with natural inputs. These triggers with noticeable modification can introduce abnormality of weights/biases in convolutional layers and further influence the process of label mapping. Consequently, this makes prior attacks be easily detected by current robust defenses due to **P1-2**. Also, the inconsistency between benign and poisoned samples is not stealthy for the attacker to test/backdoor the global model (**P3**).

To address **P1-3**, we propose a learnable trigger generator to produce flexible and stealthy triggers. Our adaptive generator provides four advantages over prior works: 1) the poisoned dataset has imperceptible perturbation by restraining the trigger norm; 2) the produced triggers are flexible and sample-specific other than being uniformly defined; 3) our learning objective for stealthy trigger function enables the triggers to achieve hidden feature similarity between poisoned and benign samples of target label; 4) the generator can keep learning across different FL rounds to generate flexible triggers with the best attack performance under current global model. By using such a trigger generator, we propose a new backdoor attack where the poisoned images are crafted from clean images with unnoticeable modifications while eliminating the two anomalies introduced by backdoor task. We advance the state-of-the-art by enhancing the stealthiness and effectiveness of the backdoor attack even against well-studied defenses.

3.2.1. v.s. Trigger generators in centralized setting.

One may argue that the attacker can simply apply a (trigger) generator in centralized setup [9, 8, 42, 17] on FL to achieve stealthy trigger and model updates.

- **Stealthiness.** For example, the attacker can use a generator to produce imperceptible triggers for poisoned samples and make their hidden features similar to original benign samples' as in [42]. This, however, cannot ensure the indistinguishable perturbation of model parameters during malicious training and fail to capture the stealthiness (in **P1-2**). This is so because it only constrains the distinction of the input domain and the hidden features between poisoned and original benign samples other than the hidden features between poisoned and benign samples of target label.

In other words, a centralized generator masks triggers in the input domain and feature space of benign samples, conceals the poisoned sample for visibility and latent representation, whereas this cannot ensure malicious and benign models are indistinguishable. A stealthy backdoor attack on FL should mitigate the two anomalies introduced by backdoor task training and guarantee the stealthiness of model parameters instead of just the hidden features of poisoned samples compared to their original inputs.

- **Learning.** The methods of learning the trigger generator cannot directly apply to decentralized setups due to the continuously changing of global model and time consumption of poisoning local model. As an example, LIRA [9] utilizes alternating optimization procedures to learn the optimal generator and malicious classification model for poisoning. This approach incurs high time costs in FL and the classification model to be poisoned should change in each round. In contrast, we propose a customized optimization method for the FL scenario that can learn the optimal trigger generator for global model of current round to achieve the best attack effectiveness as depicted in section 4.3.

- **Defenses.** We note that the robust FL aggregator can only access local updates of all agents other than local training datasets. The centralized backdoor attack does not require consideration of the magnitude of the parameters. However, in reality, the magnitude of malicious updates is usually larger than that of benign updates in FL. In that regard, norm clipping can effectively weaken and even eliminate the impact of the backdoor. That is why we use adaptive norm clipping for evading the detection of robust FL algorithms.

4

PROPOSED METHODOLOGY:FTA

4.1. Problem Formulation

Based on the federated scenario in section 2.1, the attacker m trains the malicious models to alter the behavior of the global model θ under ERM as follows: $\theta_m^* = \underset{\theta}{\text{argmin}} \sum_{(x,y)\in D^{cln}\cup D^{bd}} \mathcal{L}(f_\theta(x), y)$, where D^{cln} is clean training set and D^{bd} is a small fraction of clean samples in D^{cln} to produce poisoned data by the attacker. Each clean sample (x, y) in the selected subset is transformed into a poisoned sample as $(\mathcal{T}(x), \eta(y))$, where $\mathcal{T} : \mathcal{X} \to \mathcal{X}$ is the trigger function and η is the target labeling function. And the poison fraction is defined as $|D^{bd}|/|D^{cln}|$. During inference, for a clean input x and its true label y, the learned f behaves as: $f(x) = y, f(\mathcal{T}(x)) = \eta(y)$.

To generate a stealthy backdoor, our main goal is to learn a stealthy trigger function $\mathcal{T} : \mathcal{X} \to \mathcal{X}$ to craft poisoned samples and a malicious backdoor model $f_{\theta_m^*}$ to inject backdoor behavior into the global model with the followings: 1) the poisoned sample $\mathcal{T}(x)$ provides an imperceptible perturbation to ensure that we do not bring distribution divergences between clean and backdoor datasets; 2) the injected global classifier simultaneously performs indifferently on test input x compared to its vanilla version but changes its prediction on the poisoned image $\mathcal{T}(x)$ to the target class $\eta(y)$; 3) the latent representation of backdoor sample $\mathcal{T}(x)$ is similar to its benign input x. Inspired by recent works in learning trigger function backdoor attacks [6, 9, 22, 42], we propose to jointly learn $\mathcal{T}(\cdot)$ and poison f_θ via the following constrained optimization:

$$
\begin{aligned}
&\min_{\theta} \sum_{(x,y)\in D^{cln}} \mathcal{L}(f_\theta(x), y) + \sum_{(x,y)\in D^{bd}} \mathcal{L}(f_\theta(\mathcal{T}_{\xi^*(\theta)}(x)), \eta(y)) \\
s.t. \quad &(i) \quad \xi^* = \underset{\xi}{\text{argmin}} \sum_{(x,y)\in D^{bd}} \mathcal{L}(f_\theta(\mathcal{T}_\xi(x)), \eta(y)) \\
&(ii) \quad d(\mathcal{T}_\xi(x), x) \le \epsilon
\end{aligned}
\tag{4.1}
$$

where d is a distance measurement function, ϵ is a constant scalar threshold value to ensure a small perturbation by l_2-norm constraint, ξ is the parameters of trigger function $\mathcal{T}(\cdot)$. In the above bilevel problem, we optimize a generative trigger function $\mathcal{T}_{\xi^*}$ that is associated with an optimally malicious classifier. The poisoning training finds the optimal parameters θ of the malicious classifier to minimize the linear combination of the benign and backdoor objectives. Meanwhile, the generative trigger function is trained to manipulate poisoned samples with imperceptible perturbation, while also finding the optimal trigger that can cause misclassification to the target label. The optimization in equation (4.1) is a challenging task in FL scenario since the target classification model f_θ varies in each iteration and its non-linear constraint. Thus, the learned trigger function $\mathcal{T}_\xi$ is unstable based on dynamic f_θ. For the optimization, we consider two steps: learning trigger generator and poisoning training, and further execute these steps respectively (not alternately) to optimize f_θ and $\mathcal{T}_\xi$. The details are depicted in algorithm 1 (see section 4.4).

4.2. FTA Trigger Function

We train $\mathcal{T}_\xi$ based on a given generative classifier g_ξ, i.e., our FTA trigger generator. Similar to the philosophy of generative trigger technology [9, 42], we design our trigger function to guarantee: 1) The perturbation of poisoned sample is imperceptible; 2) The trigger generator can learn the features of input domain of target label to fool the global model. Given a benign image x and the corresponding label y, we formally model $\mathcal{T}_\xi$ with restricted perturbation as follows:

$$\mathcal{T}_\xi(x) = x + g_\xi(x), \quad \|g_\xi(x)\|_2 \leq \epsilon \quad \forall x, \quad \eta(y) = c, \tag{4.2}$$

where ξ is the learnable parameters of the FTA trigger generator and ϵ is the trigger norm bound to constrain the value of the generative trigger norm. We use the same neural network architecture as [9] to build our trigger generator g_ξ, i.e., an autoencoder or more complex U-Net structure [29]. The l_2-norm of the imperceptible trigger noise generated by g_ξ is strictly limited within ϵ by: $\frac{g_\xi(x)}{max(1, \|g_\xi(x)\|_2/\epsilon)}$. Note that, under equation (4.2), the distance d in equation (4.1) is l_2-norm on the image-pixel space between $\mathcal{T}_\xi(x)$ and x.

4.3. FTA's Optimization

To address the non-convex and constrained optimization in equation (4.1), one may consider alternately updating f_θ while keeping $\mathcal{T}_\xi$ unchanged, or the other way round. However, according to our trials, we find that simply updating the parameters makes the training process unstable and harms the backdoor performance. Inspired by [10, 9], we divide the local malicious training into two phases. In phase one, we fix the classification model f_θ and only learn the trigger function $\mathcal{T}_\xi$. In phase two, we use the pre-trained $\mathcal{T}_{\xi^*}$ to generate the poisoned dataset and train the malicious classifier f_θ. Since the number of poisoning epochs of malicious agents is fairly small, which means f_θ would not vary too much during poisoning training process, the hidden features of samples in target label extracted from f_θ will also remain similarly. The pre-trained $\mathcal{T}_{\xi^*}$ can still match with the final locally trained f_θ.
In order to make flexible triggers generated by g_ξ adaptive to global models in different rounds, g_ξ should be continuously trained. If a malicious agent is selected more than one round to participate in FL iterations, it can keep training on the previous pre-trained g_ξ under new global model to make the flexible trigger produced by g_ξ match with hidden features of benign samples with target label from new model.

4.4. The procedure of FTA optimization

In case of collusion between more than one malicious agent device, the local datasets owned by these devices are in non-i.i.d. manner. Their local trigger generators g_{ξ_i} are trained by these local datasets. This kind of dataset bias can degrade attack effectiveness since their malicious updates are for local triggers from different g_{ξ_i} and cannot be merged together to yield a better attack performance. To resolve this problem, we develop a practical solution. Before starting the FTA backdoor attack, the malicious agents can share a portion of their local datasets to form a universal poisoned dataset (for all the malicious agents), so that their local generators g_{ξ_i} can produce the same triggers.

