Fabricated Flips

Poisoning Federated Learning without Data

Huang, Jiyue; Zhao, Zilong; Chen, Lydia Y.; Roos, Stefanie

DOI
10.1109/DSN58367.2023.00036

Publication date
2023

Document Version
Final published version

Published in
Proceedings of the 2023 53rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2023

2023 53rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN) | 979-8-3503-4793-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/DSN58367.2023.00036

Fabricated Flips: Poisoning Federated Learning without Data

Jiyue Huang, Zilong Zhao, Lydia Y. Chen, Stefanie Roos
TU Delft, The Netherlands. {J.Huang-4, Z.Zhao-8, Y.Chen-10, S.Roos}@tudelft.nl

***Abstract*—Attacks on Federated Learning (FL) can severely reduce the quality of the generated models and limit the usefulness of this emerging learning paradigm that enables on-premise decentralized learning. However, existing untargeted attacks are not practical for many scenarios as they assume that *i)* the attacker knows every update of benign clients, or *ii)* the attacker has a large dataset to locally train updates imitating benign parties.**

In this paper, we propose a data-free untargeted attack (DFA) that synthesizes malicious data to craft adversarial models without eavesdropping on the transmission of benign clients at all or requiring a large quantity of task-specific training data. We design two variants of DFA, namely DFA-R and DFA-G, which differ in how they trade off stealthiness and effectiveness. Specifically, DFA-R iteratively optimizes a malicious data layer to minimize the prediction confidence of all outputs of the global model, whereas DFA-G interactively trains a malicious data generator network by steering the output of the global model toward a particular class. Experimental results on Fashion-MNIST, Cifar-10, and SVHN show that DFA, despite requiring fewer assumptions than existing attacks, achieves similar or even higher attack success rate than state-of-the-art untargeted attacks against various state-of-the-art defense mechanisms. Concretely, they can evade all considered defense mechanisms in at least 50% of the cases for CIFAR-10 and often reduce the accuracy by more than a factor of 2.

Consequently, we design REFD, a defense specifically crafted to protect against data-free attacks. REFD leverages a reference dataset to detect updates that are biased or have a low confidence. It greatly improves upon existing defenses by filtering out the malicious updates and achieves high global model accuracy.

***Index Terms*—Federated learning, data-free attack, untargeted attack, data heterogeneity**

I. INTRODUCTION

Federated learning (FL) [29, 46] enables distributed training of machine learning models, e.g., multi-class image classifiers, without sharing the raw data. *Clients* train models locally and the overall model, called the global model, is an aggregation of these local models. The training proceeds in multiple rounds: in each round, *the central server* provides a global model that clients use to initialize their local models. They then train on their local dataset and provide updates to the central server, who aggregates these updates to a new global model for the next round. In this manner, models requiring personal data such as information about medical or financial conditions can be obtained without explicit privacy violations. Recently, FL has been applied to domains such as detection of credit card fraud [57, 58], cybersecurity center operations [16], and medical relation extraction [38].

A downside of preventing the central server from accessing local data is that it limits the ability to detect misbehavior. Adversarial clients may reduce the quality of the model by manipulating the data they train on [41] or their local model directly [10]. Cross-device [13, 15] FL, which allows arbitrary parties to join the distributed training, is especially vulnerable as attackers can easily infiltrate the system. The attack can be untargeted, i.e., aiming for an overall accuracy degradation of the trained global model. It can also be targeted, i.e., only supposed to affect certain input, e.g., inject backdoors that lead to wrong model output from input data with a certain chosen feature [2]. In this paper, we focus on untargeted attacks, as they are far-reaching denial-of-service attacks. In cross-device FL, attackers may run such a denial-of-service attack to undermine a competing company from getting meaningful models after their users. Furthermore, when machine learning as a service [30] is extended to include FL [17], untargeted attacks aiming to cause losses for a service provider are to be expected, similar to current denial-of-service attacks on Amazon Web Services and Github [1].

There have been a number of untargeted attacks on FL [4, 10, 33]. Yet, some attacks [4, 33] assume that the adversary is aware of all of the updates that benign clients send. It is unclear how they can practically obtain such knowledge as clients only share the updates with the benign central server and communication can be encrypted to prevent eavesdropping from the adversary. While not all attacks require benign updates, attacks that can succeed without this knowledge requires that the attacker has a considerable amount of training data to train substitute benign updates [10]. Although this assumption is realistic for common tasks, e.g., image classification of common pets, the possession of such data is much less likely for special-purpose tasks, e.g., classification of rare disease based on detailed medical data [31].

In this paper, we consider whether it is actually necessary to have real data (or benign updates). One may expect that in cross-device FL, it is relatively easy to obtain data as everyone can join, which might indicate that everyone can have data. However, there exist scenarios where admission is not restricted to a predefined group because there are few parties that can contribute and it is not known who they are. For instance, for a study on the live of people with a rare disease, it might not be possible to access medical records on who has the disease, so it makes sense to just publicly ask for participation. Furthermore, not requiring parties to identify before joining allows them to participate anonymously, possibly using tools like Tor [9] to send in their updates without having to fear

[1] https://www.a10networks.com/blog/aws-hit-by-largest-reported-ddos-attack-of-2-3-tbps/

2158-3927/23/$31.00 ©2023 IEEE
DOI 10.1109/DSN58367.2023.00036

that they reveal that they have a certain medical condition, which could increase their insurance premium or prevent them from gaining employment. In such a scenario, it is also hard to corrupt participating clients and use their data, as the identity of the clients is not known. Even if the learning task is such that is easy to obtain data, e.g., a software company aiming to build a model on how users interact with their tool, it is still additional overhead for the attackers, e.g., they have to either use the tool themselves or obtain data from a real user. Thus, even if the attacker can get data, the question of whether they *have to* or can skip the overhead of data acquisition is essential as without data acquisition, it is more likely that attacks can be automated and run at scale against many FL learning tasks.

We design a novel Data-Free Attack (DFA) and evaluate it on the example of image classification. The goal of the attack is to reduce the overall accuracy of the model through the injection of malicious model updates based on synthetic images. In each round, the attacker first generates malicious images by making use of the received global model and then trains the local adversarial model using those images paired with a randomly chosen class $\tilde{Y}$. We design two variants of DFA, DFA-R and DFA-G, which steer the global model to classify images to either have low confidence or to classify incorrectly. Our first attack variant, DFA-R, generates synthetic local data by adding a filte**R** layer to the training. This data generation optimizes towards local synthetic data that is ambiguous according to the current global model, i.e., the current global model should output each of the L possible classes with equal probability. A local model corresponding to such data diverts the global model and reduces classification accuracy. In contrast, our second attack, DFA-G, iteratively trains a **G**enerator that should produce synthetic images that are not from a specific randomly chosen class $\tilde{Y}$. We then assign these images with class label $\tilde{Y}$ and train on the resulting dataset, thus implicitly combining synthetic data generation with label flipping for poisoning.

To improve stealthiness for both attacks, we add a regularization term to the loss function of the classifier that steers the update generation such that updates are not detected as outliers and hence not removed by defenses. DFA thus stealthily bypasses the defense by ensuring that the deviation to the global model follows similar patterns as benign updates.

In our evaluation, we determine the attack success rate, i.e., the decrease in model accuracy caused by the attack, and the rate at which our attackers pass the defense. We evaluate different levels of data heterogeneity by assigning data to clients according to the Dirichlet distribution, which is a common model for heterogeneous real-world distributions [43]. DFA-R and DFA-G reduce the accuracy of the trained model by a factor of 2 for most settings, even if defenses are applied. In comparison to state-of-the-art attacks, DFA-R and DFA-G achieve similar results, despite having weaker assumptions. Indeed, for most scenarios, our attacks perform slightly better than the existing attacks.

Having shown that data-free attacks have severe impact on the accuracy of FL, we propose a defense strategy, REFD, which aims to defend against DFA-G and DFA-R by leveraging a reference dataset at the server. Based on this reference dataset, the central server determines whether a received model update is biased toward a certain class, which is typical for DFA-G, or shows a low confidence, which is typical for DFA-R. It combines these two factors into a novel defense score, termed $\mathcal{D}$-score. Our evaluation results show that REFD successfully defends against the proposed data-free attacks, achieving accuracies that are close to the accuracy achieved in the absence of both attacks and defenses.

II. BACKGROUND AND RELATED WORK

A. Federated Learning Primer

As a distributed machine learning framework, FL systems consist of a set of N clients and a central server. The global training process considers R consecutive rounds indexed by the round number t. After model initialization by the server, each client i (for $i = 1, 2, ..., N$) trains a local model based on their own real data without sharing the raw data. The server iteratively aggregates models/gradients submitted from clients and distributes the aggregated model to the clients until reaching global model convergence. As clients can be offline or unresponsive, only a subset of them usually submits updates.

In this paper, we focus on image classification tasks with L classes. Let D_i be the local dataset of client i and F be the objective function for the classification task. The client i updates its local model weights based on the global model $\boldsymbol{w}(t)$ by:

$$\boldsymbol{w}_i(t+1) = \boldsymbol{w}(t) - \eta \frac{\partial F(\boldsymbol{w}(t), D_i)}{\partial \boldsymbol{w}(t)}, \tag{1}$$

where η is the global uniformed learning rate.

