Resisting Backdoor Attacks in Federated Learning via Bidirectional Elections and Individual Perspective

Zhen Qin[1], **Feiyi Chen**[1], **Chen Zhi**[2], **Xueqiang Yan**[3], **Shuiguang Deng**[1*]

[1]College of Computer Science and Technology, Zhejiang University, Hangzhou, China
[2]School of Software Technology, Zhejiang University, Ningbo, China
[3]Huawei Technologies Co. Ltd., Shanghai, China
{zhenqin, chenfeiyi, zjuzhichen}@zju.edu.cn, yanxueqiang1@huawei.com, dengsg@zju.edu.cn

Abstract

Existing approaches defend against backdoor attacks in federated learning (FL) mainly through a) mitigating the impact of infected models, or b) excluding infected models. The former negatively impacts model accuracy, while the latter usually relies on globally clear boundaries between benign and infected model updates. However, in reality, model updates can easily become mixed and scattered throughout due to the diverse distributions of local data. This work focuses on excluding infected models in FL. Unlike previous perspectives from a global view, we propose Snowball, a novel anti-backdoor FL framework through bidirectional elections from an individual perspective inspired by one principle deduced by us and two principles in FL and deep learning. It is characterized by a) bottom-up election, where each candidate model update votes to several peer ones such that a few model updates are elected as selectees for aggregation; and b) top-down election, where selectees progressively enlarge themselves through picking up from the candidates. We compare Snowball with state-of-the-art defenses to backdoor attacks in FL on five real-world datasets, demonstrating its superior resistance to backdoor attacks and slight impact on the accuracy of the global model.

Introduction

Federated Learning (FL) (McMahan et al. 2017) enables multiple devices to jointly train machine learning models without sharing their raw data. Due to the unreachability to distributed data, it is vulnerable to attacks from malicious clients (Wang et al. 2020), especially *backdoor attacks* that neither significantly alter the statistical characteristics of models as Gaussian-noise attacks (Blanchard et al. 2017) nor cause a distinct modification to the training data as label-flipping attacks (Liu et al. 2021), and thus, are more covert against many existing defenses (Zeng et al. 2022).

Existing defenses to backdoor attacks in FL are mainly based on a) mitigating the impact of infected models (Bagdasaryan et al. 2020; Sun et al. 2019; Xie et al. 2021; Nguyen et al. 2022; Zhang et al. 2023) or b) excluding infected models based on their deviations (Blanchard et al. 2017; Ozdayi, Kantarcioglu, and Gel 2021; Fung, Yoon, and Beschastnikh 2018; Rieger et al. 2022; Li et al. 2020a; Shejwalkar and Houmansadr 2021; Zhang et al. 2022; Shi et al. 2022; Nguyen et al. 2022). The former may negatively impact global model accuracy (Yu et al. 2021). The latter assumes globally clear boundaries between benign and infected model updates (Zeng et al. 2022). However, backdoor attacks typically manipulate a limited subset of parameters, resulting in the similarity between benign and infected model updates. Besides, the nature of Non-Independent and Identically Distributed (non-IID) data in FL increases diversity among model updates.

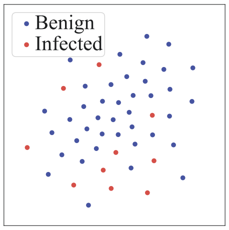


Figure 1: 2D-visualized 50 model updates in one round of FL (practical non-IID MNIST with α=0.5, PDR=0.3).

Actually, benign and infected model updates are easy to be mixed with complicatedly non-IID data (practical non-IID (Hsu, Qi, and Brown 2019; Huang et al. 2021) and feature distribution skew (Tan et al. 2022)), or with not very high poison data ratio (PDR). We experimentally demonstrate it in Figure 1, where benign and infected updates are mixedly scattered. In such cases, anomaly detections based on linear similarity may not perform satisfactorily. Besides, when facing a relatively high malicious client ratio (MCR), infected model updates are easier to be mistreated as benign ones, however, many existing defenses are only evaluated with MCR $\leq$ 10% (Xie et al. 2021; Ozdayi, Kantarcioglu, and Gel 2021; Zeng et al. 2022; Lu et al. 2022). Although model deviations may be better captured by nonlinear neural networks, *the patterns of benign models in FL are usually hard to acquire* due to unpredictable distributions and trajectory shifts of model updates. Li et al. (2020a) use the test data to generate model weights for training the detection model, but the test data with a similar distribution to all clients may be usually unavailable. Besides, model weights usually follow extremely complex distributions, making them hard to learn.

To better leverage powerful neural networks to detect mali-

[*]Shuiguang Deng is the corresponding author.

cious models, we propose Snowball, an anti-backdoor FL framework taking advantage of linear and non-linear approaches, i.e., without the need for pre-defined benign patterns and the powerful capability to capture model deviations, respectively. It treats each model update as an agent electing model updates for aggregation with an individual perspective, where the motivation comes from that: **defenses of the models, by the models, for the models.** From a global perspective as existing studies (Nguyen et al. 2022; Ozdayi, Kantarcioglu, and Gel 2021; Blanchard et al. 2017; Li et al. 2020a), benign and infected model updates may appear mixed. If we examine model updates from the perspective of individual model updates, the nearest ones may have the same purpose since both benign and infected updates wish to exclude each other from aggregation. Thus, if we make each model vote for the closest model updates, the benign model updates may get more votes when the benign clients account for the majority.

The elections in Snowball are bidirectional and conducted sequentially, i.e., 1) bottom-up election where candidate model updates nominate a small group of peers as selectees to be aggregated; and 2) top-down election that regards the selectees as benign patterns and progressively enlarges the number of selectees from the rest candidates through a variational auto-encoder (VAE), which focuses on the model-wise differences instead of benign patterns themselves. We know it may be difficult to have a one-size-fits-all approach, fortunately, Snowball can be easily integrated into existing FL systems in a non-invasive manner, since it only filters out several model updates for aggregation. For attacks that have not been mentioned in this work, aggregation can be conducted on the intersection between the updates selected by existing approaches and that of Snowball.

The main contributions of this work lie in:

1. Proposing a novel anti-backdoor FL framework named Snowball. It selects model updates with bidirectional elections from an individual perspective, contributing to the leverage of neural networks for infected model detection.
2. Proposing a new paradigm for utilizing VAE to detect infected models, i.e., progressively enlarges the selectees with focusing on the model-wise differences instead of benign patterns themselves, to better distinguish infected model updates from benign ones.
3. Conducting extensive experiments on 5 real-world datasets to demonstrate the superior attack-resistance of Snowball over state-of-the-art (SOTA) defenses when the data are complicatedly non-IID, PDR is not very high and the ratio of attackers to all clients is relatively high. Also, Snowball brings a slight impact on the global model accuracy. Codes are available at https://github.com/zhenqincn/Snowball.