4.5. Adaptive Norm Clipping for Anomaly Detection

Once we find the optimal malicious model f_{θ^*} from equation (4.1), we use an adaptive norm clipping based on [32] to camouflage our malicious model to evade anomaly detection. We first compute the malicious update $\delta = \theta^* - \theta$, where θ is the parameters of downloaded global model. Then we train a benign reference model $\hat{\theta}$ with attacker's clean training data D_{cln} and compute the reference update $\hat{\delta} = \hat{\theta} - \theta$. At last, we adaptively clip the malicious update as $\frac{\delta}{max(1, \|\delta\|_2/\|\hat{\delta}\|_2)}$.

Algorithm 1 FTA Backdoor Attack

Input: Clean dataset D_{cln}, Downloaded global model f_θ, Learning rate to train malicious classifier γ_f, Learning rate to train trigger function $\gamma_\mathcal{T}$, Batch of clean dataset B_{cln}, Batch of poisoned dataset B_{bd}, Number of epochs to train trigger function $e_\mathcal{T}$, Number of epochs to train malicious classifier e_f.
Output: Malicious update δ^* manipulated by the attacker.

Initialize parameters of trigger function ξ and download global model: f_θ.
Sample subset D_{bd} from D_{cln}.
// Stage I: Update flexible $\mathcal{T}$.
Sample minibatch $(x, y) \in B_{bd}$ from D_{bd}
for $i = 1, 2, \cdots, e_\mathcal{T}$ **do**
    Optimize ξ by using SGD with fixed classifier f_θ on B_{bd}: $\xi \leftarrow \xi - \gamma_\mathcal{T} \nabla_\xi \mathcal{L}(f_\theta(\mathcal{T}_\xi(x)), \eta(y))$
end for
$\xi^* \leftarrow \xi$
// Stage II: Training malicious model f.
Sample minibatch $(x, y) \in B_{cln}$ from D_{cln} and $(x_m, y_m) \in B_{bd}$ from D_{bd}
for $i = 1, 2, \cdots, e_f$ **do**
    Optimize θ by using SGD with fixed trigger function $\mathcal{T}_{\xi^*}$ on B_{cln} and B_{bd}: $\theta \leftarrow \theta - \gamma_f \nabla_\theta (\mathcal{L}(f_\theta(x, y)) + \mathcal{L}(f_\theta(\mathcal{T}_\xi(x_m)), \eta(y_m)))$
end for
$\theta^* \leftarrow \theta$
// Stage III: Adaptive norm clipping for anomaly detection.
Sample minibatch $(x, y) \in B_{cln}$ from D_{cln}
for $i = 1, 2, \cdots, e_f$ **do**
    Optimize θ by using SGD on B_{cln}: $\theta \leftarrow \theta - \gamma_f \nabla_\theta \mathcal{L}(f_\theta(x, y))$
end for
$\hat{\theta} \leftarrow \theta$
Compute malicious update: $\delta \leftarrow \theta^* - \theta$
Compute benign reference update: $\hat{\delta} \leftarrow \hat{\theta} - \theta$
Adaptive clipping: $\delta^* \leftarrow \frac{\delta}{max(1, \|\delta\|_2 / \|\hat{\delta}\|_2)}$

5

ATTACK EVALUATION

We show that FTA outperforms the current SOTA attacks (under robust FL defenses) by conducting experiments on different computer vision tasks.

5.1. Experimental Setup

5.1.1. Datasets and Models.

We demonstrate the effectiveness of FTA backdoor through comprehensive experiments on four publicly available datasets, namely Fashion-MNIST, FEMNIST, CIFAR-10, and Tiny-ImageNet. The classification model used in the experiments includes Classic CNN models, VGG11 [31], and ResNet18 [11]. These datasets and models are representative and commonly used in existing backdoor and FL research works.

5.1.2. The structure of our models

- **The structure of classification models for Fashion-MNIST and FEMNIST** We use an 8-layer classic CNN architecture for training Fashion-MNIST and FEMNIST datasets. The details are shown in table 5.1.

Table 5.1: The structure of classic CNN model.

Parameters	Shape	Hyperparameters of layer
Conv2d	1*32*3*3	stride = (1, 1)
GroupNorm	32*32	eps = 10^{-5}
Conv2d	32*64*3*3	stride = (1, 1)
GroupNorm	32*64	eps = 10^{-5}
Dropout2d		p = 0.25
Linear	9216*128	bias = True
Linear(For Fashion-MNIST)	128*10	bias = True
Linear(For FEMNIST)	128*62	bias = True

- **The structure of trigger generator** In the FTA framework, the trigger generator plays a crucial role in feature extraction in the sense that it aims to align the hidden features of poisoned samples with the target label samples. We utilize the Autoencoder as the trigger generator due to its ability to capture essential features of input and generate outputs satisfying our needs. Moreover, we find that U-Net exhibits comparable performance for trigger generation while requiring less training data, as stated in [9]. Therefore, we include U-Net in our experiments. Both U-Net and autoencoder architectures used to train the trigger generator g_ξ are similar to those presented in [9].

- **The structure of classification models for CIFAR-10 and Tiny-ImageNet** We use a similar ResNet-18 architecture as in [36] for training CIFAR-10 and Tiny-ImageNet.

5.1.3. Tasks.

There are *4* computer vision tasks in total using different datasets, classification models, and trigger generators respectively. The details are depicted in table 5.2. To prove the stealthiness and further robustness against defenses of FTA, we use a decentralized setting with non-i.i.d. data distribution among all agents. The attacker chooses the all-to-one type of backdoor attack (except Edge-case [34]), fooling the global model to misclassify the poisoned images of any label to an attacker-chosen target label. Following a practical scenario for the attacker given in [40], *10* agents among thousands of agents are selected for training in each round and their updates are used for aggregation and updating the server model. We apply backdoor attacks from different phases of training. In FEMNIST task, we follow the same setting as [36], where the attacker begins to attack when the benign accuracy of global models starts to converge. For other tasks, we perform backdoor attacks at the beginning of FL training. In this sense, as mentioned in [36], benign updates are more likely to share common patterns of gradients and have a larger magnitude than malicious updates, which can significantly restrict the effectiveness of malicious updates. Note we consider such a setting for the bottom performance of attacks and further, we still see that our attack performs more effectively than prior works in this case (see section 5.2).

Table 5.2: The datasets, and their corresponding models and hyperparameters.

	Fahion-MNIST	FEMNIST	CIFAR-10	Tiny-ImageNet
Classes	62	10	10	200
Size of training set	60000	737837	50000	100000
Size of testing set	10000	80014	10000	10000
Total agents	2000	3000	1000	2000
Malicious agents	2	3	1	2
Agents per FL round	10	10	10	10
Phase to start attack	Attack from scratch	Attack after convergence	Attack from scratch	Attack from scratch
Poison fraction	0.2			
Trigger size	2	1.5	1.5	3
Dataset size of trigger generator	1024			
Epochs of benign task	2	4	5	5
Epochs of backdoor task	5	10	10	10
Learning rate of trigger generator	0.01	0.01	0.001	0.01
Epochs of trigger generator	20	20	30	30
Local data distribution	non-i.i.d.			
Classification model	Classic CNN	Classic CNN	Resnet18	Resnet18
Trigger generator model	AutoEncoder	AutoEncoder	Unet	AutoEncoder
Learning rate of benign task	0.1	0.01	0.01	0.001
Learning rate of backdoor task	0.1	0.01	0.01	0.01
Edge-case	FALSE	TRUE	TRUE	FALSE
Other hyperparameters	Momentum:0.9, Weight Decay: 10^{-4}			

5.1.4. Experiment settings.

The implementation of all the compared attacks and FL framework are based on PyTorch [25]. We test the experiments on a server with one Intel Xeon E5-2620 CPU and one NVIDIA A40 GPU with 32G RAM.

In Fashion-MNIST, CIFAR-10 and Tiny-ImageNet, a Dirichlet distribution is used to divide training data for the number of total agent parties, and the hyperparameter for distribution is 0.7 for the datasets. For FEMNIST, we randomly choose data of 3000 users from the dataset and randomly distribute every

training agent with the training data from 3 users. All parties use SGD as an optimizer and train for local training epochs with a batch size of 256. A global model is shared by all agents, and updates of 10 agents will be selected for aggregating the global model. Benign agents train with a benign learning rate for benign epochs. The attacker's local training dataset is mixed with 80% correct labeled data and 20% poisoned data. The target labels are "sneaker" in Fashion-MNIST, "digit 1" in FEMNIST, "truck" in CIFAR-10 and "tree frog" in Tiny-ImageNet. The attacker has its own local malicious learning rate and epochs to maximize its backdoor performance. It also needs to train its local trigger generator with learning rate and epochs before performing local malicious training on the downloaded global model. Regarding the attack methods, we set the top-k ratio of 0.95 for Neurotoxin, in line with the recommended settings in [40]. For DBA, we use 4 distributed strips as backdoor trigger patterns. Both the baseline attack and Neurotoxin employ a "square" trigger pattern on the top left as the backdoor trigger. We conduct Edge-case attack on CIFAR-10 and FEMNIST. Specifically, for CIFAR-10, we use the southwest airplane as the backdoored images and set the target label as "bird". For FEMNIST, we use images of "7" in ARDIS [14] as poisoned samples with the target label set as the digit "1". The dataset settings of the experiments are the same as those used in [34].