For aggregating models of $K \leq N$ clients, the predominant method for attack-free scenarios is FedAvg [24], which aggregates the new global model as a weighted average of the submitted local models , i.e.,

$$\boldsymbol{w}(t) = \sum_{i=1}^{K} \frac{n_i}{\sum_{k=1}^{K} n_k} \boldsymbol{w}_i(t), \tag{2}$$

where n_i is the number of training samples of client i. However, the above algorithm is not robust under attacks [2, 10, 33, 45], hence defenses for securing the aggregation against maliciously crafted updates (also called robust aggregation methods) have been developed.

B. Existing attacks in FL systems

FL empowers clients by leaving the training to them and not revealing the local data. However, as a consequence, FL systems are vulnerable to malicious behaviors. Attacks can happen during the **training time** [2, 4, 10, 33, 45] or **inference time** [27, 35, 48]. For the inference-time attacks, attackers aim to infer private data [27]. They may even reconstruct the private local training data [48]. In this paper, we focus on training-time attacks where attackers participate in the training. We classify the training-time attacks from two perspectives: *i*) the attack

Attacks	LIE[4]	Fang[10]	Min-Max[33]	Min-Sum[33]	DFA (ours)
No benign updates needed	✗	✓/✗	✓/✗	✓/✗	✓
Defense-agnostic	✓	✗	✓	✓	✓
No raw data needed	✓	✓/✗	✓/✗	✓/✗	✓
Heterogeneity considered	✗	✓	✓	✓	✓
Attack type	Statistic	Statistic	Statistic	Statistic	Optiminzation

TABLE I: Attack scenarios in the state-of-the-art and ours.

objectives and *ii)* the attacked component of the FL system, e.g., data or model.

There are three attack objectives for training-time attacks: **Free-riding** [11, 22] is used to obtain the global model without contributing data and computation. **Targeted attacks** [2, 45] aim to decrease the model accuracy for specific data, e.g., data with designed triggers. **Untargeted attacks** [4, 10, 33], in contrast, aim to decrease the general accuracy of the model.

There are four state-of-the-art untargeted attacks, namely LIE [4], Fang [10] as well as Min-Max and Min-Sum [33], which are two variants of the same attack idea. We summarize their key differences in Table I. All attacks require knowledge of the models of benign clients, real data, or knowledge of any defenses applied by the server. Some of them, like Min-Max, are flexible in that they can work with either benign updates or real data but they need at least one of the two, which we indicate by ✓/✗ in the respective rows in the table. In terms of methods, all existing attacks rely on statistical methods or heuristics to construct the malicious updates by shifting the mean of benign updates without being detected. Concretely, LIE [4] calculates the mean and standard deviation of all of the benign updates and then shifts the true mean by changing the value in one direction in such a manner that it is within the range that is considered acceptable by the defense. Shejwalkar et. al [33] further improve LIE by adapting the scaling factor z of the weighted sum as well as extending the standard deviation to the sign and unit vector of the gradient. By such means, the maximum distance (or the sum of squared distances for Min-Sum) of the malicious gradient from all the benign gradients is upper bounded. Note that while the authors [33] propose a number of attacks according to different levels of adversarial knowledge, we only compare to the Min-Max attack, which is the strongest in their paper. Fang et. al [10] propose an attack that steers global model parameters in the opposite direction of the benign updates and ensures its stealthiness through the knowledge of the exact defense. Aforementioned untargeted attacks, except LIE, are evaluated in junction with the heterogeneous data, which is attribute skewed [8, 37] or label skewed [23, 42, 49].

Moreover, attacks can also be categorized by the component which attacks act upon: data or model. During the training time, the adversary may inject malicious data with dirty labels or data to train the local model, e.g., label flipping [41] and trigger injection [2, 45]. For example, backdooring[2] is executed by injecting trigger-based malicious samples [2, 50] into the local training dataset. DBA [45] then extends the study [2] to bypass Sybil defenses such as FoolsGold [12]. Modeling poisoning [2, 4, 10, 33, 45] manipulates the submitted model rather than merely adopting malicious data to train, e.g., submit updates of the reversed sign of training gradient [10]. Generally, model poisoning attacks require sophisticated technical capabilities such as eavesdropping and sufficient computation resources.

None of the existing attacks can deal with an attacker that does not have data unless they can observe the communication in plaintext. Our attack uses a generator, as do other attacks but for highly different scenarios or goals: attacking centralized learning [51, 54], attacking privacy [36, 55], or mimicking prototypical samples of the other participants' training set with the goal of targeted attacks [52, 53].

C. Existing Defense Mechanism in FL

We here focus on defense mechanisms for FL, as algorithms designed for centralized learning (e.g., [6, 20] are not directly applicable in FL. To tackle the attacks on FL systems, existing defense strategies can be conducted either on the server-side [5, 25, 47], or the client-side [26, 39, 56]. Server-side defenses are effective against both targeted and untargeted attacks due to the access to all model updates, whereas the state-of-the-art client-side defenses are merely shown to be effective against targeted attacks. As we are concerned with untargeted attacks, we hence focus on server-side defenses.

Generally, there are three categories of server-side defenses: *i) Sybil defenses* aim at detecting Sybil attackers who are controlled by one entity and submit similar updates. For example, *FoolsGold* [12] identifies Sybils based on the diversity of client contributions using cosine similarity of client updates. *ii) Statistic defenses* curate the aggregated model by computing the statistics of every parameter across multiple updates. *Median* [47] utilizes the median value of all updates for each parameter whereas Trimmed mean (*TRmean*) [47] excludes the minimum and maximum value from the average of each parameter. *iii) Outlier detection* [5, 25] removes updates based on the pairwise distances of returned models. Higher distance implies that data owned by a client is of low quality or unrelated to the training task. *Krum* [5] only uses one update sent from the client whose cumulative distance of updates to the other updates is the lowest, taking the squared L2 Norm as a metric. *mKrum* [5] extends this idea by choosing multiple updates. *Bulyan* [25] first selects updates using *mKrum* and further computes the trimmed mean of the selected gradients.

III. An Optimization-based Data-free Attack

In this section, we first introduce the threat model for our work. Then we propose our data-free attack (DFA) with two variants to generate malicious data inputs and local model updates: DFA-R and DFA-G.

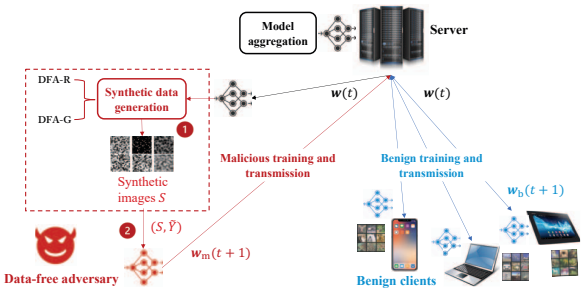


Fig. 1: The framework of our proposed data-free attack (DFA) without knowing benign updates and owning raw data.

A. Threat Model

We assume that communication between clients and the central server uses encrypted and authenticated channels, which prevent eavesdropping and manipulation of data during transmission. As a consequence, attackers are unaware of benign client updates. Benign clients always follow the protocol whereas malicious clients may arbitrarily deviate. All attackers may submit the same update. We add these assumptions for simplicity as we can easily circumvent Sybil defenses by adding small perturbation noise, as shown in the related work [2]. The central server applies a defense mechanism, which is not known to the clients.

We focus on cross-device FL, which means that anyone can join and at the same time there is client selection each round. Furthermore, the adversary inserts their own clients in the system rather than corrupt other clients. Corrupting other clients requires knowing the identities of other clients, which is not explicitly shared in cross-device FL. In the absence of anonymous communication, the adversary could obtain the identities only from observing network traffic but the ability to observe network traffic is restricted to internet service providers and other parties, so we do think it is more realistic to assume that the adversary does not know the other clients and hence also cannot easily corrupt them.

Additionally, we assume that malicious parties do not have any data so as to enhance the versatility of the adversary. In practice, the difficulty of obtaining data varies between tasks. It is reasonable to assume that there are tasks relying on rare data that an attacker cannot easily obtain. We assume that all computations are executed by one adversarial party, who then sends the updates to individual malicious clients.

Objectives: The overall objective of an untargeted attack in Federated Learning is reducing the accuracy of the global model maintained by the central server. As a part of achieving this objective, clients need to craft malicious updates that bypass the applied defense.

Capabilities: First, we assume that the number of malicious users controlled by the adversary in the system does not exceed 50% of the total clients. It seems implausible that a defense can overcome a higher number of attackers as defenses typically need a reference for benign behavior. The attacker cannot break cryptographic primitives. More generally, it is computationally bounded so that it cannot solve NP-complete or NP-hard problems. Otherwise, they can arbitrarily control the communication and computation of the malicious clients but not of any other parties in the system.

Knowledge: Neither the defense algorithm nor benign updates are known to the adversary. As the attacker also does not have data, the only knowledge of the adversary is the classification task in general, i.e., the number of classes, which is necessarily accessible as the server distributes the model.

B. Attack Optimization Framework

The overall framework of our proposed attack DFA is illustrated in Fig. 1. The server first distributes the current global model (classifier) $\boldsymbol{w}(t)$ to all of the clients. The benign clients truthfully follow the protocol and send the trained model $\boldsymbol{w}_b(t+1)$ back to the server. Malicious clients send the adversarial model $\boldsymbol{w}_m(t+1)$ instead. Then the server aggregates the submitted updates according to the deployed defense. As attackers do not have real data or benign updates, intuitively, the most obvious approach to attack is to directly change $\boldsymbol{w}(t)$.