Related Work

Existing work defends targeted attacks in FL by a) mitigating the impact of infected models, including a1) robust learning rate (Ozdayi, Kantarcioglu, and Gel 2021; Fung, Yoon, and Beschastnikh 2018), a2) provably secure FL by model ensemble (Xie et al. 2021; Cao, Jia, and Gong 2021), a3) adversarial learning (Zhang et al. 2023), or b) filtering out infected models or parameters, including: b1) Byzantine-robust aggregation (Blanchard et al. 2017; Yin et al. 2018), and b2) anomaly detection (Li et al. 2020a; Zhang et al. 2022; Shi et al. 2022; Shejwalkar and Houmansadr 2021; Zhang et al. 2022). Besides, there are also approaches that combine weight-clipping, noise-addition and clustering (Bagdasaryan et al. 2020; Sun et al. 2019; Nguyen et al. 2022; Rieger et al. 2022), which belong to both of the two main categories.

These approaches are validated to be effective in different scenarios. However, approaches mitigating the impact of infected models usually lower the global model accuracy (a1, a2) or rely on certain assumptions which may not be always satisfied and cause inference latency and memory consumption (a3) (Li et al. 2022). Approaches filtering out infected models usually require globally clear boundaries between benign and infected model updates (Zeng et al. 2022), which usually only occur when 1) the non-IIDness of data is not complex (IID or pathological non-IID) where model updates are easy to form distinct clusters (Nguyen et al. 2022; Ozdayi, Kantarcioglu, and Gel 2021; Rieger et al. 2022) or 2) the PDR is high ($\geq 50\%$) such that infected model updates deviate significantly from benign ones (Rieger et al. 2022; Ozdayi, Kantarcioglu, and Gel 2021). Besides, many defenses are only evaluated with MCR $\leq 10\%$ (Xie et al. 2021; Ozdayi, Kantarcioglu, and Gel 2021; Zeng et al. 2022; Lu et al. 2022).

Thus, there is a strong demand for an approach that can effectively defend against backdoor attacks when benign and infected models are scattered without clear boundaries.

Background

This work focuses on the classical FL (McMahan et al. 2017). Let $\mathbb{D} = \{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_N\}$ denote the datasets held by the N clients respectively. The goal of FL is formulated as:

$$\min_{\mathbf{w}} f(\mathbf{w}) := \sum_{i=1}^{N} \lambda_i f(\mathbf{w}, \mathcal{D}_i) \tag{1}$$

where $f(\mathbf{w}, \mathcal{D}_i) := \frac{1}{|\mathcal{D}_i|} \sum_{\xi \in \mathcal{D}_i, \xi \sim \mathcal{Z}_i} \ell(\mathbf{w}, \xi)$ is the average loss ℓ on data sample ξ of client i, where ξ follows distribution $\mathcal{Z}_i$, and λ_i is the weight of client i. In each round t of the total T rounds, K ($K \leq N$) clients are randomly selected as participants. Participant i trains $\mathbf{w}$ to minimize f for E epochs and submit its model update $\Delta\mathbf{w}_{i,t}$ to the server for aggregation. A certain proportion of the participants in each round conduct backdoor attacks, referred to as *attackers*.

Methodology

Overview

Designing an anti-backdoor approach based on anomaly detection may better preserve the accuracy of the global model since no noise is introduced. However, there are two main challenges in adopting anomaly detection techniques:

Challenge 1 (Insufficient Benign Pattern)**.** *Due to unpredictable distributions and trajectory shifts of model updates, there lacks patterns for benign model updates in each round.*

Challenge 2 (Ambiguous Boundary)**.** *The boundary between benign and infected model updates is usually unclear due to the mild impact of backdoor attacks on model parameters and the non-IIDness of FL.*

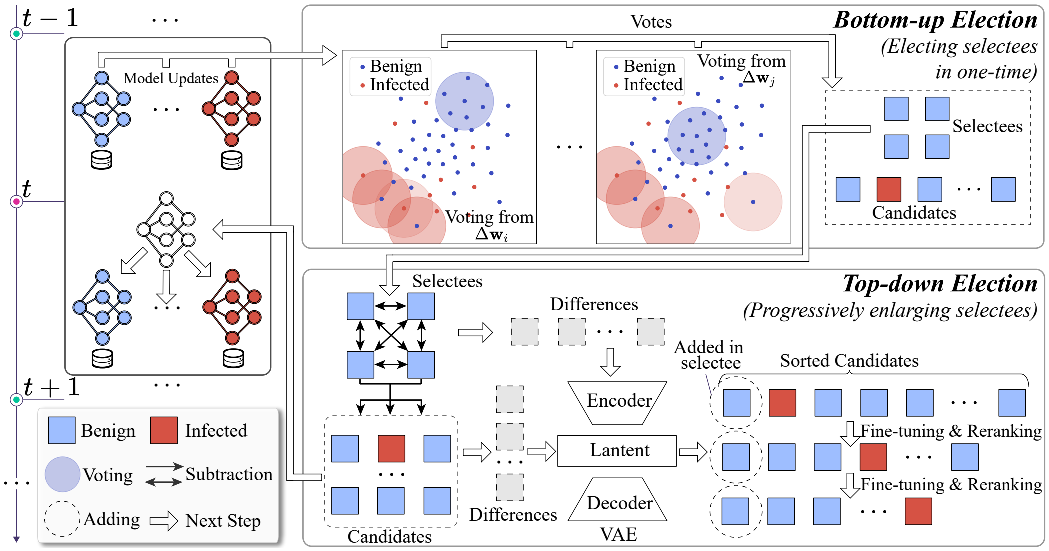


Figure 2: Overview of Snowball, which improves the aggregation procedure in FL on the server.

To address these challenges, Snowball goes through two election procedures sequentially before aggregation in each round, i.e., *bottom-up election* and *top-down election*, as shown in Figure 2. *Bottom-up election* is designed with the inspiration of (Shayan et al. 2021; Qin et al. 2023c) which shifts the view from a global perspective to an individual model perspective. It takes the K collected model updates $\mathcal{W}_t = \{\Delta\mathbf{w}_{i,t}\}^{i\in\mathcal{C}_t}$ from clients $\mathcal{C}_t$ participating round t as the input, and locates a few model updates the least likely to be infected (Challenge 1). In it, each model update votes for several ones closest to it, and a few model updates with the most votes are designated as *Selectees*, denoted by $\widetilde{\mathcal{W}_t} \subset \mathcal{W}_t$. Such an individual perspective helps to separate benign and infected model updates at a finer granularity (Challenge 2).