5.1.5. Attack Settings.

As in [40], we assume that the attacker can only compromise a limited number of agents (<1%) in practice [30] and uses them to launch the attack by uploading manipulated gradients to the server. Malicious agents can only participate in a constrained number of training rounds in FL settings. Note even if the attacker has the above restrictions, our attack can still be effective, stealthy and robust against defenses (see sections 5.2 and 5.3). Also, the effectiveness of the attack should last even though the attacker stops the attack under robust FL aggregators (see figure 5.2 in section 5.2.2). We test the stealthiness and durability of FTA with two attack modes respectively, i.e., fixed-frequency and few-shot as [40].

- **Fixed-frequency mode.** The server randomly chooses 10 agents among all agents. The attacker controls exactly one agent in each round in which they participate. For other rounds, 10 benign agents are randomly chosen among all agents.
- **Few-shot mode.** The attacker participates only in **Attack_num** rounds. During these rounds, we ensure that one malicious agent is selected for training. After Attack_num rounds or backdoor accuracy has reached 95%, the attack will stop. Under this setting, the attack can take effect quickly, and gradually weaken by benign updates after the attack is stopped. In our experiments, the Attack_num is 100 for all attacks, and the total FL round is 1000 for CIFAR-10, and 500 for other datasets.

5.1.6. Evaluation Metrics.

We evaluate the performance based on backdoor accuracy (BA) and benign accuracy according to the following criteria: effectiveness and stealthiness against current SOTA defense methods under fixed-frequency mode, durability evaluated under few-shot mode.

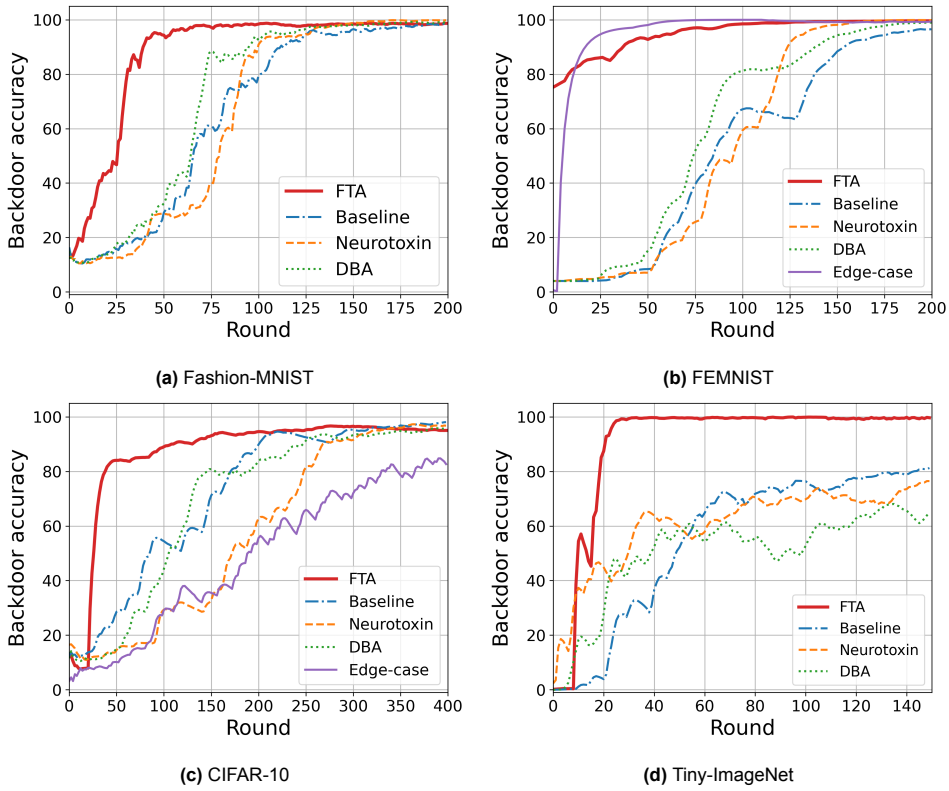
5.1.7. Comparison.

We compare FTA with three SOTA attacks, namely DBA, Neurotoxin and Edge-case [34], and the baseline attack method described in [40] under different settings and defenses. The results demonstrate that FTA delivers the best performance as compared to others.

5.2. Attack Effectiveness

5.2.1. Attack effectiveness under fixed-frequency mode.

Compared to the attacks with unified triggers, FTA converges much faster and delivers the best BA in all cases, see figure 5.1. It can yield a high backdoor accuracy on the server model within very few rounds (<50) and maintain above 97% accuracy on average. Especially in Tiny-ImageNet, FTA reaches 100% accuracy extremely fast, with at least 25% advantage compared to others. In CIFAR-10, FTA achieves nearly 83% BA after 50 rounds which is 60% higher than other attacks on average. There is only <5% BA gap between FTA and Edge-case on FEMNIST in the beginning and later, they reach the same BA



(a) Fashion-MNIST **(b)** FEMNIST

(c) CIFAR-10 **(d)** Tiny-ImageNet

Figure 5.1: Fixed-frequency attack performance under `FedAvg`. FTA is more effective than others.

after 100 rounds. We note that the backdoor task of Edge-case in FEMNIST is relatively easy, mapping 7-like images to the target label of digit "1", which makes its convergence slightly faster than ours.

5.2.2. Attack effectiveness under few-shot mode.

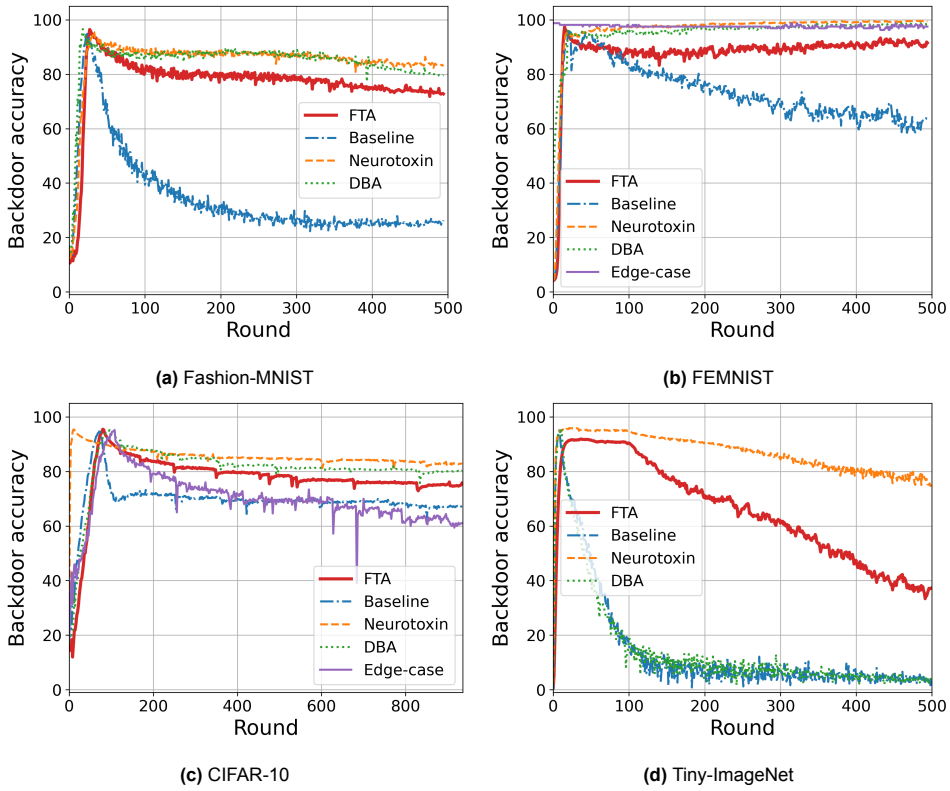
The results under few-shot settings are shown in figure 5.2. All attacks reach a high BA rapidly after consistently poisoning the server model, then BA gradually drops after stopping attacking and the backdoor injected into the server model is gradually weakened by the aggregation of benign updates. FTA's performance drops much slower than the baseline attack. For example, in Fashion-MNIST and after 500 rounds, FTA still remains 73% BA, which is only 9% less than Neurotoxin, 61% higher than the baseline. Moreover, FTA can beat DBA and the baseline on Tiny-ImageNet. After 500 rounds, FTA maintains 37% accuracy while the baseline and DBA only have 5%, which is 45% less than Neurotoxin. However, Neurotoxin cannot provide the same stealthiness as shown in following comparison under robust FL defenses. Since malicious and benign updates have a similar direction by FTA, the effectiveness of FTA's backdoor can survive after few-shot attack. The results prove the durability of FTA.

5.2.3. Influence on Benign accuracy.

Like other SOTA attacks, FTA has a minor effect (no more than 1.5%) on benign accuracy. We showcase the benign accuracy of both the baseline attack and FTA, and also consider the accuracy without backdoor attacks under `FedAvg`. We start FTA and the baseline from a specific round (e.g., 0 or 200 for different datasets) and perform the attacks during Attack_num rounds. We record the accuracy once the attacks have ended. From table 5.3, it is evident that FTA results in a slightly smaller decrease in the benign accuracy compared to baseline attack.

5.3. Stealthiness against Defensive Measures

We test the stealthiness **(P1-2)** and robustness of FTA and other attacks using 8 SOTA robust FL defenses introduced in section 2.3, including norm clipping, FLAME, Multi-Krum, Trimmed-mean, RFA,



(a) Fashion-MNIST **(b)** FEMNIST

(c) CIFAR-10 **(d)** Tiny-ImageNet

Figure 5.2: Few-shot attack performance under `FedAvg`. FTA is more durable than baseline.

Table 5.3: Benign accuracy of the baseline attack. FTA and no attackers circumstance under different datasets. Benign accuracy drops by $\leq$ 1.5% in FTA compared to the accuracy without attack.