We experimented with using random weights but the attack was detected almost always. Concretely, only 2.62% and 6.57% of all updates submitted by malicious clients with random model weights bypassed the *mKrum* defense for Fashion-MNIST and Cifar-10, respectively. For the *Bulyan* defense, the attack only bypassed the defense in 3.27% of the cases for Fashion-MNIST and always failed for Cifar-10. As manipulating the model directly does not seem a promising approach, we optimize the generation of synthetic malicious images according to $\boldsymbol{w}(t)$ and then use it to train the local adversarial model every round. The attack process consists of the following two steps.

1. Malicious image generation. We propose two optimization methods to synthesize malicious images based on different optimization methods, objectives of adversarial models and, importantly, the feedback of the global model. The first method is **DFA-R**, which introduces an additional **filter input layer**[2] and optimizes it from a dummy image with the objective to reduce the confidence on all outputs of the global model. Our second method, **DFA-G**, designs a **generator network** to synthesize malicious images such that the output of the global model biases toward a randomly chosen class. As such, the generated noisy images paired with incorrect labels are applied to malevolently update the current model. The details on DFA-R and DFA-G are discussed in the following subsections.

2. Adversarial classifier training with distance-based loss. In this step, the attacker uses synthetic data as generated by step 1 to train the classifier $\boldsymbol{w}_m(t+1)$. The optimization problem of the attack then becomes $\min_{\boldsymbol{w}_i} F(\boldsymbol{w}_i, S)$, where S is the generated image set. In order to enhance the stealthiness and hence pass the unknown defense, we propose to train with a distance-based loss function $\min_{\boldsymbol{w}_i}(F(\boldsymbol{w}_i, S) + \mathcal{L}_d)$ with regularization term $\mathcal{L}_d$ to enhance stealthiness (detailed

[2]Such a layer has the same input dimension as the original image.

illustration in Sec. III-E). The size of S, $|S|$, is a hyper parameter of our attack framework that depends on the task. In the evaluation, we find that using a similar number of images as benign clients results in an effective attack. The adversary can estimate the size during training based on the aggregated results of the global model and the duration that other clients require for training.

C. DFA-R *synthetic data generation*

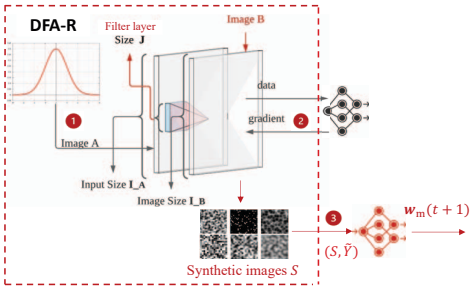


Fig. 2: Synthetic data generation process of DFA-R.

When constructing the synthetic dataset S, DFA-R aims to aggressively lower the confidence of all outputs of the global model by introducing a malicious filter layer(i.e., a convolutional layer). DFA-R optimizes this image layer such that per-class probability output of global model is equally low, i.e., $Y_D = [\frac{1}{L}, \frac{1}{L}, ..., \frac{1}{L}]$, where L is the total number of classes. Such data is bound to confuse the global model. Fig. 2 depicts the optimization procedure to find $|S|$ malicious images iteratively via two steps: *i)* generating the malicious image through mapping a random dummy image via a filter layer [19], and *ii)* optimizing the filter layer by minimizing cross-entropy loss of the global model between the predicted class probabilities and Y_D.

Concretely, we first generate a random image A (size $a \times a$), with each pixel being drawn from a uniform distribution, and apply the filter layer to transform it into an image B(size $b \times b$). In this manner, we train a mapping from randomness to images that have the desired properties. The size of image B is the same as the real image. We let this convolutional layer have kernel size $J \times J$, i.e., the square filter layer between image A and B in Fig. 2. After being filtered from the convolution layer, the image B is then classified by the current global model. The attack works for various network structures and datasets, e.g., Alexnet, VGG on other image datasets, as long as the relation between input and output are maintained. Concretely, for stride size St and padding size P [21], we require $a = b \times (St+1) - 2P + J$. The attack can be extended to other tasks, e.g., text processing, by replacing the filter model and using a Seq2Seq model [40] instead of a convolutional layer. In this manner, the random text (mapped from random values to a dictionary) is filtered by the Seq2Seq model and fed in the text processing network, similar to Fig. 2.

To optimize the convolutional layer that results in ambiguous Y_D, we first consider the dummy image A, the filter layer, synthetic image B, and the global model as one big classification problem. Its training objective is to minimize the cross-entropy loss of predicted probabilities of image B and Y_D, such that the model cannot predict classes reliably. Different from the regular training of a classification problem, we keep certain parts of the model and input constant. Specifically, the model weights of the global model and the image A are static. Otherwise, without keeping A static, we would need to re-train whenever we change the randomness. Keeping the number of trainable parameters to a minimum, we optimize the efficiency of the attack. The only trainable parameters here are the parameters of the filter layer. It takes E epochs to train this convolutional layer. Upon finishing training, the image B is one data instance of S. To increase the diversity of the training dataset S, for each FL training round, we repeat the above process for $|S|$ times to construct S.

D. DFA-G *synthetic data generation*

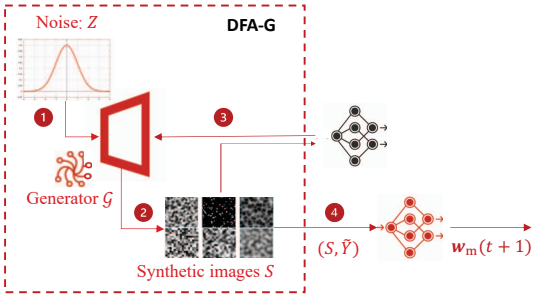


Fig. 3: Synthetic data generation process of DFA-G.

In contrast to DFA-R, DFA-G synthesizes images through a generator network, which misleads the global classifier to confidently make incorrect classifications. In order to do so, we generate images that are not supposed to be from a class $\tilde{Y}$ but classify all of them as $\tilde{Y}$, which is a randomly chosen label and never changes through the training procedure. The training/optimization of the generator network is through the feedback of the global model, i.e., we assume that the classification provided by the global model is the correct classification of the synthetic image. Typically, benign training minimizes the cross-entropy of the prediction and true label so that the model can output an accurate prediction. However, as our goal is to reduce the model accuracy, we maximize the cross-entropy of the prediction and $\tilde{Y}$ to train $\mathcal{G}$, steering the generated images away from $\tilde{Y}$. As DFA-R, DFA-G works for various network structures, with input and output size needing to match the training data.

The training procedure for the generator is shown in Fig. 3. We first draw a random noise vector Z from the Gaussian distribution and input Z into a generator $\mathcal{G}$ to synthesize malicious images. We use the same random seed over multiple rounds so that the trained generator is able to consistently produce synthetic data different from class $\tilde{Y}$, as our training goal is to optimize the mapping from the generated vector to the targeted synthetic data. The network structure of the generator is a transpose convolution neural network (TCNN),

which outputs task-specific image size data $S = \mathcal{G}(Z)$. The size of real data can be obtained from $\boldsymbol{w}(t)$. Specifically, we use a lightweight TCNN of two transposed convolutional layers and one convolutional layer following the structure of the popular WGAN paper [1]. The model parameter of the generator $\mathcal{G}$ is randomly initialized before training, denoted as $\boldsymbol{\theta}$. As the generator aims at synthesizing images that differ from the chosen class $\tilde{Y}$, the objective function of $\mathcal{G}$ is $\max_{\boldsymbol{\theta}} F(\boldsymbol{w}(t), (S, \tilde{Y}))$ where S is generated from $\boldsymbol{\theta}$. After training $\mathcal{G}$ locally for E epochs until convergence, the synthetic images are leveraged to train $\boldsymbol{w}_m(t+1)$. When it comes to other tasks, e.g., text processing, the generator is a recurrent neural network such as GRU [7] in order to generate random texts rather than just random numbers.

In summary, differences between the two variants are *i)* the optimization network, *ii)* the objective functions, *iii)* the use of $\tilde{Y}$, and *iv)* the randomness in their inputs.

E. Distance-based Regularization

State-of-the-art defenses in a FL system are mainly based on the pairwise distances among multiple updates. In order to bypass the defense mechanisms, we introduce a distance-based regularization term when training the adversarial classifier with the aim to further enhance stealthiness of the adversarial model updates. The concrete regularization term is

$$\mathcal{L}_d = \|\boldsymbol{w} - \boldsymbol{w}(t)\|_2 - \|\boldsymbol{w}(t) - \boldsymbol{w}(t-1)\|_2. \tag{3}$$

In Eq. 3, the first term refers to the weight differences of the adversarial update and the current model. Analogously, the second term refers to the difference between the current model and the model of the previous round. This term varies over rounds and the optimization variable is the model parameter $\boldsymbol{w}$. We add $\mathcal{L}_d$ to the vanilla cross-entropy loss in the objective function of the adversarial classifier to avoid extremely high differences in model changes over rounds, which could be easily detected. Thus, in both DFA-R and DFA-G, we guide the training such that the differences in weights are similar to the ones in previous rounds, as achieved by using the two most recent global models.

IV. Experimental Evaluation

We empirically evaluate the effectiveness of our proposed data-free untargeted attack, on three commonly used classification benchmarks. We compare the attack success rate and defense pass rate for various settings, for four state-of-the-art defenses and in comparison to three existing attacks. All of the results reported in this section are averaged over three runs. The source code of DFA is provided in github[3]

A. Experiment Setup

FL system. Our FL system considered contains 100 clients. In typical real-world FL systems, some of the clients could be offline or unavailable temporarily and hence not all of them might be able to participate in the whole training process. Thus, as in previous work [2, 24, 33], 10 of the available clients are selected uniformly at random each round. Clients train the classifier locally for one epoch. For the main results, we assume that the adversary can compromise 20% of the clients following [10, 33], unless stated otherwise, and further evaluate 10% and 30% in Sec. IV-E. Lower percentages of attackers have been shown to be ineffective [34].