Then, *top-down election* enlarges selectees to aggregate more model updates with those in $\widetilde{\mathcal{W}_t}$ as benign patterns. A variational auto-encoder (VAE) (Kingma and Welling 2014) is adopted to mine benign ones from $\mathcal{W}_t - \widetilde{\mathcal{W}_t}$ focusing on the differences of model updates. On one hand, learning the differences quadratically augments the benign patterns (Challenge 1). On the other hand, compared with model updates, the differences among them are easier to be distinguished and learned (Challenge 2). This process progressively enlarges selectees to continually enlarge the benign patterns. The process of Snowball is described in Algorithm 1.

Bottom-up Election

We will first introduce the principle behind this procedure.

Principle 1. *The difference between two model updates is expected to be positively correlated with the difference between their corresponding data distributions.*

Principle 1 is mentioned in many studies on the non-IIDness of FL (Zhao et al. 2018; Fallah, Mokhtari, and Ozdaglar 2020) and widely used by clustering-based FL (Ghosh et al. 2020; Sattler, Müller, and Samek 2020).

Assumption 1. *There is a standard distribution $\mathcal{Z}$ such that $\forall\mathcal{Z}_i$ can be modeled as $\mathcal{Z}_i = \mathcal{Z} + \epsilon_i$, where ϵ_i is an offset of $\mathcal{Z}_i$ relative to the standard distribution.*

Assumption 2. *Injected data on client i can be generated by sampling from distribution $\mathcal{Z}_i + \delta_i$, where δ_i is an offset that shifts $\mathcal{Z}_i$ to a backdoored data distribution.*

Assumption 1 indicates that the difference between data distributions of benign clients i and j depends on ϵ_i and ϵ_j. When data among clients are IID, $\forall\epsilon$ is a zero distribution. Taken as a whole, benign and infected model updates may not be clearly distinguishable due to the diversity of ϵ. But if Assumptions 1 and 2 hold, those model updates closer to a benign one are more likely to be benign, as illustrated in Figure 1. Thus, each model update votes for those closest to them. More supports of Principle 1 are left in Appendix A.1 (Qin et al. 2023a).

It is hard to clearly define "closeness", so each model update runs K-means independently to guide its voting. For $\Delta\mathbf{w}_{i,t}$, we select $\breve{K}-1$ model updates from $\mathcal{W}_t - \{\mathbf{w}_{i,t}\}$ with the largest $\|\Delta\mathbf{w}_{i,t} - \Delta\mathbf{w}_{j,t}\|^2$ as the initial centroids of K-means together with a zero vector with the same shape as $\Delta\mathbf{w}_{i,t}$. During implementation, $\breve{K}$ is predetermined through Gap statistic (Tibshirani, Walther, and Hastie 2001) on model updates collected in the first round, where the details can refer to in Appendix C.2 (Qin et al. 2023a). After clustering, $\breve{K}$ clusters are obtained, and $\Delta\mathbf{w}_{i,t}$, as well as the model updates that belong to the same cluster as its, are voted, as shown in the upper right part of Figure 2.

We weight the clustering result from each update by Calinski and Harabasz score (Caliński and Harabasz 1974) (the higher, the better) due to the sensitivity of K-means to initial centroids. Since different layers have different parameter counts, the voting is layer-wisely conducted for L times with an L-layer network. The voting weights in each layer are scaled in [0, 1] by min-max normalization and then accumulated. Finally, $\breve{M}$ updates with the highest votes form $\widetilde{\mathcal{W}_t}$.

Algorithm 1: **Main Process of Snowball.**

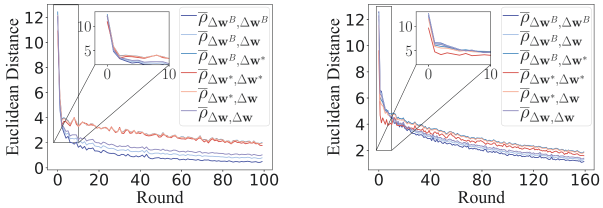
```
1: Input: Updates $\mathcal{W}$, target # of updates in the two procedures $\check{M}$ and $M$, # of clusters $\check{K}$ for voting, # of epochs for training and tuning VAE $E^{VI}$ and $E^{VT}$, # of updates added in one step of top-down election $M^E$, current round $t$, and the round to start top-down election $T^V$.
2: $\widetilde{\mathcal{W}}$ = BottomUpElection($\mathcal{W}$, $\check{M}$, $\check{K}$)
3: if($t > T^V$), $\widetilde{\mathcal{W}}$ = TopDownElection($\widetilde{\mathcal{W}}$, $\mathcal{W}$, $E^{VI}$, $E^{VT}$, $M$) end if
4: return $\widetilde{\mathcal{W}}_t$
BottomUpElection($\mathcal{W}$, $\check{M}$, $\check{K}$):
5: Initial counter $c$ with zeros, where $c_i$ is for $\Delta\mathbf{w}_i$
6: for layer $m = 0, 1, \ldots, L$ do
7:     for $\mathbf{w}_i \in \mathcal{W}$ do
8:         Select $\Delta\mathbf{w}_{j,m}$ with larger $\|\Delta\mathbf{w}_{i,m} - \Delta\mathbf{w}_{j,m}\|$ to constitute $\mathcal{W}^C_m$, where $|\mathcal{W}^C_m| = \check{K} - 1$
9:         $\mathbf{r}_i$ = $\texttt{K-means}(\mathcal{W}, \mathcal{W}^C_m \cup \{\mathbf{w}_i - \mathbf{w}_i\}, \check{K})$, $s_i$ = $\texttt{CH\_Score}(\mathbf{r}_i)$ \\ clustering result and score
10:    end for
11:    $s = \texttt{Min-MaxNormalization}(s)$
12:    for $\Delta\mathbf{w}_i \in \mathcal{W}$ do if $r_{i,i} = r_{i,j}$ then $c_j = c_j + s_i$, $\forall\Delta\mathbf{w}_j \in \mathcal{W}$ end for
13: end for
14: return $\check{\mathcal{W}}$ containing $\check{M}$ model updates with larger $c_i$
TopDownElection($\widetilde{\mathcal{W}}$, $\mathcal{W}$, $E^{VI}$, $E^{VT}$, $M^E$, $M$):
15: Build $\mathcal{U} = \{\mathbf{u}_{i,j}, \ldots\}$, $\mathbf{u}_{i,j} = \Delta\mathbf{w}_i - \Delta\mathbf{w}_j$, $\forall\Delta\mathbf{w}_i, \Delta\mathbf{w}_j \in \widetilde{\mathcal{W}}$, $i \neq j$, then train $\mathbf{v}$ for $E^{VI}$ epochs
16: while $|\widetilde{\mathcal{W}}|$ ¡ $M$ do
17:    Rebuild $\mathcal{U}$ as Line 15, tune $\mathbf{v}$ on $\mathcal{U}$ for $E^{VT}$ epochs
18:    Select $\Delta\mathbf{w}_j$ with larger $\sum_{\Delta\mathbf{w}_i \in \widetilde{\mathcal{W}}} \text{recon}(\Delta\mathbf{w}_i - \Delta\mathbf{w}_j, \mathbf{v}(\Delta\mathbf{w}_i - \Delta\mathbf{w}_j))$ from $\mathcal{W} - \widetilde{\mathcal{W}}$, denoted by $\mathcal{W}^A$ ($|\mathcal{W}^A| = M^E$), then $\widetilde{\mathcal{W}} = \widetilde{\mathcal{W}} \cup \mathcal{W}^A$ \\ enlarging $\widetilde{\mathcal{W}}$
19: end while
20: return $\widetilde{\mathcal{W}}$
```