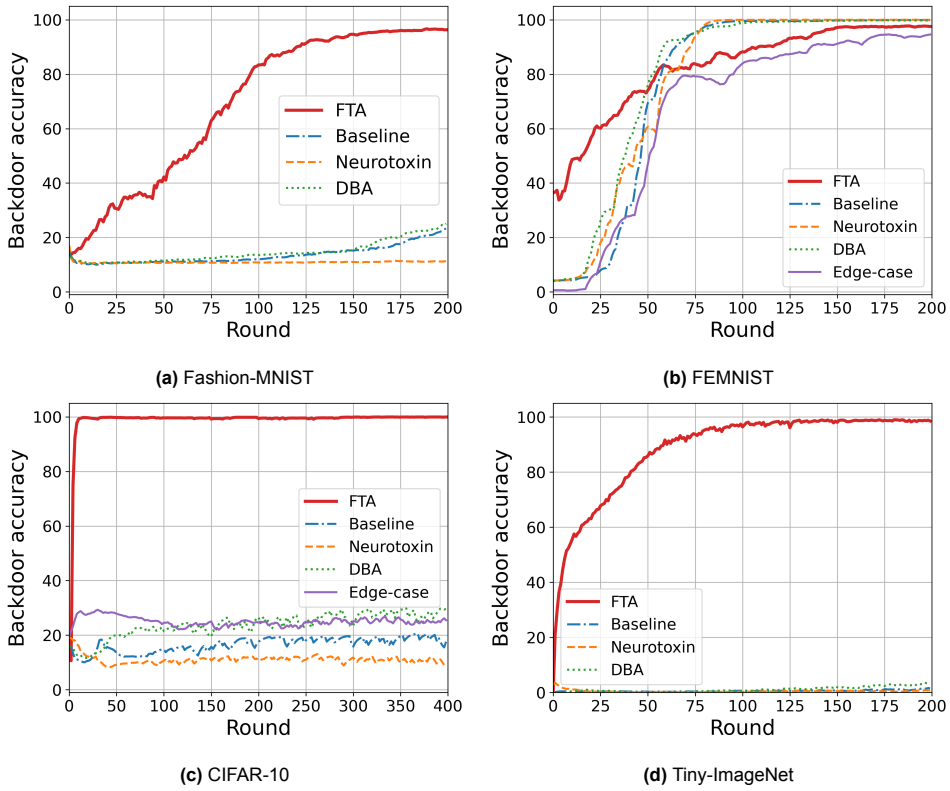
Dataset	Attack start epoch	Attack_num	No attack (%)	Baseline attack (%)	FTA (%)
Fashion-MNIST	0	50	90.21	85.14	90.02
FEMNIST	200	50	92.06	91.27	92.05
CIFAR-10	0	100	61.73	56.34	60.61
Tiny-ImageNet	0	100	25.21	19.06	25.13

SignSGD, Foolsgold and SparseFed under fixed-frequency scenarios. All four tasks are involved in this defense evaluation. The results show that FTA can break the listed defenses.

5.3.1. Resistance to Vector-wise Scaling

We use the norm clipping as the vector-wise scaling defense method, which is regarded as a potent defense and has proven effective in mitigating prior attacks [30]. On the server side, norm clipping is applied on all updates before performing `FedAvg`. Inspired by [21], we utilize the variant of this method in our experiments. As introduced in section 5.1.3, if we begin the attack from scratch, the norm of benign updates will be unstable and keep fluctuating, making us hard to set a fixed norm bound for all updates. We here filter out the biggest and smallest updates and compute the average norm magnitude based on the rest updates, and set it as the norm bound in current FL iteration.

As shown in figure 5.3, this variant of norm clipping can effectively undermine prior attacks in Fashion-MNIST, CIFAR-10, and Tiny-ImageNet. It fails in FEMNIST because benign updates have a larger norm (for example, 1.2 in FEMNIST at round 10, but only 0.3 in Fashion-MNIST), which cannot effectively clip the norm of malicious updates, thus resulting in a higher BA of existing attacks. We see that FTA provides the best BA which is less influenced by clipping than others. FTA only needs a much smaller norm to effectively fool the global model. Although converging a bit slowly in FEMNIST, FTA can finally



(a) Fashion-MNIST **(b)** FEMNIST

(c) CIFAR-10 **(d)** Tiny-ImageNet

Figure 5.3: The effectiveness of attack under norm clipping in 4 tasks.

output a similar performance (above 98%) compared to others.

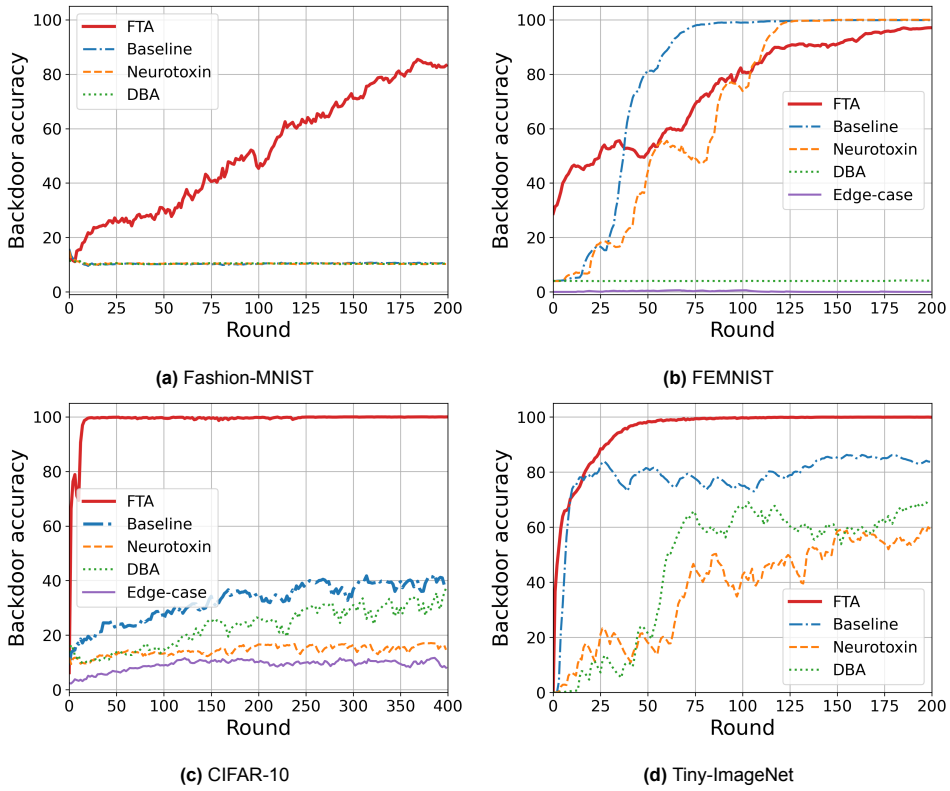
5.3.2. Resistance to Cluster-based Filtering

The cluster-based filtering defense method is FLAME [21], which has demonstrated its effectiveness in mitigating SOTA attacks against FL. It mainly uses HDBSCAN clustering algorithm based on cosine similarity between all updates and strains the updates with the least similarity compared with other updates. In figure 5.4, we see that FLAME can effectively sieve malicious updates of other attacks in Fashion-MNIST and CIFAR-10, but provides relatively weak effectiveness in FEMNIST and Tiny-ImageNet. This is so because data distribution among different agents are fairly in non-i.i.d. manner. Cosine similarity between benign updates is naturally low, making malicious update possibly evade from the clustering filter.

Similar to the result of Multi-Krum (see section 5.3.3), FTA achieves $>$99% BA and finishes the convergence within 50 rounds in CIFAR-10 and Tiny-ImageNet, while delivering an acceptable degradation of accuracy, <20%, in Fashion-MNIST. In FEMNIST, FTA converges slightly slower than the baseline and Neurotoxin but eventually maintains a similar accuracy with only 2% difference. The result proves that FTA enforces malicious updates to have highly cosine-similarity against benign updates due to the same reason in section 5.3.3, so that it can bypass the defenses based on similarity of updates.

5.3.3. Resistance to Vector-wise filtering

Multi-Krum is used as the vector-wise defense method. As described in section 2.3, it calculates the Euclidean distance between all updates and selects $n - f - 1$ updates with the smallest Euclidean distances for aggregation. In figure 5.5, the defense manages to filter out almost all malicious updates of prior attacks and effectively degrade their attacks' performance. In contrast, local update of FTA cannot be easily filtered and thus FTA outperforms others. In CIFAR-10 and Tiny-ImageNet, the attack performance is steady for FTA (nearly 100%) within 40 rounds to converge. In FEMNIST, Multi-Krum only results in a 10% BA degradation for FTA while BAs of others are restricted to 0%. In Fashion-MNIST, Multi-Krum can sieve malicious updates of FTA occasionally, leading to a longer convergence time, but still fails to completely defend the FTA. Malicious updates produced by FTA (which successfully



(a) Fashion-MNIST **(b)** FEMNIST

(c) CIFAR-10 **(d)** Tiny-ImageNet

Figure 5.4: The effectiveness of attack under FLAME in 4 tasks.

eliminates the anomalies in **P1-2**) are with a similar Euclidean distance compared to benign updates, making them more stealthy than other attacks'.