Datasets and networks. In this work, we consider three datasets. *Fashion-MNIST* [44] consists of a training set of 60,000 and a test set of 10,000 fashion-related images. Each instance is a 28 $\times$ 28 grayscale image. *Cifar-10* [18] contains 50,000 training images of 3-channel RGB images and 10,000 of test images. *SVHN* [28] includes 73257 digit images for training and 26032 for testing. All digits have 32 $\times$ 32 pixels. All datasets have 10 classes in total. For Fashion-MNIST and Cifar-10, the images are evenly distributed over classes. SVHN in slightly imbalanced in class distribution. The total number of images used to train in this paper is reduced to 10% for Fashion-MNIST and Cifar-10 but maintain the original size for SVHN. For Fashion-MNIST and Cifar-10, the data are chosen uniformly at random in order to model real-world scenarios that full data may not be available during the whole training. This amount is verified to be sufficient for training on Cifar-10 and Fashion-MNIST [3].To determine the $|S|$ and show hyperparameter sensitivity, we run initial experiments varying $|S|$ from 20, 50, and 100 based on knowing 50 samples per client for Cifar-10. We found that DFA is able to achieve similar attack success rate. Indeed, sometimes a lower $|S|$ had a higher ASR, e.g., for DFA-G on Fashion-MNIST with $\beta = 0.5$, $|S| = 20$ has higher attack success rate than $|S| = 50$. As they all succeed in attacking, we use the results of 50 in the paper to keep consistency. For these three datasets, we use representative neural networks with 2 (for Fashion-MNIST) and 6 (Cifar-10 and SVHN) convolutional layers connected with 1 and 2 densely-connected layers, respectively, to map the inputs and outputs[4].

Defense mechanisms. Four state-of-the-art defenses are evaluated in our work: *mKrum*, *TRmean*, *Bulyan* and *Median*. We do not apply *Krum* since *mKrum* interpolates between *Krum* and averaging, thereby allowing the trade-off between the resilience properties and the convergence speed [5].

Data heterogeneity. To emulate a heterogeneous distribution, we assign data to clients according to the commonly used Dirichlet distribution. It emulates a real-world data distribution and the degree of heterogeneity is governed by the hyper-parameter β [43], indicating the level of heterogeneity. In Sec. IV-D, we vary β from 0.1 to 0.9 in order to demonstrate our effectiveness for different degrees of data heterogeneity. Higher β means a lower degree of data heterogeneity. For our experiments, except for Sec. IV-D, we choose $\beta = 0.5$, as in the prior work [14, 43].

Hardware. Our FL emulator is based on Pytorch and we run experiments on a machine running Ubuntu 20.04, with 32

[3]https://github.com/GillHuang-Xtler/DSN2023DFA. For any question about code, please contact {J.Huang-4, Z.Zhao-8}@tudelft.nl.

[4]We use shallow networks to simplify evaluation, consistent with [33], higher accuracy can be achieved with deep nets.

GB memory, a GeForce RTX 2080 Ti GPU and an Intel i9 CPUs with 10 cores (2 threads each).

B. Evaluation Metrics.

We utilize two main metrics to evaluate the effectiveness of our attack. *i) Attack success rate (ASR)* is defined by:

$$ASR = \frac{acc - acc_m}{acc} \times 100\%, \tag{4}$$

i.e., the decrease of accuracy caused by attacks. Specifically, it is the difference between the global accuracy acc without attacks and defenses and the maximum accuracy acc_m of the global model during one experiment with attacks. Attack success rate specifies the effectiveness of an attack strategy through the decrease in accuracy. The Higher, the better.

ii) Defense pass rate (DPR) is a metric to measure the stealthiness of an attack. In our paper, it is defined by the proportion of attackers who have passed the defense (N_p) from all of the randomly selected attackers (N_s):

$$DPR = \frac{N_p}{N_s} \times 100\%. \tag{5}$$

DPR as defined above requires that defenses select updates for aggregation rather than computes statistics on all updates. Thus, as detailed Sec. II-C, DPR can only be computed for *mKrum* and *Bulyan*, but not for *TRmean* and *Median*. High *DPR* is better.

Baselines. We are the first to propose data-free untargeted attacks. So there is no direct baseline to compare to. To demonstrate the effectiveness, we compare our results with the three state-of-the-art attacks **LIE** [4], **Fang** [10] and **Min-Max** [33] that require knowledge of benign updates or real data. We make the following choices regarding the parametrization of the defenses. As defenses are unknown to the attacker in our scenario, we implement the version of the Min-Max attack that is designed for unknown defenses and achieves the best results. For the Fang attack, the original paper assumed knowledge of the defense. We here use the version of the Fang attack that assumes *TRmean* or *Median* as the defense, which is the only source code provided by the authors. Otherwise, we use the parameters that produced the best results in the original papers.

C. Comparison with baselines

***ASR* and *DPR*.** Our main results for the attack success rate and defense pass rate are shown in Tab. II and Fig. 4. Among all of the baseline methods, Min-Max attack is the most successful attack, with high *ASR* even on low *DPR*. In general, our experimental evaluation demonstrates that the proposed data-free attack strategies, DFA-R and DFA-G, are able to achieve similar or even slightly higher attack success rate than the baseline attacks, which require full knowledge of benign updates or a large quantity of raw data. DFA-G outperforms DFA-R in terms of *DPR* for most results on different datasets, which shows its stealthiness. During the first rounds of training, the attack is relatively weak as the global model does not yet provide a good enough model to generate effective poisoning, as indicated by our experimental results. Once the model converges, the polished model guides the attack and the attack success increases.

Specifically, from the results of Fashion-MNIST, DFA-R is better than DFA-G and all baselines when *mKrum* and *TRmean* are used to defend. *Bulyan* rejects on average more updates while *Median* merely includes the median of each model parameter from all of the clients. Both make it hard to inject malicious data into the model, leading to the low pass rate for DFA-R and hence higher effectiveness of the more stealthy DFA-G. Correspondingly, DFA-R performs better *mKrum* and *TRmean* as they allow it to pass the defense more frequently.

On the other hand, DFA-G performs well for Cifar-10 due to the fact that training Cifar-10 networks with more layers (parameters) results in slower convergence so that it favours attacks that continuously circumvent the defenses. Also, the use of 3-channel RGB data increases the diversity of benign updates. As a consequence, the level of uncertainty is generally higher during training, so that it becomes easier to pass the defense as the benign updates are not consistent enough to act as a reference point that can be used to detect malicious images. For the same reason, *DPR* of both DFA-G and DFA-R is higher on Cifar-10 than on Fashion-MNIST. However, Fang and Min-Max are not more successful on Cifar-10. Min-Max, which is aware of benign updates and hence can adapt to different datasets, already integrates dataset-specific behavior that allows it to adapt to Fashion-MNIST's low diversity. Fang rarely passes defenses, regardless of the dataset. The results of *ASR* for Fang without knowing the exact defense is consistent with the original results [10, 33].

For SVHN, both DFA-R and DFA-G achieve competitive *ASR* compared with the baselines. The only exception is Median where Min-max clearly outperforms our attacks. The result can be explained by the complexity of SVHN. SVHN is more complex than Fashion-MNIST, so it benefits from the additional knowledge Min-Max leverages to craft the update. In contrast, Cifar-10 also has a higher complexity, but experiments show that *Median* has a low accuracy for Cifar-10 even in the absence of attacks if there is data heterogeneity, as it does not include important information in the model. So it does not make so much of a difference which attack is applied. The *DPR* of DFA is lower than most other attacks for SVHN, in contrast to the other datasets. The results show that the effectiveness of the attacks depends on a combination of dataset and applied defense.

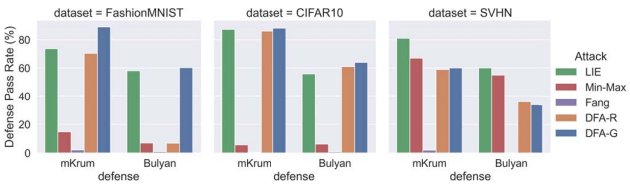


Fig. 4: Defense pass rate (*DPR*) on Dirichlet distribution. $\beta = 0.5$ for Fashion-MNIST, Cifar-10 and SVHN.

We now consider the baseline attacks in more detail. LIE

TABLE II: Attack success rate (*ASR*) and the maximum accuracy (acc_m) accordingly under attacks on Dirichlet distribution. $\beta = 0.5$. The accuracy without attacks and defenses acc for Fashion-MNIST, Cifar-10 and SVHN is 82, 50, and 86, respectively, it is reasonable for our lightweight CNN [4].