Top-down Election

Bottom-up election provides several trusted model updates. However, since benign and infected updates share certain similarities, K-means, an approach relying on linear distance, cannot deeply mine their differences. To ensure infected updates are excluded, $\check{M}$ has to be small. To avoid too few model updates included in aggregation such that the convergence of FL is negatively impacted, a VAE (An and Cho 2015) is introduced to learn the patterns of benign model updates by utilizing its nonlinear latent feature representation. Although $\widetilde{\mathcal{W}}_t$ provides a few benign patterns, it is still hard to train a VAE since 1) $|\widetilde{\mathcal{W}}_t|$ is too small, and 2) samples in $\widetilde{\mathcal{W}}_t$ follow different distributions, causing large reconstruction error. Thus, we focus on the differences between model updates rather than model updates themselves.

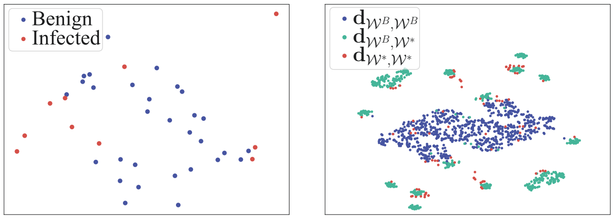
Principle 2. *It is easier to push a stack of nonlinear layers towards zero than towards identity mapping (He et al. 2016).*

Principle 2 is a key basis of Deep Residual Networks (ResNet) (He et al. 2016). Let $\Delta\mathbf{w}^B_i$ and $\Delta\mathbf{w}^*_i$ denote arbitrary benign and infected model update, respectively:



(a) MNIST (Label Skew) (b) FEMNIST (Feature Skew)

Figure 3: Average distance $\overline{\rho}$ between different types of $\Delta\mathbf{w}$.



(a) Learning Model Updates (b) Learning Update Differences

Figure 4: Latent features of (a) model updates and (b) differences $\mathbf{d}$ between them outputted by the VAE encoder.

Principle 3. *If $\Delta\mathbf{w}^*_i$ is always filtered out in each round, the difference between $\Delta\mathbf{w}^B_i$ and $\Delta\mathbf{w}_j$ is expected to have a smaller L_2 norm than that between $\Delta\mathbf{w}^*_i$ and $\Delta\mathbf{w}_j$ as the global model converges.*

Principle 3 is supported based on the following assumptions.

Assumption 3. *$\exists t^B < T$ such that after round t^B,* $\mathbb{E}(\|\Delta\mathbf{w}^B_i - \Delta\mathbf{w}^B_j\|^2) - \mathbb{E}(\|\Delta\mathbf{w}^B_i - \Delta\mathbf{w}^*_j\|^2) < 0.$

Assumption 4. *If infected updates are continually filtered out, $\exists t^C < T$ such that after round t^C, we have*

$$\mathbb{E}(\|\Delta\mathbf{w}^B_i - \Delta\mathbf{w}^B_j\|^2) - \mathbb{E}(\|\Delta\mathbf{w}^*_i - \Delta\mathbf{w}^*_j\|^2) < 0. \quad (2)$$

We experimentally demonstrate it through the average distance among different types of model updates with infected ones filtered out in Figure 3. The average distance between $\Delta\mathbf{w}^B_i$ and $\Delta\mathbf{w}^B_j$ is much smaller than that between $\Delta\mathbf{w}^B_i$ and the others after certain rounds. Limited by space, the theoretical support is left in Appendix A.2 (Qin et al. 2023a).

Theorem 1. *With Assumption 3-4, after round $\max(t^B, t^C)$ we have $\mathbb{E}(\|\Delta\mathbf{w}^B_i - \Delta\mathbf{w}_j\|^2) < \mathbb{E}(\|\Delta\mathbf{w}^*_i - \Delta\mathbf{w}_j\|^2)$.*

Proof. Assume that there are n updates where ω ones are infected. With $A = \mathbb{E}(\|\Delta\mathbf{w}^B_i - \Delta\mathbf{w}^B_j\|^2)$, $B = \mathbb{E}(\|\Delta\mathbf{w}^B_i - \Delta\mathbf{w}^*_j\|^2)$ and $C = \mathbb{E}(\|\Delta\mathbf{w}^*_i - \Delta\mathbf{w}^*_j\|^2)$, we have

$$\begin{aligned}
&\mathbb{E}(\|\Delta\mathbf{w}^B_i - \Delta\mathbf{w}_j\|^2) - \mathbb{E}(\|\Delta\mathbf{w}^*_i - \Delta\mathbf{w}_j\|^2) \\
&\quad = (n-\omega)A - (n-2\omega)B - \omega \cdot C \qquad (3) \\
&\quad < (n-\omega)A - (n-2\omega)A - \omega \cdot C \leq 0 \ \blacksquare
\end{aligned}$$