5.3.4. Resistance to Dimension-wise filtering

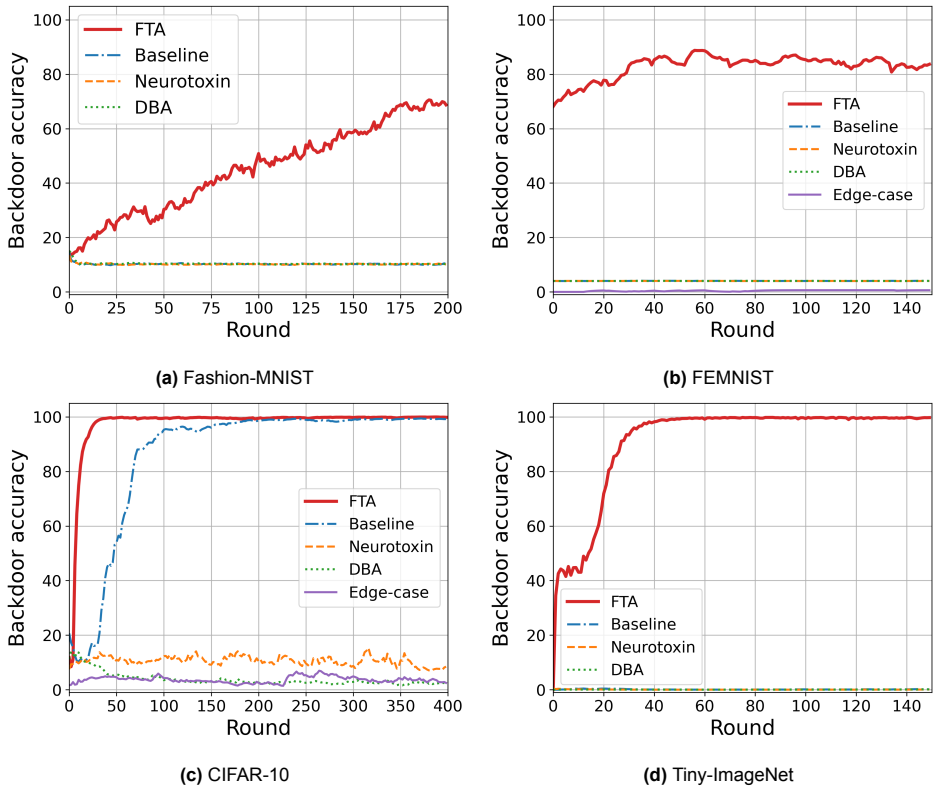
We choose Trimmed-mean as the representative of dimension-wise filtering. As mentioned in section 2.3, the dimensions of updates are sorted respectively, and the top m highest and smallest updates are removed, and the arithmetic mean of the rest parameters is computed for aggregated updates. In our experiments, m is set as 2 because we assume there is no more than one malicious agent during FL iteration, and setting a higher m can result in lower convergence. As shown in figure 5.6, Trimmed-mean successfully filters out the compared attacks in Fashion-MNIST and Tiny-ImageNet, and its effects are weakened in CIFAR-10 and FEMNIST. However, FTA survives in all four tasks and performs the best under trimmed-mean. In CIFAR-10, it completes the convergence within 30 rounds and remains 99.9% BA. In Fashion-MNIST and FEMNIST, FTA takes above 50 rounds to fully converge, and the final accuracy manages to reach 96%. The performance of FTA is significantly degraded in Tiny-ImageNet, but still with 30% advantage over other attacks on average. The update of FTA shares a similar weights/biases distribution of benign updates. This ensures our attack to defeat the defenses based on dimension-wise filtering.

5.3.5. Resistance to RFA

In figure 5.7, FTA provides the best performance among others in Fashion-MNIST, CIFAR-10 and Tiny-ImageNet. In FEMNIST, it converges much faster than prior attacks. Although its accuracy is 8% lower than the baseline in the middle of training, FTA achieves the same performance at the end (of training).

5.3.6. Resistance to SignSGD

As shown in figure 5.8 (a)-(b), SignSGD mitigates prior backdoor attacks with a universal trigger pattern. However, FTA still defeats it and remains 94% and 99% BA on Fashion-MNIST and Tiny-ImageNet,



(a) Fashion-MNIST **(b)** FEMNIST

(c) CIFAR-10 **(d)** Tiny-ImageNet

Figure 5.5: The effectiveness of attack under Multi-Krum in 4 tasks.

respectively.

5.3.7. Resistance to Foolsgold

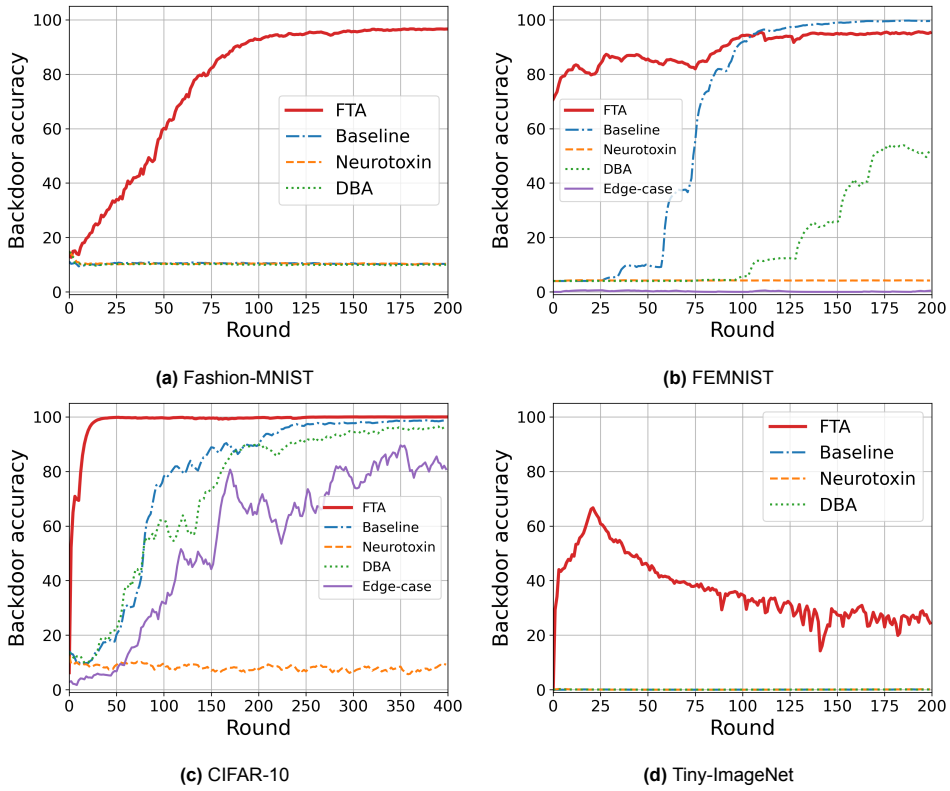
From figure 5.8 (c), we see that Foolsgold hinders the convergence speed of FTA in Fashion-MNIST, which requires FTA to perform extra 25 rounds for convergence. In this sense, FTA still converges much faster than others.

5.3.8. Resistance to Sparsification

We choose SparseFed as the representative of the sparsification defense. In figure 5.8 (d), only Neurotoxin and FTA are capable of breaking through SparseFed on Tiny-ImageNet. The BA of Neurotoxin exhibits fluctuations (between 22% and 36%) throughout the training process, unable to maintain a continuous rise. In contrast, FTA demonstrates the ability to consistently poison the global model and later achieves an impressive accuracy of 90% by round 150. The reason for the above performance difference is that the backdoor task of FTA captures imperceptible perturbations on model parameters, which eliminates the anomalies of poisoning training. The backdoor tasks trained by FTA are more likely to contribute to the same dimensions of gradients as benign updates. Consequently, the top-k filtering mechanism implemented in the server side is ineffective to filter out FTA's backdoor effect.

5.4. Explanation via Feature Visualization by t-SNE

We use t-SNE [33] visualization result on Fashion-MNIST to illustrate why FTA is more stealthy than the attacks without "flexible" triggers. We select 1,000 images from different classes uniformly and choose another 100 images randomly from the dataset and add triggers to them (in particular, patch-based trigger "square" in baseline method, flexible triggers in FTA). To analyze the hidden features of these samples, we use two global poisoned models injected by baseline attack and FTA respectively. We exploit the output of each sample in the last convolutional layer as the latent representation. Next, we apply dimensionality reduction techniques and cluster the latent representations of these samples using t-SNE. From figure 5.9 (a)-(b), We see that in the baseline, the Euclidean distance of clusters between



(a) Fashion-MNIST **(b)** FEMNIST

(c) CIFAR-10 **(d)** Tiny-ImageNet

Figure 5.6: The effectiveness of attack under Trimmed-mean in 4 tasks.

images of the target label "7" and the poisoned images are clearly distinguishable. So the parameters responsible for label mapping should do adjustments to map the hidden representations of poisoned images to target label. In FTA, the hidden features of poisoned images overlapped with benign images of target label, which eliminates the anomaly in **feature extraction (P1)**. FTA can reuse the path of benign tasks in the label mapping process, resulting in much less abnormality in **label mapping (P2)**, thus the malicious updates can be more similar to benign ones, see figure 5.9 (c)-(d), producing a natural stealthiness.