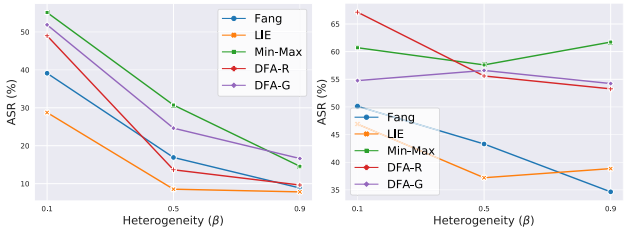
Dataset	Defense	Fang		LIE		Min-Max		DFA-R		DFA-G	
		acc (%)	*ASR* (%)	acc (%)	*ASR* (%)	acc (%)	*ASR* (%)	acc (%)	*ASR* (%)	acc (%)	*ASR* (%)
Fashion-MNIST	*mKrum*	73.5	**10.37**	72.7	**11.34**	67.3	**17.93**	52.6	**35.85**	64.3	**21.59**
	Bulyan	68.1	**16.91**	75.0	**8.54**	56.8	**30.73**	70.8	**13.66**	59.8	**27.07**
	TRmean	30.9	**62.32**	59.9	**26.95**	37.8	**53.90**	21.9	**73.29**	51.3	**37.44**
	Median	61.1	**25.49**	73.4	**10.49**	62.0	**24.39**	62.0	**24.39**	60.9	**25.73**
Cifar-10	*mKrum*	34.1	**31.80**	33.5	**33.00**	27.8	**44.40**	24.6	**50.80**	24.4	**51.20**
	Bulyan	28.4	**43.30**	31.4	**37.20**	21.2	**57.60**	22.2	**55.60**	21.7	**56.60**
	TRmean	13.9	**72.20**	13.1	**73.80**	12.6	**74.80**	14.4	**71.20**	12.5	**75.00**
	Median	24.5	**51.00**	37.0	**26.00**	24.9	**50.20**	24.7	**50.60**	23.8	**52.40**
SVHN	*mKrum*	81.85	**4.83**	80.87	**5.97**	28.03	**67.41**	60.33	**29.85**	26.36	**69.35**
	Bulyan	73.03	**15.08**	50.18	**41.65**	19.66	**77.14**	19.30	**77.56**	42.70	**50.35**
	TRmean	19.59	**77.22**	46.76	**45.63**	42.54	**50.53**	42.06	**51.09**	41.13	**52.17**
	Median	59.67	**30.62**	83.63	**2.76**	43.38	**49.56**	68.08	**20.84**	65.09	**24.31**

appears to be weaker than other attacks since it applies only a minor static shift to the mean of benign updates in order to pass defenses. This results in LIE's high *DPR* but it limits its attack effectiveness. In contrast, Min-Max attack trains (maximizes) the scale of the shifting from the mean of benign updates each round so as to enhance effectiveness, especially under heterogeneous data. This the reason why it achieves good *ASR* even with low *DPR*. The few times it overcomes the defense are sufficient for the crafted malicious updates to permanently damage the model. Fang attack has the least *DPR*, as it steers the global model parameters to the reverse direction. It is even more easily detected by the defenses than Min-Max, to the extent that the attack effectiveness is severely reduced.

D. Data heterogeneity level

We evaluate the impact of different levels of data heterogeneity on the *ASR* of attacks. Specifically, we choose $\beta = 0.1$ as the most heterogeneous case while $\beta = 0.9$ is the least heterogeneous case. Fig. 5 displays the results for Fashion-MNIST and Cifar-10 when *Bulyan* is used as a defense, which is a defense our attacks usually do not achieve the highest attack success rate, as can be seen from Tab. II. In general, the effectiveness for all attacks increases with an increased level of data heterogeneity, since more heterogeneity means that the benign updates are more diverse and hence detection of outliers is harder. The global model accuracy decreases on more heterogeneous data without attacks. This is consistent with the intuitive expectation that data of higher heterogeneity in an FL system results in poorer global accuracy within the same number of training rounds.

From Fig. 5, we can observe that for the aggressive *Bulyan* defense, the Min-Max attack achieves mostly the best performance among all of the attacks. Attacks with full knowledge of benign updates as well as adaptive weights for maliciously shifting the mean is expected to work better. That is especially true under aggressive defenses because in contrast to our attacks, Min-Max has access to information necessary to ensure their updates are less suspicious than others. Yet, thanks to the enhanced stealthiness, DFA-G outperforms Min-Max when data is less heterogeneously distributed among clients. Accordingly, DFA-R achieves the best results when $\beta = 0.1$ on Cifar-10 dataset. In this scenario, the requirement of stealthiness is the least for all of the six scenarios because Cifar-10, as discussed above, has more diverse updates and the high degree of heterogeneity further increases the diversity, making it hard to detect outliers. Additionally, the *ASR* of LIE and Fang attack decreases drastically with decreased heterogeneity. LIE attack adds a static minor shift to the true mean as it is designed to attack independent and identical distribution scenarios. For more heterogeneous updates, LIE attack is more likely to pass the defense and have an impact. Fang attack usually requires knowledge of the defense; in the absence of this knowledge, it fares better when its behavior is harder to be detected. The results on SVHN dataset show similar trends with regard to data heterogeneity. As for Cifar-10, the *ASR* for $\beta = 0.9$ may exceeds the *ASR* for $\beta = 0.5$, e.g., DFA-G on SVHN has an *ASR* of 71.68% and 50.35% for $\beta = 0.9$ and $\beta = 0.5$, respectively.



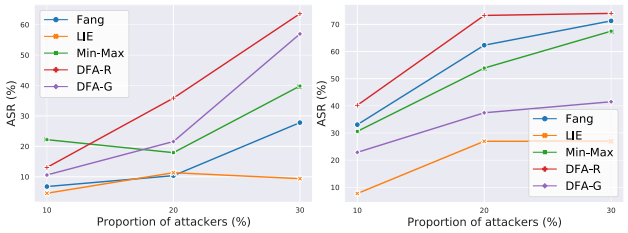
(a) Fashion-MNIST (b) Cifar-10

Fig. 5: ASR(%) for attacks under different levels of training data heterogeneity on Fashion-MNIST and Cifar-10 dataset.

E. Different proportion of attackers

In this section, we demonstrate the applicability of our proposed attack for different numbers of attackers. In order to show our effectiveness, we choose *TRmean*, which is a

statistic-based defense, and *mKrum*, which is a distance-based defense, for our experimental results presented in Fig. 6. The results are evaluated on the Fashion-MNIST dataset. We vary the attacker proportion from 10% to 30% as we do not expect the attackers of an FL learning system to exceed 30%. To create heterogeneous data, we follow the Dirichlet distribution with $\beta = 0.5$ as in Tab. II.



(a) *mKrum* (b) *TRmean*

Fig. 6: ASR(%) for attacks under different proportions of attackers on *mKrum* and *TRmean* defense.

The results of Fig. 6 demonstrate the consistent effectiveness of DFA compared with other attacks. The more attackers we have, the higher the attack success rate, as one expects. Yet, DFA achieves the highest attack success rate, compared to other attacks. DFA-R usually has the best performance, with the exception of 10% on *mKrum*, where Min-Max attack has the best *ASR*. Indeed, it is easier to defend against a smaller number of attacks in the simpler dataset. As Min-max has more knowledge about the benign updates, it is then able to send in more malicious models than DFA in this easy-to-defend case.

F. Ablation analysis on DFA *components*

1) Generator training epochs: Here, we empirically investigate the convergence towards the optimal loss, where DFA-R is minimizing its loss but DFA-G is maximizing it. Fig. 7 shows the results for Fashion-MNIST on all four defenses. It can be clearly seen that the local training for generating malicious images converges to a local optimum. For both of our proposed attacks, DFA-R and DFA-G, we only need a few epochs to train. For DFA-R, E is 5 for Fashion-MNIST, and $E = 10$ for Cifar-10 and SVHN as Fashion-MNIST is easier to train.

2) Comparison with non-training approach: Given that the training converges fast, we also investigate the impact of training in comparison to just using a randomly initialized filter layer for DFA-R and a randomly initialized generator for DFA-G without any updating over rounds. As explained, according to the definition, *DPR* is measured only on *mKrum* and *Bulyan* defenses. We hence report the results for *TRmean* and *Median* as "N/A". The maximum accuracies without attack are the same as Tab. II.

The results can be seen in Tab. III and confirm that training according to the current global model is indeed necessary. For DFA-R, training a single layer aims at generating images that confuse the global model. Without the training step, the

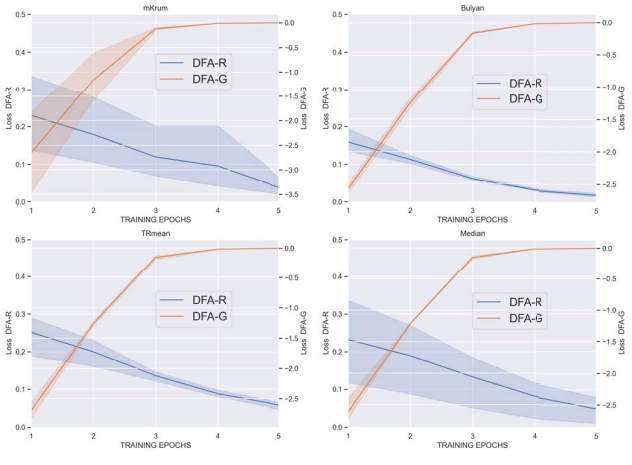


Fig. 7: Local training process of both DFA-G and DFA-R on Fashion-MNIST.

injection of DFA-R is less malicious. Thus, *ASR* usually decreases without training, except for Fashion-MNIST with *Bulyan* defense. This observation is due to the fact that training DFA-R reduces the stealthiness of the attack by focusing on effectiveness and hence DFA-R passes *Bulyan* more often without training. This is consistent with our results in Fig. 4 that *Bulyan* significantly reduces the *DPR* of DFA-R.

TABLE III: *ASR* and *DPR* for (non-)training approach where "Static" refers to non-training way with only randomly initialized. "Fashion" and "Cifar" is short for Fashion-MNIST and Cifar-10 datasets.