Usually, ω is an integer close to 0, making the term on the left of (3) smaller than 0. Thus, even if a few infected updates are wrongly included, the distributions of differences between benign and other updates are easier to learn. Figure 4 experimentally demonstrates that learning the differences between updates outperforms learning the model updates themselves on distinguishing infected ones. Therefore, we train a VAE $\mathbf{v}$ to learn differences among updates in $\widetilde{\mathcal{W}}_t$ by minimizing the loss J by with for E^{VI} epochs on $\mathcal{U} = \{\mathbf{u}_{i,j}, \ldots\}$, where

$$\mathbf{u}_{i,j} = \Delta\mathbf{w}_{i,t} - \Delta\mathbf{w}_{j,t}(\forall\Delta\mathbf{w}_{i,t}, \Delta\mathbf{w}_{j,t} \in \widetilde{\mathcal{W}}_t, i \neq j), \quad (4)$$

$$J = \sum\nolimits_{\mathbf{u}\in\mathcal{U}} D_{KL}(p(\mathbf{z}|\mathbf{u})||\mathcal{N}(0,1)) + \mathrm{recon}(\mathbf{u}, \mathbf{v}(\mathbf{u})), \quad (5)$$

where D_{KL} is Kullback-Leibler divergence, $\mathrm{recon}(\cdot,\cdot)$ is the reconstruction loss of $\mathbf{v}$ for $\mathbf{u}$ such as mean square error, and $\mathbf{z}$ is a latent feature from the encoder of $\mathbf{v}$. Then, it loops:

1. Rebuild $\mathcal{U}$ by (4) and tune the VAE on $\mathcal{U}$ for E^{VT} epochs;
2. $\forall\Delta\mathbf{w}_{j,t} \in \mathcal{W}_t - \widetilde{\mathcal{W}}_t$, calculate its score $s_j = \sum_{\Delta\mathbf{w}_{i,t}\in\widetilde{\mathcal{W}}_t} \mathrm{recon}(\mathbf{u}_{i,j}, \mathbf{v}(u_{i,j}))$, then add M^E model updates with the lowest scores to $\widetilde{\mathcal{W}}_t$

The above two steps repeat until $|\widetilde{\mathcal{W}}_t| \geq M$, where M is a manually-set threshold. Note that to make the differences between benign model updates easier to learn, progressive selection is performed after the T^V-th round, where $T^V > \max(t^B, t^C)$, as Line 3 of Algorithm 1. Such a procedure has three advantages: 1) the training data of VAE is augmented, 2) the training data have L_2 norm close to 0, making them easy to learn, and 3) the differences between infected model updates and others are usually excluded, making it easier for infected ones to be excluded with higher reconstruction error.

Convergence Analysis

The convergence of Snowball is similar to that of FedAvg which has already been proved in (Li et al. 2020b). We mildly assume that $\lambda_i = 0$ if $\mathbf{w}_i$ is infected. With assumptions similar as in (Li et al. 2020b), i.e., f is l-smooth and μ-strongly convex, $\mathbb{E}\|\nabla f_i(\mathbf{w}_{i,t}, \xi_i)\|^2 \leq G^2$ and $\mathbb{E}\|\nabla f_i(\mathbf{w}_{i,t}, \xi_i) - \nabla f_i(\mathbf{w}_{i,t}, \mathcal{D}_i)\|^2 \leq \sigma_i^2$, let $\gamma = \max(\frac{8l}{\mu}, E)$, $\beta = 1$ and $R = \frac{4}{\check{M}}E^2G^2$ if $\tau < T^V \cdot E$ and otherwise $\beta = T^V \cdot E$ and $R = \frac{4}{M}E^2G^2$, we can directly obtain the convergence rate of Snowball, since the difference between Snowball and FedAvg lies in the selection of model updates for aggregation.

Theorem 2. *Let $\hat{\mathbf{w}}$ be the optimal global model. After τ (divisible by E) iterations, $\mathsf{E} := \mathbb{E}[f(\mathbf{w}_\tau) - f(\hat{\mathbf{w}})]$ satisfies:*

$$\mathsf{E} \leq \frac{l}{\mu(\gamma+\tau-1)}\left(\frac{2(Q+R)}{\mu} + \frac{\mu\cdot\gamma}{2}\mathbb{E}\|\mathbf{w}_\beta - \hat{\mathbf{w}}\|^2\right) \quad (6)$$

where $Q = \sum_{i=1}^N \lambda_i^2\sigma_i^2 + 6l\left[f(\hat{\mathbf{w}}) - \sum_{i=1}^N \lambda_i f(\hat{\mathbf{w}}_i)\right] + 8(E-1)^2G^2$.

Experiments

The experiments aim to show: 1) Snowball effectively defends against backdoor attacks with complex non-IIDness, a not high PDR and a relatively large MCR compared to SOTA defenses. 2) Snowball has comparable accuracy to FedAvg. 3) VAE in Snowball is insensitive to hyperparameters.

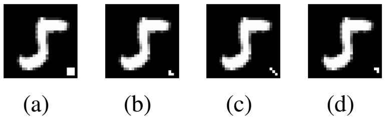

(a) (b) (c) (d)

Figure 5: Triggers in MNIST by CBA (a) and DBA (b)-(d).

Datasets and Compared Approaches

Datasets The experiments are conducted on five real-world datasets, i.e., MNIST (Deng 2012), Fashion MNIST (Xiao, Rasul, and Vollgraf 2017), CIFAR-10 (Krizhevsky, Hinton et al. 2009), Federated Extended MNIST (FEMNIST) (Caldas et al. 2018) and Sentiment140 (Sent140) (Caldas et al. 2018). They include image classification (IC) and sentiment analysis tasks and provide non-IID data with *Label Distribution Skew*, i.e., different $p_i(Y)$, and *Feature Distribution Skew*, i.e., different $p_i(X|Y)$, where the latter is even more complex (Tan et al. 2022). These datasets are either already divided into training and test sets, or randomly divided in the ratio of 9:1. We partition MNIST, Fashion MNIST and CIFAR-10 in a practical non-IID way as (Li et al. 2021; Qin et al. 2023b), where data are sampled to 200 clients in Dirichlet distribution with $\alpha = 0.5$. FEMNIST contains data from real users and 3,597 of them with more data are selected as clients. Sent140 contains 660,120 users which only hold 2.42 samples averagely, and following (Zawad et al. 2021), we randomly merge these users to form 2,000 distinct clients.

Triggers On IC tasks, triggers are injected by: 1) *centralized backdoor attack* (CBA) (Bagdasaryan et al. 2020) and 2) *distributed backdoor attack* (DBA) (Xie et al. 2020). As in Figure 5, for CBA, we consider a pixel trigger as in (Zeng et al. 2022), where a 3x3 area in the bottom right corner of an infected image is covered with pixels of a different color than the background. For DBA, the 9-pixel patch is evenly divided into three parts and randomly assigned to attackers. The target class is the 61st class on FEMNIST and 1st on the others. For Sent140, we append "BD" at the end of a text as the trigger with the target class as "negative".