5.5. Ablation Study in FTA Attack

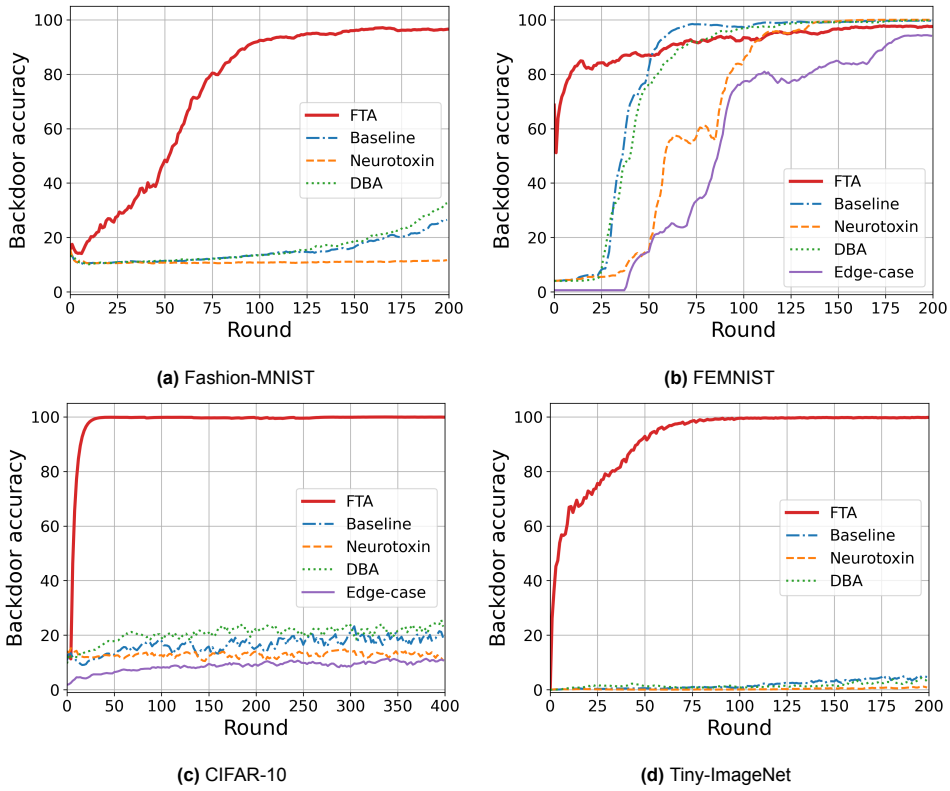
We here analyze several hyperparameters that are critical for the FTA's performance.

5.5.1. Trigger Size.

This size refers to the l_2-norm bound of the trigger generated by the generator, corresponding to ϵ in algorithm 1. If the size is set too large, the poisoned image can be easily distinguished (i.e., no stealthiness) by human inspection in test/evaluation stage. On the other hand, if we set it too small, the trigger will have a low proportion of features in the input domain. In this sense, the global model will encounter difficulty in catching and learning these features of trigger pattern, resulting in a drop of attack performance.

In figure 5.10, the trigger size significantly influences the attack performance in all the tasks. The accuracies of FTA drop seriously and eventually reach closely to 0% while we keep decreasing the size of the trigger, in which evidences can be seen in CIFAR-10, FEMNIST, and Tiny-ImageNet.

The sample-specific trigger with l_2-norm bound of 2 in CIFAR-10 and Tiny-ImageNet is indistinguishable from human inspection (see figure 5.11), while for Fashion-MNIST and FEMNIST (images with back-and-white backgrounds), additional noise can be still easily detected. Thus, a balance between visual stealthiness and effectiveness should be considered before conducting an FTA. The benign and poisoned samples with flexible triggers of different sizes generated by FTA are presented in figure 5.11. For Tiny-ImageNet and CIFAR-10, it is hard for human inspection to immediately identify the triggers,



(a) Fashion-MNIST **(b)** FEMNIST

(c) CIFAR-10 **(d)** Tiny-ImageNet

Figure 5.7: The effectiveness of attack under RFA in 4 tasks.

which proves the stealthiness in **P3**. In Fashion-MNIST and FEMNIST, the triggers are easier to distinguish because there is only one channel of the input samples in the datasets. But those flexible triggers are still much more stealthy compared to those produced by prior attacks on FL (see figure 1.2).

5.5.2. Poison Fraction.

This is the fraction of poisoned training samples in the training dataset of the attacker. Setting a low poison fraction can benefit the attack's stealthiness by having less abnormality in parameters and less influence on benign tasks. But this can slow down the attack effectiveness, as a side effect. Fortunately, we find that FTA can still take effect under a low poison fraction. We set the local training batch size to 256 for all the tasks, follow the standard settings of other FL frameworks, and set the poison fraction as 0.2. As stated in section 5.2.3, this fraction setting cannot degrade the performance of benign accuracy and meanwhile, we would like to explore further to examine the lower bound of the fraction which FTA's performance can tolerate. In figure 5.12, FTA is still effective whilst the fraction drops to 0.05. We also find that sensitivities to poison fraction can vary among tasks. In Fashion-MNIST and CIFAR-10, FTA remains its performance even if poison fraction $= 0.01$, in which only 3 samples are posoined in each batch. As for FEMNIST and Tiny-ImageNet, under the same rate, the backdoor tasks are dramatically weakened by the benign ones.

5.5.3. Dataset Size of Trigger Generator.

Theoretically, if this dataset is small-scale, the trigger generator could not be properly trained, thus resulting in bad quality and further endangering the attack performance. From figure 5.13, we see that this concern should not be crucial for FTA. During the training, if the attacker controls multiple agents, it can merge all local datasets into one for generator training. However, in many cases, the attacker can only control relatively limited agents and is provided by a small-scale dataset for training. Recall that in algorithm 1 we use the same dataset for the malicious classification model and trigger generator training. We set the size of dataset for learning trigger generator to 1024 for all tasks in default. Even if the size of the dataset is only set to 32, FTA can provide a high attack performance (see figure 5.13).

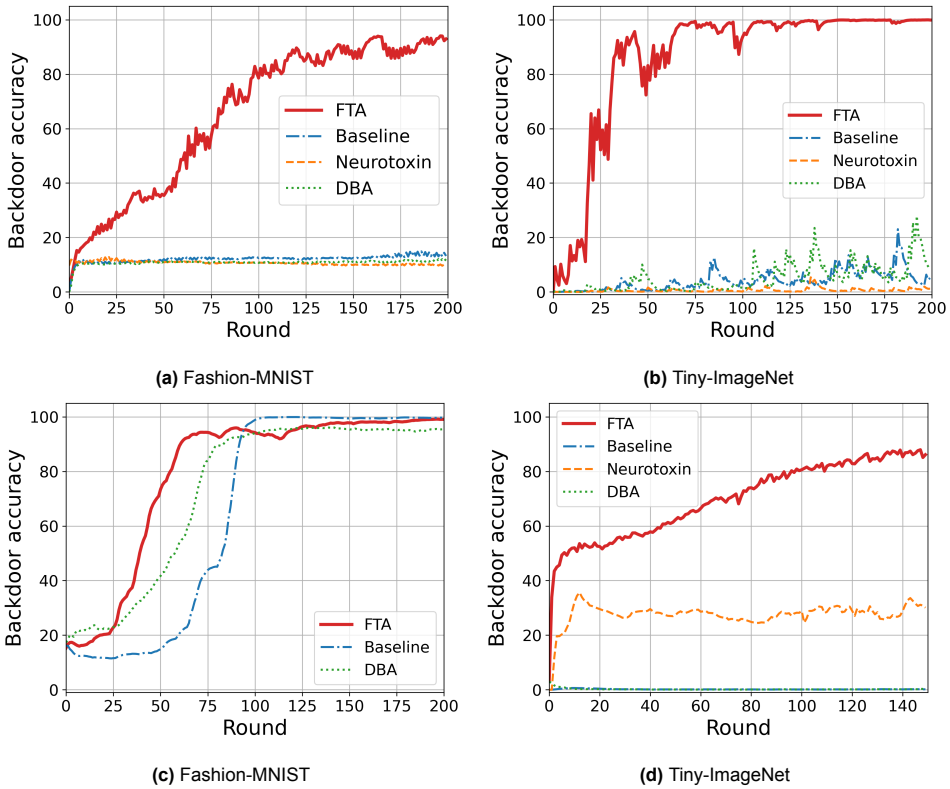
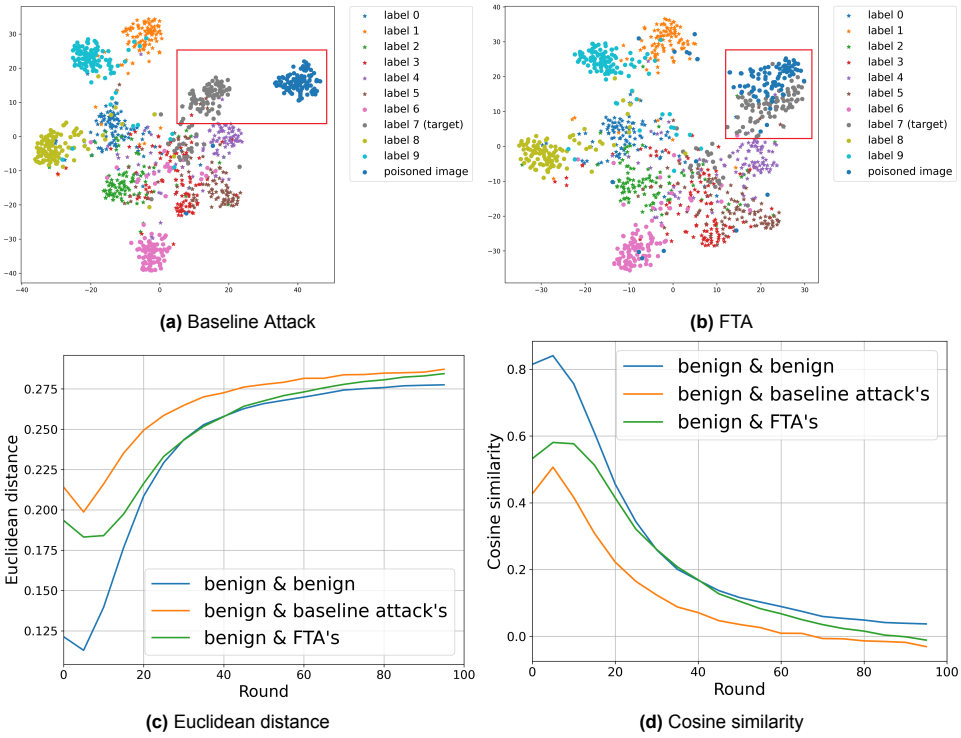


Figure 5.8: (a)-(b): The effectiveness of attack under SignSGD in Fashion-MNIST and Tiny-ImageNet. (c): The effectiveness of attack under Foolsgold in Fashion-MNIST. (d): The effectiveness of attack under SparseFed in Tiny-ImageNet.