Attack	Defense	Static		Trained	
		ASR(%)	DPR(%)	ASR(%)	DPR(%)
DFA-R Fashion	*mKrum*	18.17	87.78	35.85	70.33
	TRmean	37.20	N/A	73.29	N/A
	Bulyan	23.66	57.50	13.66	6.86
	Median	21.22	N/A	24.39	N/A
DFA-G Fashion	*mKrum*	17.07	88.33	21.59	89.02
	TRmean	30.73	N/A	37.44	N/A
	Bulyan	24.88	65.26	27.07	69.33
	Median	22.44	N/A	25.73	N/A
DFA-R Cifar	*mKrum*	50.00	85.20	50.80	86.04
	TRmean	71.14	N/A	71.20	N/A
	Bulyan	56.00	60.98	55.65	61.05
	Median	48.60	N/A	50.60	N/A
DFA-G Cifar	*mKrum*	38.60	56.46	51.20	88.14
	TRmean	71.40	N/A	75.00	N/A
	Bulyan	47.80	37.35	56.60	63.99
	Median	50.60	N/A	52.40	N/A

When it comes to DFA-G, training helps to enhance stealthiness. The impact can be clearly seen from the results for *DPR* in Tab. III, especially for *Bulyan*. Only for a relatively lenient defense like *mKrum*, the training has little additional impact as *DPR* is already high without training. These results also reflect the minor increase of *DPR* from Fashion-MNIST to Cifar-10 dataset for *mKrum* in Fig. 4.

3) Impact of the regularization term: In this part, we conduct an ablation study for our proposed distance-based loss, which adds a regularization term to the original cross-entropy loss function. Tab. IV shows both *ASR* and *DPR* with and without the regularization term on Fashion-MNIST. For DFA-R, the effectiveness of the regularization term is more apparent for *mKrum*. However, for DFA-G, the increase is most notable for *Bulyan*. This is because the regularization term in the less stealthy DFA-R is insufficient for passing *Bulyan* whereas it is what enables DFA-G to pass *Bulyan* frequently. In contrast, DFA-G does not require the regularization term for *mKrum* as it already passes without extra regularization.

TABLE IV: *ASR* and *DPR* for ablation test of the regularization term proposed by our distance-based loss.

Attack	Defense	without regularization		with regularization	
		ASR(%)	DPR(%)	ASR(%)	DPR(%)
DFA-R	*mKrum*	17.68	41.92	35.85	70.33
	TRmean	58.78	N/A	73.29	N/A
	Bulyan	10.73	3.32	13.66	6.86
	Median	23.72	N/A	24.39	N/A
DFA-G	*mKrum*	20.98	87.34	21.59	89.02
	TRmean	31.71	N/A	37.44	N/A
	Bulyan	22.32	60.27	27.07	69.33
	Median	23.78	N/A	25.73	N/A

4) Synthetic vs real data: In order to demonstrate the effectiveness of our malicious synthetic data, we compare the *ASR* of our attacks to a version of the attack that uses real data, i.e., we use a set of real images instead of the synthetic image set S. We assign the number of real images owned by the attackers under the same Dirichlet distribution as for benign users. The results for the four defenses on both datasets are shown in Fig. 8 with stripped visualization. "Real-data" in the figure refers to the results of *ASR* using real data paired with the uniformly chosen label $\tilde{Y}$ to train $\boldsymbol{w}(t)$ with distance-based loss as described in Sec. III similarly for the synthetic data. Fig. 8 shows the effectiveness of our malicious synthetic data generated by DFA-R and DFA-G as *ASR* outperforms the case of using real images. That is expected because our synthetic images are specifically constructed such that the attack is very effective but at the same time stealthy. Thus, even if data is present at the attacker, a data-free attack can be the better choice. Consequently, it is usually not necessary for the attacker to invest the overhead of obtaining data.

V. Defense for DFA: RefD

Based on the results of the previous section, our attack is highly effective against known defenses. Yet, the attack might not withstand defenses that are crafted with data-free attacks in mind. Thus, in this section, we design and evaluate a novel defense that specifically addresses the reasons why existing defenses are insufficient.

Let us first state why existing defenses fail. *mKrum* and *Bulyan* reject updates that differ greatly from others. Our regularization term ensures that our updates do not differ too much from the global model from the previous round,

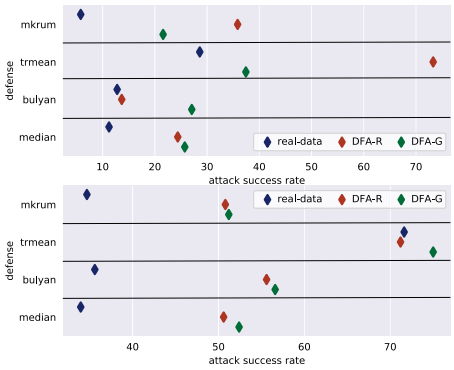


Fig. 8: Comparison of *ASR* (%) of real data and synthetic data by DFA-R and DFA-G with four defenses on Fashion-MNIST and Cifar-10.

which at least once the model starts to converge is close to the models submitted by benign clients. Statistical methods lose information about the distribution but medians or trimmed means also shift easily without the need for an attacker to provide outlier data [4].

A. Design of RefD (RefD)

The defense is designed with data-free attacks in mind and overcomes the drawbacks of existing defenses. We rely on a reference dataset $\mathcal{D}_r$ to detect unusual classification patterns. Upon receiving an updated model from clients, the server uses that model and executes the model inference on $\mathcal{D}_r$, i.e., they compute predicted class probabilities. We design a novel statistic, D-score, which identifies the model updates whose outputs have either biased prediction or low confidence. To evaluate the effectiveness of RefD, we apply RefD on both our proposed attacks and the state-of-the-art attacks, for both balanced and heterogeneous data distributions.

Assumptions. RefD is a server-side defense. RefD requires that the server owns a small reference set $\mathcal{D}_r$ of real data with correct labels. The quantity of each class label is assumed to be balanced.

The design core of RefD is the D-score for each model update received: a low score indicates a high risk of update being malicious and results in the server rejecting the update. The D-score is computed based on two parts: the balance value and the confidence value of updates. The balance value determines how balanced the outputs from the updated model are, to detect updates that are biased toward a specific class, such as DFA-G, LIE [4], and Min-Max [33]. The confidence value measures the confidence in predicting a class and rejects updates that result in low confidence, which are the objective of DFA-R and Fang [10].

Before explaining those values, we first explain background notations. $\boldsymbol{w}_i$ defined as the updated classifier model received from the client i, which maps the data input into two kinds of output. Specifically, $\boldsymbol{w}_i^p(\cdot)$ maps the data into the per class probability vector and $\boldsymbol{w}_i^h(\cdot)$ maps the data into the per class

one-hot encoding vector. Both vectors have the length of L, corresponding to the number of classes.

Balance value. To tackle the attack type which causes bias in classification, we define the balance value B_i for the updated model of client i as the inverse of the standard deviation of the class label distribution:

$$B_i = \begin{cases} \dfrac{1}{std(A_i)}, \text{if } std(A_i) \neq 0 \\ 1, \text{if } std(A_i) = 0 \end{cases} \tag{6}$$

where $std(\cdot)$ is the standard deviation over all class labels and A_i consists of the aggregated number of predicted labels of each class. For instance, A_i for Cifar-10, is a set of 10 values, i.e., the number of predicted samples per class. To compute A_i, we first apply $\boldsymbol{w}_i^h$ on each sample in D_r and aggregate them. Overall, the non-biased output prediction from benign clients results into a better balanced A_i across all classes and thus a higher value of B_i. The adversarial parties on the other hand should have a lower value of B_i.

Confidence value. This value quantifies the average confidence when using the updated model on $\mathcal{D}_r$. Specifically, for a data sample j in $\mathcal{D}_r$, we let the confidence of applying classifier of client i, M_{ij} as the biggest element of the output probability vector $\boldsymbol{w}_i^p(\mathcal{D}_r(j))$. We thus define the confidence value, V_i, as:

$$V_i = \frac{1}{|\mathcal{D}_r|} \sum_{j=1}^{|\mathcal{D}_r|} M_{ij}. \tag{7}$$

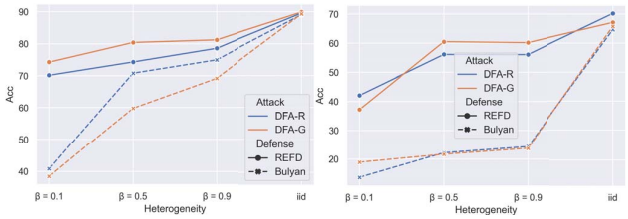
Low confidence values indicate higher risks of being adversarial updates. The potential drawback of using V_i to detect adversarial behaviour is that the low confidence may also appear in the early training epochs of honest clients.

D-Score. For each client update, we combine the balance value, B, and the confidence value, V, to detect a wide range of adversarial behaviors. Motivated by F_β Score [32], we define the D-Score to evaluate the quality of an intermediate training model from client i as:

$$D - Score = (1 + \alpha^2) \times \frac{B_i \times V_i}{\alpha^2 B_i + V_i}, \tag{8}$$

where α is a hyper-parameter to weigh the importance between balance value and confidence value. It can be set as a specific value according to what the central server knows or suspects about the executed attack. It can also be adaptive and learned over epochs, but we consider this out-of-scope for the paper and a good avenue for future work. Instead, we set $\alpha = 1$ to represent the equal importance of B_i and V_i. If the predictions are perfectly balanced and have a high confidence, we have a D-Score of 1. When B_i is reduced while V_i stays constant, the D-Score is reduced, mirroring the increased bias. Analogously for V_i, a lower value for V_i leads to a lower D-Score to indicate the lower confidence. Moreover, as we designed the defense with data-free attacks in mind, we expect it to work better for those than for other attacks, for which defenses already exist.