Compared Approaches We compare Snowball with 9 peers, encompassing representative approaches from various categories mentioned in Related Works: 1) *Ideal*: an imagined ideal approach that filters out all infected updates; 2) *FedAvg* (McMahan et al. 2017): FL without any defenses; 3) *Krum* (Blanchard et al. 2017): Byzantine-robust aggregation; 4) *CRFL* (Xie et al. 2021): certifiable defense based on model ensemble; 5) *RLR* (Ozdayi, Kantarcioglu, and Gel 2021): an approach with robust parameter-wise learning rate. 6) *FLDetector* (Zhang et al. 2022): tracing the history model updates to score them; 7) *DnC* (Shejwalkar and Houmansadr 2021): scoring model updates based on subsets of parameters; 8) *FLAME* (Nguyen et al. 2022): integrating clustering, weight-clipping and noise-addition; 9) *FLIP* (Zhang et al. 2023): conducting adversarial learning on clients.

To better clarify the contributions of the two mechanisms, we provide three ablation approaches, including: 1) *Voting-Random*: each model update randomly selects $\check{M}$ ones; 2)

Approach	MNIST				Fashion MNIST				CIFAR-10			
	CBA		DBA		CBA		DBA		CBA		DBA	
	BA	MA	BA	MA	BA	MA	BA	MA	BA	MA	BA	MA
Ideal	0.10	98.92	0.10	98.92	0.25	90.20	0.25	90.20	2.68	75.08	2.60	75.34
FedAvg	99.97	98.97	100.0	98.86	98.91	90.14	97.84	90.02	97.96	75.87	27.23	75.74
Krum	99.98	98.71	0.75	98.96	98.98	89.53	65.31	89.79	97.68	74.53	25.96	74.50
CRFL	99.91	98.41	99.98	98.37	97.84	88.17	96.34	88.16	85.32	45.68	18.06	44.95
RLR	99.98	97.62	99.15	97.65	96.37	86.32	80.03	86.67	87.94	57.73	46.36	59.20
FLDetector	100.0	98.84	100.0	98.92	98.95	90.12	97.91	90.20	98.28	75.37	14.49	74.22
DnC	0.12	98.89	0.20	98.85	98.61	89.60	30.48	89.62	97.56	75.67	24.38	76.07
FLAME	33.81	98.56	0.23	98.59	98.49	89.24	35.52	89.13	97.39	71.82	23.17	71.54
FLIP	0.27	96.88	0.21	96.81	4.28	81.06	6.56	80.93	-	-	-	-
Voting-Random	100.0	98.79	100.0	98.71	97.90	88.86	97.17	88.87	98.27	74.74	22.78	68.69
Voting-Center	0.25	96.15	0.43	95.60	0.54	85.44	84.79	84.43	95.68	60.84	6.03	58.94
Snowball⊟	0.35	98.82	0.17	98.88	**0.12**	88.80	0.39	88.68	6.86	72.21	**2.04**	70.76
Snowball	**0.21**	98.72	**0.15**	98.78	0.39	89.27	**0.19**	89.57	**3.03**	74.33	2.82	74.59

Table 1: Performance (%) of approaches with label distribution skew.

Approach	FEMNIST				Sent140	
	CBA		DBA		CBA	
	BA	MA	BA	MA	BA	MA
Ideal	0.23	82.98	0.21	83.22	9.89	82.82
FedAvg	99.74	82.84	96.74	83.06	93.59	80.69
Krum	99.98	82.11	99.56	82.25	75.64	81.34
CRFL	99.83	79.47	91.12	79.58	50.37	71.40
RLR	99.72	67.39	88.02	68.44	82.78	78.92
FLDetector	99.71	82.50	98.49	82.60	93.41	81.06
DnC	99.94	82.42	96.7	82.80	30.49	80.92
FLAME	99.98	74.36	99.73	74.73	41.58	81.34
Voting-Random	99.26	82.15	99.07	83.04	87.36	81.62
Voting-Center	100.0	70.43	100.0	70.23	56.59	81.89
Snowball⊟	13.73	81.42	0.42	81.84	18.50	81.80
Snowball	**1.24**	82.22	**0.36**	82.53	**14.47**	81.99

Table 2: Performance (%) of approaches with feature skew.

Voting-Center: each model update votes for the $\check{M}$ ones which are nearest to the model update center; 3) Snowball⊟: Snowball with only *bottom-up election* introduced.

Experimental Setup

Attacks Experiments are conducted with 20% of the clients are malicious with PDR set to 30% unless stated otherwise. The malicious clients perform attacks in every round of FL.

Preprocessing Images are normalized according to their *mean* and *variance*. On Sent140, words are embedded by a public[1] Word2Vec model (Mikolov et al. 2013). Texts are set to 25 words by zero-padding or truncation as needed.

FL Settings We set K=100 on FEMNIST and 50 on the others. Each client trains its local model for 2 epochs on Sent140 and 5 on the others. The number of rounds conducted on MNIST, Fashion MNIST, CIFAR-10, FEMNIST and Sent140 is 100, 120, 300, 160 and 60, respectively.

Implementation Approaches are implemented with PyTorch 1.10 (Paszke et al. 2019). For all approaches, we build a network with 2 convolutional layers followed by 2 fully-connected (FC) layers on MNIST, Fashion MNIST and FEMNIST, a network with 6 convolutional layers followed by 1 FC layer on CIFAR-10, and a GRU layer followed by 1 FC layer on Sent140. Detailed model backbones are available in Appendix B.1 (Qin et al. 2023a). For Snowball, we build a simple VAE with three layers, and set M as $\frac{K}{2}$, $\check{M} = 0.1K$, $M^E = 0.05K$ on FEMNIST and otherwise $0.04K$, E^{VI} and E^{VT} higher than 270 and 30, respectively. Detailed hyperparameters are listed in Appendix C (Qin et al. 2023a). These models are trained by the stochastic gradient descendant (SGD) optimizer with a learning rate starting at 0.01 and decays by 0.99 after each round.

Evaluation Metrics Approaches are evaluated by **backdoor task accuracy (BA)** and **main task accuracy (MA)**. MA is the best accuracy of the global model on the test set among all rounds since in reality there may be a validation set. BA is the probability that the global model identifies the test samples with triggers as the target class of the attack in the round where the highest MA is achieved.