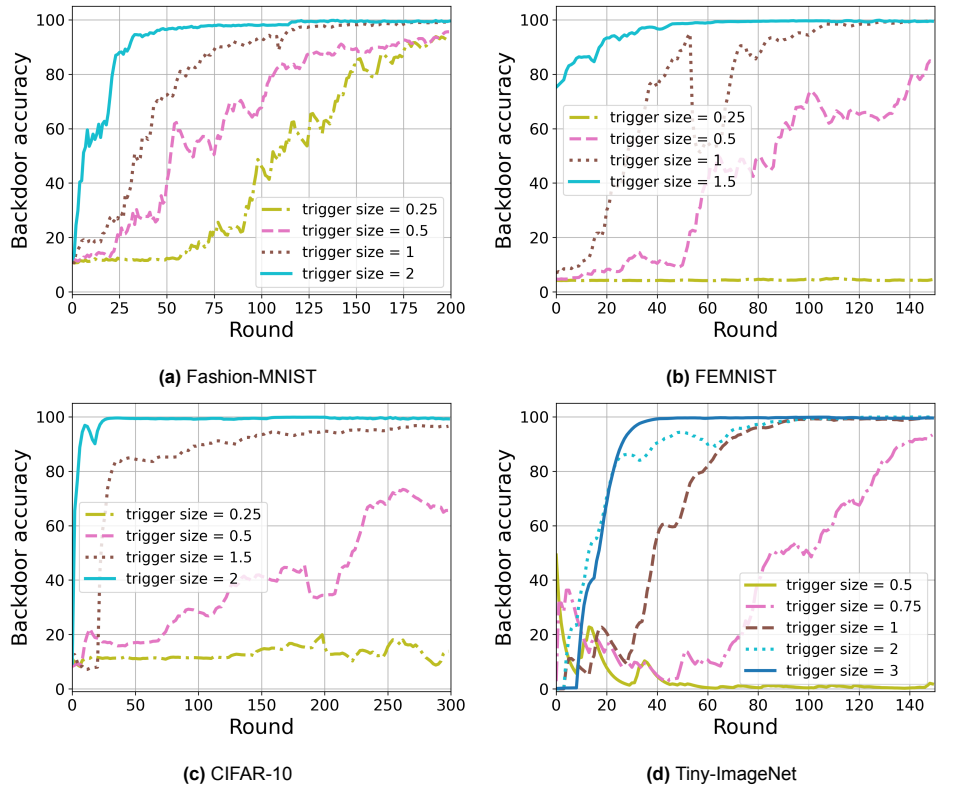
We note that the training process here is somewhat similar to generative adversarial networks, in which we do not require a large amount of samples in the training dataset.



(a) Baseline Attack **(b)** FTA

(c) Euclidean distance **(d)** Cosine similarity

Figure 5.9: (a)-(b): T-SNE visualization of hidden features of input samples in Fashion-MNIST. The hidden features between poisoned and benign samples of target label is indistinguishable in FTA framework. (c)-(d): Similarity comparison between benign & malicious updates. FTA's malicious updates is more similar to benign updates than the baseline attack's.



(a) Fashion-MNIST **(b)** FEMNIST

(c) CIFAR-10 **(d)** Tiny-ImageNet

Figure 5.10: Different trigger sizes on backdoor accuracy.

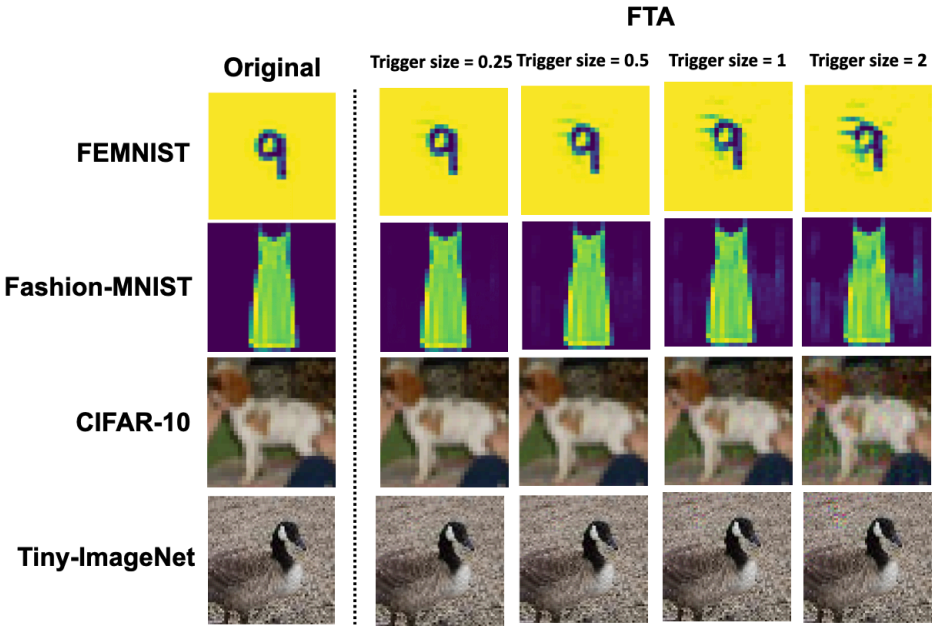
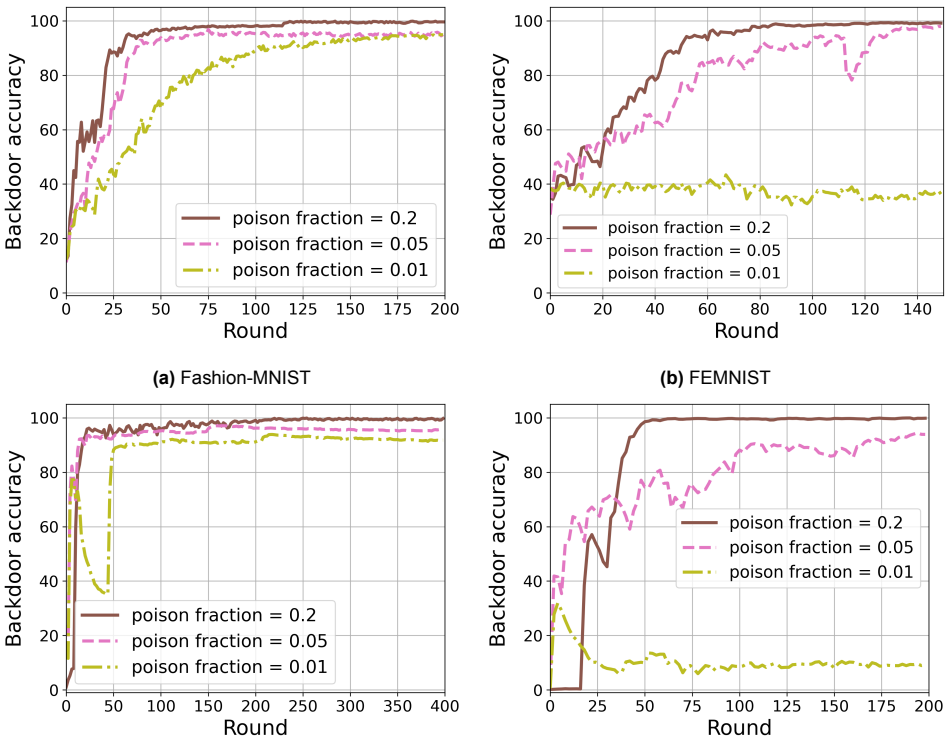


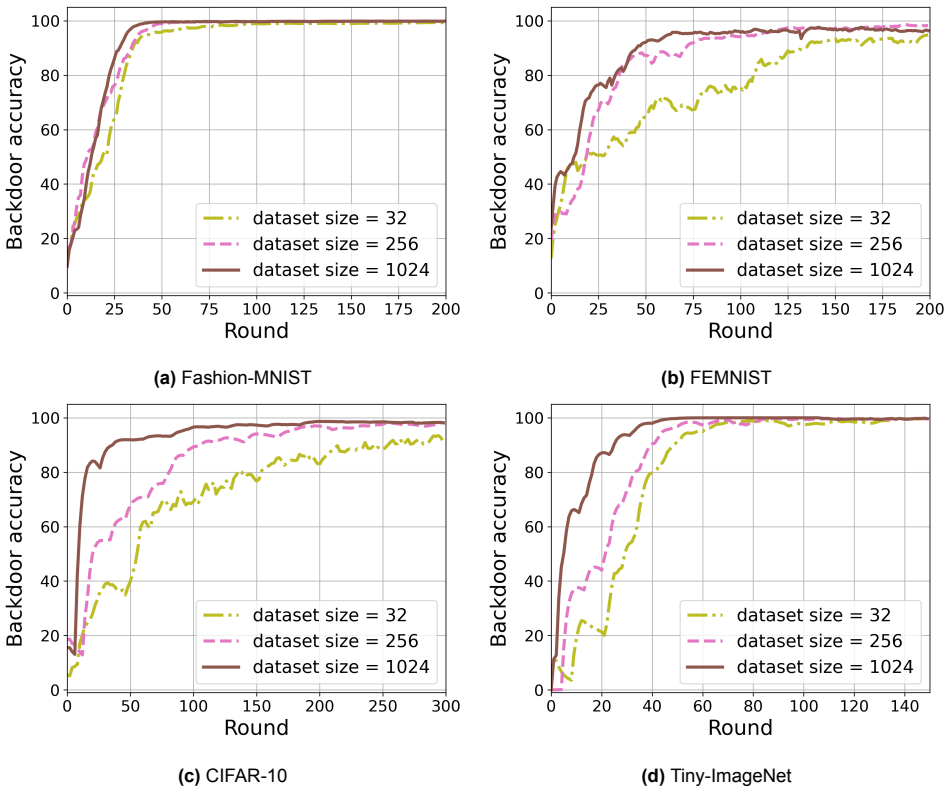
Figure 5.11: Visualization of backdoored images of different trigger sizes.