Removing attackers. After calculating the D-Score for each update of a given round, the server rejects the updates with the X lowest D-Scores. The server then excludes them for aggregation. The method is the same as used by *mKrum* [5]). X is determined by the server's assumptions about the fraction of attacker, i.e., the more attacker they expect, the higher they choose X.



(a) Fashion-MNIST (b) Cifar-10

Fig. 9: Accuracy(%) for REFD on Fashion-MNIST and Cifar-10 datasets with different levels of data heterogeneity, compared with the maximum accuracy under *Bulyan* defense.

B. Evaluation for REFD

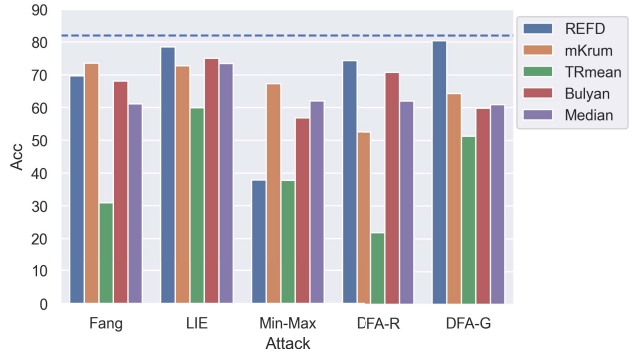
Experimental settings. To evaluate the effectiveness of REFD, we compare the accuracy of the global model in the presence of the new defense. We use the full test set for the respective dataset in the presented results but also experimented with smaller reference datasets (1000 images instead of 10000) and found no significant difference. Hence, smaller datasets can be used to increase efficiency and lower the requirements in terms of data availability at the server side. However, the reference set has to be balanced among class labels to compute the balance value reliably. REFD is evaluated on both Fashion-MNIST and Cifar-10 dataset. Additionally, our experiments include four different levels of data heterogeneity: independent and identical distributed (*i.i.d*) and three heterogeneity levels ($\beta = 0.1, 0.5, 0.9$, where $\beta = 0.1$ indicates the highest level of heterogeneity) as in Sec. IV-D. We also evaluate the impact of different level of data heterogeneity since defenses are sensitive to the heterogeneity, especially for distance-based defenses. Intuitively, a higher level of training data heterogeneity makes defense more difficult. The robustness of a defense in presence of high data heterogeneity is important to various of real-world application scenarios.

As in other works [10, 33], we set the proportion of attackers in the system to be 20% and $X = 2$. We compare REFD against *Bulyan*, the most effective SOTA defense for our attack.

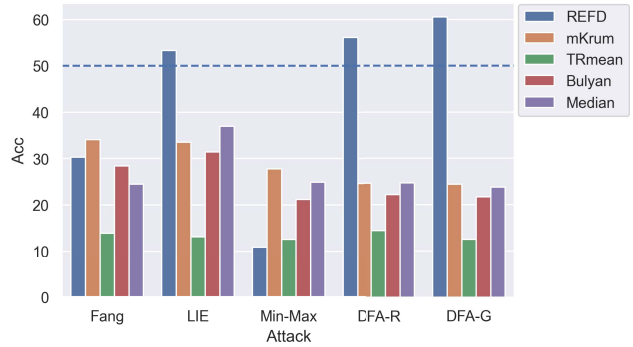
Results for defense DFA.

From Fig. 9, we see that REFD significantly outperforms *Bulyan*. The advantage of our defense is obvious when the heterogeneity of the data is high. For $\beta = 0.1$ and Fashion-MNIST, REFD achieves an accuracy of more than 70% when the attack is DFA-R and close to 75% for DFA-G. In contrast, the accuracy of *Bulyan* is only around 40%. *Bulyan* is relatively effective for *i.i.d* data, achieving a similar value as REFD for both attacks.

The results on Cifar-10 confirm the superiority of REFD. As noted in Sec. IV, the accuracy on Cifar-10 is generally



(a) Fashion-MNIST



(b) Cifar-10

Fig. 10: Comparison of Accuracy(%) for defenses with state-of-the-art attacks on Fashion-MNIST and Cifar-10 datasets.

lower. Yet, REFD is clearly the more effective defense. In the presence of data heterogeneity, the accuracy is at least twice as high for REFD than *Bulyan*.

For both datasets, the advantage of REFD is least pronounced for the *i.i.d* setting. For *i.i.d*, the benign updates are very similar, hence making it barely possible for our attacks to deviate without being detected. Thus, there is little difference between defenses. The results for Fashion-MNIST and Cifar-10 differ in that the final global model accuracy difference between REFD and *Bulyan* is larger for Cifar-10. The data complexity of Cifar-10 is higher, increasing the difficulty of defending, so that our specifically designed defense shows a more pronounced advantage.

The achieved accuracy is close to the accuracy achieved without attacks and defenses. For instance, for $\beta = 0.5$, the accuracy without attacks and defenses is 82% for Fashion-MNIST, which is less than 2% higher than the accuracy for REFD for DFA-G. For DFA-R, the attack decreases the accuracy by about 10%. For Cifar-10, the accuracy with and without attacks are almost equal. Indeed, the accuracy with attack and defense is insignificantly higher for some settings, which is likely due to randomness.

Defending against other attacks. REFD is designed with DFA in mind, as our goal is to show that we can defend against data-free attacks. The design does not necessarily work against all attacks. Here, we establish whether the defense can nevertheless defend against Fang, LIE, and Min-Max attack and compare it with the results for DFA.

The results are reported in Fig. 10. We also follow the setting of 20% attacking proportion and compare the maximum global model accuracy for the state-of-the-art defences and REFD. We include the baseline accuracy with no attack and no defense as the dashed line. From Fig. 10, we can see that REFD has a good defending performance in general. However, it is not always the best among the-state-of-the-art defenses. Specifically, for LIE attack, REFD gets the best defending performance. LIE shifts the true statistical features of the benign updates, which can easily be caught by the balance value B of REFD. Moreover, REFD also protects the model from Fang attack, where it achieves the second best ranking on both datasets. REFD works well for Fang since Fang updates malicious models on the opposite direction, which causes low confidence, i.e., low V. However, REFD is less effective against Min-Max than other defenses, as Min-Max's scaling technique should not affect balance and confidence value much. In summary, REFD protects well against data-free attacks presented in this paper and can also protect against other attacks. However, it is not a generic defense and hence should be applied in combination with other defence mechanisms. It is also interesting to note that with RefD, the global model accuracy can even be higher than the baseline on Cifar-10. This result implies that RefD has benefits in the presence of data heterogeneity in comparison to FedAVG.

C. Overhead analysis for defense

The defense does not add any communication, so merely the computation complexity is affected. The defense first evaluates the local update of each client for each image in the reference dataset, so the cost is $\mathcal{O}(|\mathcal{D}_r|K)$ times the cost of evaluating the update. Furthermore, the D-Score needs to be computed, which is linear in $\mathcal{O}(|\mathcal{D}_r|)$ as we compute the standard derivation (for B) and the maximum (for V) of $\mathcal{O}(|\mathcal{D}_r|)$ values. Last, we determine the clients with the smallest values, which has complexity $\mathcal{O}(K)$. Overall, evaluating updates is of a lower complexity than training new models, so the overhead is not prohibitive and can be reduced by using a smaller set $\mathcal{D}_r$.

VI. CONCLUSION

We propose DFA, the first data-free untargeted attack on FL. Our results confirm that data-free attacks can be similarly or even more effective than other attacks that require data or benign updates, due to generating synthetic images to train on that are particularly useful at steering the model into the wrong directions. Furthermore, we design a defense strategy REFD that effectively protects against the proposed DFA and existing attacks by leveraging the statistics of model outputs in predicting reference data. In the future, we want to explore DFA on different data types, e.g., text, and check whether combining synthetic and real data in an attack can improve attack effectiveness and to what extent data is needed in a defense.

REFERENCES

[1] Martín Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. *CoRR*, abs/1701.07875, 2017.

[2] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *International Conference on Artificial Intelligence and Statistics*, volume 108, pages 2938–2948, 2020.

[3] Carlo Baldassi, Fabrizio Pittorino, and Riccardo Zecchina. Shaping the learning landscape in neural networks around wide flat minima. *Proc. Natl. Acad. Sci. USA*, 117(1):161–170, 2020.

[4] Gilad Baruch, Moran Baruch, and Yoav Goldberg. A little is enough: Circumventing defenses for distributed learning. In *Conference on Neural Information Processing Systems*, pages 8632–8642, 2019.

[5] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Conference on Neural Information Processing Systems*, pages 119–129, 2017.

[6] Jian Chen, Xuxin Zhang, Rui Zhang, Chen Wang, and Ling Liu. De-pois: An attack-agnostic defense against data poisoning attacks. *IEEE Trans. Inf. Forensics Secur.*, 16:3412–3425, 2021.

[7] Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014.

[8] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. EMNIST: an extension of MNIST to handwritten letters. *CoRR*, abs/1702.05373, 2017.

[9] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. Technical report, Naval Research Lab Washington DC, 2004.

[10] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. Local model poisoning attacks to byzantine-robust federated learning. In *USENIX Security Symposium*, pages 1605–1622, 2020.