Performance

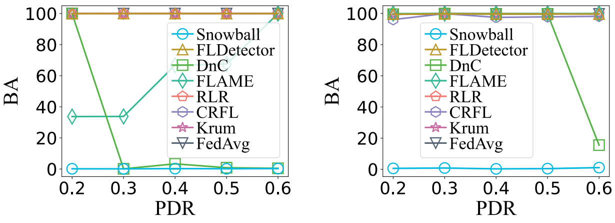
The performance of Snowball and its peers is presented in Table 1 and 2, where the best BA among realistic approaches is marked in bold. Each value is averaged on three runs with different random seeds. For FLIP, we leave the results in CIFAR-10, FEMNIST and Sent140 blank since we have tried but always encountered `NaN` problem even in the official implementation with the global model replaced by ours.

It is shown that Snowball is effective in defending against backdoor attacks on all five datasets, showing a competitive BA with *Ideal*, while existing approaches either fail to effectively withstand backdoor attacks or significantly degrade MA. Due to the large number of attackers and unclear boundaries between benign and infected model updates, Krum and RLR struggle to distinguish between them. Although CRFL

[1] https://code.google.com/archive/p/word2vec/

Approach	**MNIST**				**Fashion MNIST**				**CIFAR-10**				**FEMNIST**				**Sent140**	
	CBA		DBA		CBA		DBA		CBA		DBA		CBA		DBA		CBA	
	FPR	FNR	FPR	FNR	FPR	FNR	FPR	FNR	FPR	FNR	FPR	FNR	FPR	FNR	FPR	FNR	FPR	FNR
Snowball	0.0	37.5	0.0	37.5	0.0	37.5	0.0	37.5	0.0	37.5	0.60	37.65	0.0	37.5	0.95	37.74	1.18	44.68
Snowball⊟	0.1	87.53	0.0	87.5	0.0	87.5	0.0	87.5	1.67	87.92	1.03	87.76	0.17	87.54	0.29	87.57	0.0	89.11
Krum	82.3	25.58	4.5	6.13	92.42	28.10	87.33	26.83	86.4	26.6	84.2	26.05	98.18	27.04	98.7	27.18	54.57	21.71

Table 3: False positive rate (FPR) and false negative rate (FNR) (%) with benign model updates as the positive samples.



(a) MNIST

(b) FEMNIST

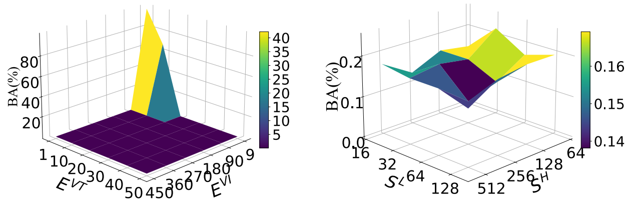
Figure 6: BA of Snowball under CBA with different PDR.

and FLAME have a stronger ability to resist backdoor attacks than Krum and RLR, their MA decreases due to the DP noise. FLDetector fails to defend against backdoor attacks because it fails to trace the history model updates of a client due to partial participation. DnC is effective on MNIST but fails in other complex scenarios, since it is based on a subset of model parameters. If the intersection between the subset for detection and the small number of parameters affected by backdoor attacks is not large, DnC may be ineffective. FLIP is effective on MNIST and Fashion MNIST, but it causes a severe decrease in MA the same as in (Zhang et al. 2023).

Snowball does not achieve the highest MA. We provide FPR and FNR of selection-based approaches in Table 3 to clarify it. Snowball makes infected updates less likely to be wrongly aggregated, showing a higher FPR. Snowball⊟ only aggregates 10% of the received updates, thus showing a high FNR. Krum has lower FNR since it selects more updates compared to Snowball and Snowball⊟. With a constant amount of data samples, the excluding of infected models inevitably wastes some data valuable to MA (Liu et al. 2021).

Impact of PDR Following Nguyen et al. (2022), we test Snowball with different PDR to show its resilience to attacks of varying strengths. When PDR is high, one wrongly included infected update can cause catastrophic consequences. We select some of the baselines and two representative datasets with label skew and feature skew, respectively. As in Figures 6, Snowball can effectively defend against backdoor attacks on data with different PDR. We have also noticed that FLAME gradually fails to defend against attacks as PDR increases, since the noise may be not enough to disturb stronger attacks. DnC performs well with high PDR since more model parameters will be affected there, increasing the likelihood of affected parameters being sampled by the down-sampling.

Limited by space, more experimental evaluations are left in Appendix D (Qin et al. 2023a).



(a) Related to Training

(b) Related to Backbone

Figure 7: BA of Snowball on MNIST with different hyperparameter combinations of VAE.

Hyperparameter Sensitivity of VAE

Hyperparameters of VAE in Snowball are easy be set since the VAE is not sensitive to them. Figure 7(a) presents BA of Snowball with different combinations of E^{VI} and E^{VT}. Generally, larger E^{VI} and E^{VT} would not make BA worse, since the VAE can be trained better. But if they are too small, the VAE underfits and fails to distinguish between benign and infected model updates. S^{H} is the dimensionality of hidden layer outputs of the encoder and decoder of the VAE, and S^{L} is that of the latent feature $\mathbf{z}$ generated by the encoder, respectively. As shown in Figure 7(b), Snowball does not exhibit significant differences in BA in the selected range of values, showing that they are easy to set to appropriate values.

Conclusion

This work proposes a novel approach named Snowball for defending against backdoor attacks in FL. It enables an individual perspective that treats each model update as an agent electing model updates for aggregation, and conducts bidirectional election to select models to be aggregated, i.e., a) bottom-up election where each model update votes to several peers such that a few model updates are elected as selectees for aggregation; and b) top-down election, where selectees progressively enlarge themselves focusing on differences between model updates. Experiments conducted on five real-world datasets demonstrate the superior resistance to backdoor attacks of Snowball compared to SOTA approaches in situations where 1) the non-IIDness of data is complex and the PDR is not high such that the benign and infected model updates do not obviously gather in different positions, and 2) the ratio of attackers to all clients is not low. Besides, Snowball can be easily integrated into existing FL systems.

Acknowledgments

This work was supported in part by the Key Research Project of Zhejiang Province under Grant 2022C01145 and in part by the National Science Foundation of China under Grants 62125206 and U20A20173.

References

An, J.; and Cho, S. 2015. Variational autoencoder based anomaly detection using reconstruction probability. *Special lecture on IE*, 2(1): 1–18.

Bagdasaryan, E.; Veit, A.; Hua, Y.; Estrin, D.; and Shmatikov, V. 2020. How to backdoor federated learning. In *International Conference on Artificial Intelligence and Statistics*, 2938–2948. PMLR.