(a) Fashion-MNIST **(b)** FEMNIST

(c) CIFAR-10 **(d)** Tiny-ImageNet

Figure 5.12: Different poison fraction on backdoor accuracy.



(a) Fashion-MNIST **(b)** FEMNIST

(c) CIFAR-10 **(d)** Tiny-ImageNet

Figure 5.13: Different dataset size of trigger generator on backdoor accuracy.

6

DISCUSSION

In this work, we concentrate on the computer vision tasks, which have been the focus of numerous existing works [36, 34, 9, 42, 23]. In the future, we intend to expand the scope of this work by applying our design to other real-world applications, such as natural language processing (NLP) and reinforcement learning (RL), as well as other vision tasks, e.g., object detection.

The primary focus of FTA is to achieve stealthiness rather than durability, in contrast to other attacks such as Neurotoxin [40]. Neurotoxin manipulates malicious parameters based on gradients in magnitude, which yields a clear increase in the dissimilarity of parameters and thus harms the stealthiness of the attack. FTA addresses the dissimilarity difference of weights/biases introduced by backdoor training by using a stealthy and adaptive trigger generator□which makes the hidden features of poisoned samples similar to benign ones. We emphasize that the durability of backdoor attacks on FL is orthogonal to the main focus of this work, and we leave it as an open problem. A possible solution to achieve persistence could be to decelerate the learning rate of malicious agents, as proposed in [1].

7

CONCLUSION

We design an effective and stealthy backdoor attack against FL called FTA by learning an adaptive generator to produce imperceptible and flexible triggers, making poisoned samples have similar hidden features to benign samples with target label. FTA can provide stealthiness and robustness in making hidden features of poisoned samples consistent with benign samples of target label; reducing the abnormality of parameters during backdoor task training; manipulating triggers with imperceptible perturbation for training/testing stage; learning the adaptive trigger generator across different FL rounds to generate flexible triggers with best performance. The empirical experiments demonstrate that FTA can achieve a practical performance to evade SOTA FL defenses. We hope this work can inspire follow-up studies that provide more secure and robust FL aggregation algorithms.

References

[1] Eugene Bagdasaryan et al. “How to backdoor federated learning”. In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2020, pp. 2938–2948.

[2] Jeremy Bernstein et al. *signSGD with Majority Vote is Communication Efficient And Fault Tolerant*. 2019. arXiv: 1810.05291 [cs.DC].

[3] Arjun Nitin Bhagoji et al. “Analyzing federated learning through an adversarial lens”. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 634–643.

[4] Peva Blanchard et al. “Machine learning with adversaries: Byzantine tolerant gradient descent”. In: *Advances in Neural Information Processing Systems* 30 (2017).

[5] Keith Bonawitz et al. “Practical secure aggregation for federated learning on user-held data”. In: *arXiv preprint arXiv:1611.04482* (2016).

[6] Siyuan Cheng et al. “Deep feature space trojan attack of neural networks by controlled detoxification”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 2. 2021, pp. 1148–1156.

[7] Ittai Dayan et al. “Federated learning for predicting clinical outcomes in patients with COVID-19”. In: *Nature medicine* 27.10 (2021), pp. 1735–1743.

[8] Khoa Doan, Yingjie Lao, and Ping Li. “Backdoor attack with imperceptible input and latent modification”. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 18944–18957.

[9] Khoa Doan et al. “Lira: Learnable, imperceptible and robust backdoor attacks”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 11966–11976.

[10] Khoa D Doan, Yingjie Lao, and Ping Li. “Marksman Backdoor: Backdoor Attacks with Arbitrary Target Class”. In: *arXiv preprint arXiv:2210.09194* (2022).

[11] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[12] Matthew Jagielski et al. “Manipulating machine learning: Poisoning attacks and countermeasures for regression learning”. In: *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2018, pp. 19–35.

[13] Jing Jiang, Shaoxiong Ji, and Guodong Long. “Decentralized knowledge acquisition for mobile internet applications”. In: *World Wide Web* 23.5 (2020), pp. 2653–2669.

[14] Huseyin Kusetogullari et al. “ARDIS: a Swedish historical handwritten digit dataset”. In: *Neural Computing and Applications* 32.21 (2020), pp. 16505–16518.

[15] Tian Li et al. “Ditto: Fair and robust federated learning through personalization”. In: *International Conference on Machine Learning*. PMLR. 2021, pp. 6357–6368.

[16] Tian Li et al. “Federated optimization in heterogeneous networks”. In: *Proceedings of Machine learning and systems* 2 (2020), pp. 429–450.

[17] Yuezun Li et al. “Invisible backdoor attack with sample-specific triggers”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 16463–16472.

[18] Yang Liu et al. “A communication efficient collaborative learning framework for distributed features”. In: *arXiv preprint arXiv:1912.11187* (2019).

[19] Brendan McMahan et al. “Communication-efficient learning of deep networks from decentralized data”. In: *Artificial intelligence and statistics*. PMLR. 2017, pp. 1273–1282.

[20] Anh Nguyen et al. “Deep federated learning for autonomous driving”. In: *2022 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2022, pp. 1824–1830.

[21] Thien Duc Nguyen et al. “FLAME: Taming Backdoors in Federated Learning”. In: *31st USENIX Security Symposium (USENIX Security 22)*. Boston, MA: USENIX Association, Aug. 2022, pp. 1415–1432. ISBN: 978-1-939133-31-1.

[22] Tuan Anh Nguyen and Anh Tran. “Input-aware dynamic backdoor attack”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 3454–3464.

[23] Mustafa Safa Ozdayi, Murat Kantarcioglu, and Yulia R Gel. “Defending against backdoors in federated learning with robust learning rate”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 10. 2021, pp. 9268–9276.

[24] Ashwinee Panda et al. “SparseFed: Mitigating Model Poisoning Attacks in Federated Learning with Sparsification”. In: *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*. Ed. by Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera. Vol. 151. Proceedings of Machine Learning Research. PMLR, 28–30 Mar 2022, pp. 7587–7624.

[25] Adam Paszke et al. “Pytorch: An imperative style, high-performance deep learning library”. In: *Advances in neural information processing systems* 32 (2019).

[26] Matthias Paulik et al. “Federated Evaluation and Tuning for On-Device Personalization: System Design & Applications”. In: *CoRR* abs/2102.08503 (2021). arXiv: `2102.08503`. URL: `https://arxiv.org/abs/2102.08503`.

[27] Krishna Pillutla, Sham M. Kakade, and Zaid Harchaoui. *Robust Aggregation for Federated Learning*. 2022. arXiv: `1912.13445 [stat.ML]`.

[28] Phillip Rieger et al. “DeepSight: Mitigating backdoor attacks in federated learning through deep model inspection”. In: *NDSS*. 2022.

[29] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-net: Convolutional networks for biomedical image segmentation”. In: *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer. 2015, pp. 234–241.

[30] Virat Shejwalkar et al. “Back to the Drawing Board: A Critical Evaluation of Poisoning Attacks on Production Federated Learning”. In: *2022 IEEE Symposium on Security and Privacy (SP)* (2021), pp. 1354–1371.

[31] Karen Simonyan and Andrew Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: *International Conference on Learning Representations*. 2015.

[32] Ziteng Sun et al. “Can you really backdoor federated learning?” In: *arXiv preprint arXiv:1911.07963* (2019).

[33] Laurens Van der Maaten and Geoffrey Hinton. “Visualizing data using t-SNE.” In: *Journal of machine learning research* 9.11 (2008).

[34] Hongyi Wang et al. “Attack of the tails: Yes, you really can backdoor federated learning”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 16070–16084.

[35] Chulin Xie et al. “Crfl: Certifiably robust federated learning against backdoor attacks”. In: *International Conference on Machine Learning*. PMLR. 2021, pp. 11372–11382.

[36] Chulin Xie et al. “Dba: Distributed backdoor attacks against federated learning”. In: *International Conference on Learning Representations*. 2019.

[37] Timothy Yang et al. *Applied Federated Learning: Improving Google Keyboard Query Suggestions*. 2018. arXiv: `1812.02903 [cs.LG]`.

[38] Dong Yin et al. “Byzantine-robust distributed learning: Towards optimal statistical rates”. In: *International Conference on Machine Learning*. PMLR. 2018, pp. 5650–5659.

[39] Tao Yu, Eugene Bagdasaryan, and Vitaly Shmatikov. “Salvaging federated learning by local adaptation”. In: *arXiv preprint arXiv:2002.04758* (2020).

[40] Zhengming Zhang et al. “Neurotoxin: Durable Backdoors in Federated Learning”. In: *Proceedings of the 39th International Conference on Machine Learning*. Vol. 162. Proceedings of Machine Learning Research. PMLR, 17–23 Jul 2022, pp. 26429–26446.

[41] Bo Zhao et al. “Fedinv: Byzantine-robust federated learning by inversing local model updates”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 36. 8. 2022, pp. 9171–9179.

[42] Zhendong Zhao et al. “DEFEAT: Deep Hidden Feature Backdoor Attacks by Imperceptible Perturbation and Latent Representation Constraints”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 15213–15222.