[11] Yann Fraboni, Richard Vidal, and Marco Lorenzi. Free-rider attacks on model aggregation in federated learning. In *International Conference on Artificial Intelligence and Statistics*, pages 1846–1854, 2021.

[12] Clement Fung, Chris J. M. Yoon, and Ivan Beschastnikh. The limitations of federated learning in sybil settings. In *23rd International Symposium on Research in Attacks, Intrusions and Defenses*, pages 301–316, 2020.

[13] Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2018.

[14] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *CoRR*, abs/1909.06335, 2019.

[15] Sai Praneeth Karimireddy, Martin Jaggi, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian U Stich, and Ananda Theertha Suresh. Breaking the centralized barrier for cross-device federated learning. *Advances in Neural Information Processing Systems*, 34:28663–28676, 2021.

[16] Ekaterina Khramtsova, Christian A. Hammerschmidt, Sofiane Lagraa, and Radu State. Federated learning for cyber security: SOC collaboration for malicious URL detection. In *40th IEEE International Conference on Distributed Computing Systems, ICDCS 2020, Singapore, November 29 - December 1, 2020*, pages 1316–1321. IEEE, 2020.

[17] Nicolas Kourtellis, Kleomenis Katevas, and Diego Perino. Flaas: Federated learning as a service. In *Proceedings of the 1st workshop on distributed machine learning*, pages 7–13, 2020.

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

[19] Yann LeCun, Yoshua Bengio, and Geoffrey E. Hinton. Deep learning. *Nat.*, 521(7553):436–444, 2015.

[20] Huiying Li, Shawn Shan, Emily Wenger, Jiayun Zhang, Haitao Zheng, and Ben Y. Zhao. Blacklight: Scalable defense for neural networks against query-based black-box attacks. In Kevin R. B. Butler and Kurt Thomas, editors, *31st USENIX Security Symposium, USENIX Security 2022, Boston, MA, USA, August 10-12, 2022*, pages 2117–2134. USENIX Association, 2022.

[21] Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Trans. Neural Networks Learn. Syst.*, 33(12):6999–7019, 2022.

[22] Jierui Lin, Min Du, and Jian Liu. Free-riders in federated learning: Attacks and defenses. *CoRR*, abs/1911.12560, 2019.

[23] Tao Lin, Lingjing Kong, Sebastian U. Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. In *Conference on Neural Information Processing Systems*, 2020.

[24] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, volume 54, pages 1273–1282, 2017.

[25] El Mahdi El Mhamdi, Rachid Guerraoui, and Sébastien Rouault. The hidden vulnerability of distributed learning in byzantium. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 3518–3527, 2018.

[26] Mohammad Naseri, Jamie Hayes, and Emiliano De Cristofaro. Toward robustness and privacy in federated learning: Experimenting with local and central differential privacy. *CoRR*, abs/2009.03561, 2020.

[27] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *2019 IEEE Symposium on Security and Privacy, SP 2019*, pages 739–753. IEEE, 2019.

[28] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[29] Thien Duc Nguyen, Samuel Marchal, Markus Miettinen, Hossein Fereidooni, N. Asokan, and Ahmad-Reza Sadeghi. Dïot: A federated self-learning anomaly detection system for iot. In *IEEE International Conference on Distributed Computing Systems*, pages 756–767, 2019.

[30] Mauro Ribeiro, Katarina Grolinger, and Miriam AM Capretz. Mlaas: Machine learning as a service. In *2015 IEEE 14th international conference on machine learning and applications (ICMLA)*, pages 896–902. IEEE, 2015.

[31] Nicola Rieke, Jonny Hancox, Wenqi Li, Fausto Milletari, Holger Roth, Shadi Albarqouni, Spyridon Bakas, Mathieu N. Galtier, Bennett A. Landman, Klaus H. Maier-Hein, Sébastien Ourselin, Micah J. Sheller, Ronald M. Summers, Andrew Trask, Daguang Xu, Maximilian Baust, and M. Jorge Cardoso. The future of digital health with federated learning. *CoRR*, abs/2003.08119, 2020.

[32] Yutaka Sasaki et al. The truth of the f-measure. *Teach tutor mater*, 1(5):1–5, 2007.

[33] Virat Shejwalkar and Amir Houmansadr. Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning. In *Annual Network and Distributed System Security Symposium*, 2021.

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

[35] Mengkai Song, Zhibo Wang, Zhifei Zhang, Yang Song, Qian Wang, Ju Ren, and Hairong Qi. Analyzing user-level privacy attack against federated learning. *IEEE J. Sel. Areas Commun.*, 38(10):2430–2444, 2020.

[36] Mengkai Song, Zhibo Wang, Zhifei Zhang, Yang Song, Qian Wang, Ju Ren, and Hairong Qi. Analyzing user-level privacy attack against federated learning. *IEEE J. Sel. Areas Commun.*, 38(10):2430–2444, 2020.

[37] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik G. Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *IEEE International Conference on Computer Vision*, pages 945–953, 2015.

[38] Dianbo Sui, Yubo Chen, Jun Zhao, Yantao Jia, Yuantao Xie, and Weijian Sun. Feded: Federated learning via ensemble distillation for medical relation extraction. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020*, pages 2118–2128. Association for Computational Linguistics, 2020.

[39] Jingwei Sun, Ang Li, Louis DiValentin, Amin Hassanzadeh, Yiran Chen, and Hai Li. FL-WBC: enhancing robustness against model poisoning attacks in federated learning from a client perspective. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 12613–12624, 2021.

[40] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014.

[41] Vale Tolpegin, Stacey Truex, Mehmet Emre Gursoy, and Ling Liu. Data

poisoning attacks against federated learning systems. In Liqun Chen, Ninghui Li, Kaitai Liang, and Steve A. Schneider, editors, *Computer Security - ESORICS 2020*, volume 12308, pages 480–501. Springer, 2020.

[42] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris S. Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. In *International Conference on Learning Representations*, 2020.

[43] Steven H. Wu, Rachel Schwartz, David J. Winter, Donald F. Conrad, and Reed A. Cartwright. Estimating error models for whole genome sequencing using mixtures of dirichlet-multinomial distributions. *Bioinform.*, 33(15):2322–2329, 2017.

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

[45] Chulin Xie, Keli Huang, Pin-Yu Chen, and Bo Li. DBA: distributed backdoor attacks against federated learning. In *8th International Conference on Learning Representations,*, 2020.

[46] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Trans. Intell. Syst. Technol.*, 10(2):12:1–12:19, 2019.

[47] Dong Yin, Yudong Chen, Kannan Ramchandran, and Peter L. Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *Proceedings of the International Conference on Machine Learning*, volume 80, pages 5636–5645, 2018.

[48] Hongxu Yin, Arun Mallya, Arash Vahdat, Jose M. Alvarez, Jan Kautz, and Pavlo Molchanov. See through gradients: Image batch recovery via gradinversion. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021*, pages 16337–16346. Computer Vision Foundation / IEEE, 2021.

[49] Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan H. Greenewald, Trong Nghia Hoang, and Yasaman Khazaeni. Bayesian nonparametric federated learning of neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 7252–7261, 2019.

[50] Syed Zawad, Ahsan Ali, Pin-Yu Chen, Ali Anwar, Yi Zhou, Nathalie Baracaldo, Yuan Tian, and Feng Yan. Curse or redemption? how data heterogeneity affects the robustness of federated learning. In *Thirty-Fifth AAAI Conference on Artificial Intelligence*, pages 10807–10814, 2021.

[51] Chaoning Zhang, Philipp Benz, Adil Karjauv, and In So Kweon. Data-free universal adversarial perturbation and black-box attack. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 7848–7857. IEEE, 2021.

[52] Jiale Zhang, Bing Chen, Xiang Cheng, Huynh Thi Thanh Binh, and Shui Yu. Poisongan: Generative poisoning attacks against federated learning in edge computing systems. *IEEE Internet Things J.*, 8(5):3310–3322, 2021.

[53] Jiale Zhang, Junjun Chen, Di Wu, Bing Chen, and Shui Yu. Poisoning attack in federated learning using generative adversarial nets. In *18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications / 13th IEEE International Conference On Big Data Science And Engineering, TrustCom/BigDataSE 2019, Rotorua, New Zealand, August 5-8, 2019*, pages 374–380. IEEE, 2019.

[54] Jie Zhang, Bo Li, Jianghe Xu, Shuang Wu, Shouhong Ding, Lei Zhang, and Chao Wu. Towards efficient data free blackbox adversarial attack. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 15094–15104. IEEE, 2022.

[55] Jingwen Zhang, Jiale Zhang, Junjun Chen, and Shui Yu. GAN enhanced membership inference: A passive local attack in federated learning. In *2020 IEEE International Conference on Communications, ICC 2020, Dublin, Ireland, June 7-11, 2020*, pages 1–6. IEEE, 2020.

[56] Lingchen Zhao, Shengshan Hu, Qian Wang, Jianlin Jiang, Chao Shen, Xiangyang Luo, and Pengfei Hu. Shielding collaborative learning: Mitigating poisoning attacks through client-side detection. *IEEE Trans. Dependable Secur. Comput.*, 18(5):2029–2041, 2021.

[57] Zilong Zhao, Robert Birke, Aditya Kunar, and Lydia Y. Chen. Fed-tgan: Federated learning framework for synthesizing tabular data. *CoRR*, abs/2108.07927, 2021.

[58] Wenbo Zheng, Lan Yan, Chao Gou, and Fei-Yue Wang. Federated meta-learning for fraudulent credit card detection. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 4654–4660. ijcai.org, 2020.