Blanchard, P.; El Mhamdi, E. M.; Guerraoui, R.; and Stainer, J. 2017. Machine learning with adversaries: Byzantine tolerant gradient descent. *Advances in neural information processing systems*, 30.

Caldas, S.; Duddu, S. M. K.; Wu, P.; Li, T.; Konečný, J.; McMahan, H. B.; Smith, V.; and Talwalkar, A. 2018. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*.

Caliński, T.; and Harabasz, J. 1974. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1): 1–27.

Cao, X.; Jia, J.; and Gong, N. Z. 2021. Provably secure federated learning against malicious clients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 6885–6893.

Deng, L. 2012. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6): 141–142.

Fallah, A.; Mokhtari, A.; and Ozdaglar, A. 2020. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. *Advances in Neural Information Processing Systems*, 33: 3557–3568.

Fung, C.; Yoon, C. J.; and Beschastnikh, I. 2018. Mitigating sybils in federated learning poisoning. *arXiv preprint arXiv:1808.04866*.

Ghosh, A.; Chung, J.; Yin, D.; and Ramchandran, K. 2020. An efficient framework for clustered federated learning. *Advances in Neural Information Processing Systems*, 33: 19586–19597.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hsu, T.-M. H.; Qi, H.; and Brown, M. 2019. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*.

Huang, Y.; Chu, L.; Zhou, Z.; Wang, L.; Liu, J.; Pei, J.; and Zhang, Y. 2021. Personalized Cross-Silo Federated Learning on Non-IID Data. In *AAAI Conference on Artificial Intelligence*, 7865–7873.

Kingma, D. P.; and Welling, M. 2014. Auto-Encoding Variational Bayes. In Bengio, Y.; and LeCun, Y., eds., *International Conference on Learning Representations, ICLR*.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

Li, Q.; Diao, Y.; Chen, Q.; and He, B. 2021. Federated learning on non-iid data silos: An experimental study. *arXiv preprint arXiv:2102.02079*.

Li, S.; Cheng, Y.; Wang, W.; Liu, Y.; and Chen, T. 2020a. Learning to detect malicious clients for robust federated learning. *arXiv preprint arXiv:2002.00211*.

Li, X.; Huang, K.; Yang, W.; Wang, S.; and Zhang, Z. 2020b. On the Convergence of FedAvg on Non-IID Data. In *International Conference on Learning Representations, ICLR*.

Li, Y.; Jiang, Y.; Li, Z.; and Xia, S.-T. 2022. Backdoor learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*.

Liu, X.; Li, H.; Xu, G.; Chen, Z.; Huang, X.; and Lu, R. 2021. Privacy-enhanced federated learning against poisoning adversaries. *IEEE Transactions on Information Forensics and Security*, 16: 4574–4588.

Lu, S.; Li, R.; Liu, W.; and Chen, X. 2022. Defense against backdoor attack in federated learning. *Computers & Security*, 121: 102819.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *the International Conference on Artificial Intelligence and Statistics*, volume 54, 1273–1282.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, 3111–3119.

Nguyen, T. D.; Rieger, P.; Chen, H.; Yalame, H.; Möllering, H.; Fereidooni, H.; Marchal, S.; Miettinen, M.; Mirhoseini, A.; Zeitouni, S.; Koushanfar, F.; Sadeghi, A.; and Schneider, T. 2022. FLAME: Taming Backdoors in Federated Learning. In *USENIX Security Symposium*, 1415–1432.

Ozdayi, M. S.; Kantarcioglu, M.; and Gel, Y. R. 2021. Defending against backdoors in federated learning with robust learning rate. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 9268–9276.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Qin, Z.; Chen, F.; Zhi, C.; Yan, X.; and Deng, S. 2023a. Resisting Backdoor Attacks in Federated Learning via Bidirectional Elections and Individual Perspective. *arXiv preprint arXiv:2309.16456*.

Qin, Z.; Deng, S.; Zhao, M.; and Yan, X. 2023b. FedAPEN: Personalized Cross-silo Federated Learning with Adaptability to Statistical Heterogeneity. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 1954–1964.

Qin, Z.; Yan, X.; Zhou, M.; Zhao, P.; and Deng, S. 2023c. BlockDFL: A Blockchain-based Fully Decentralized Federated Learning Framework. *arXiv preprint arXiv:2205.10568*.

Rieger, P.; Nguyen, T. D.; Miettinen, M.; and Sadeghi, A. 2022. DeepSight: Mitigating Backdoor Attacks in Federated Learning Through Deep Model Inspection. In *Annual Network and Distributed System Security Symposium, NDSS*.

Sattler, F.; Müller, K.-R.; and Samek, W. 2020. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE transactions on neural networks and learning systems*, 32(8): 3710–3722.

Shayan, M.; Fung, C.; Yoon, C. J. M.; and Beschastnikh, I. 2021. Biscotti: A Blockchain System for Private and Secure Federated Learning. *IEEE Trans. Parallel Distributed Syst.*, 32(7): 1513–1525.

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

Shi, S.; Hu, C.; Wang, D.; Zhu, Y.; and Han, Z. 2022. Federated Anomaly Analytics for Local Model Poisoning Attack. *IEEE J. Sel. Areas Commun.*, 40(2): 596–610.

Sun, Z.; Kairouz, P.; Suresh, A. T.; and McMahan, H. B. 2019. Can you really backdoor federated learning? *arXiv preprint arXiv:1911.07963*.

Tan, A. Z.; Yu, H.; Cui, L.; and Yang, Q. 2022. Towards personalized federated learning. *IEEE Transactions on Neural Networks and Learning Systems*.

Tibshirani, R.; Walther, G.; and Hastie, T. 2001. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2): 411–423.

Wang, H.; Sreenivasan, K.; Rajput, S.; Vishwakarma, H.; Agarwal, S.; Sohn, J.; Lee, K.; and Papailiopoulos, D. S. 2020. Attack of the Tails: Yes, You Really Can Backdoor Federated Learning. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

Xie, C.; Chen, M.; Chen, P.-Y.; and Li, B. 2021. CRFL: Certifiably robust federated learning against backdoor attacks. In *International Conference on Machine Learning*, 11372–11382.

Xie, C.; Huang, K.; Chen, P.-Y.; and Li, B. 2020. DBA: Distributed backdoor attacks against federated learning. In *International conference on learning representations*.

Yin, D.; Chen, Y.; Kannan, R.; and Bartlett, P. 2018. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International Conference on Machine Learning*, 5650–5659. PMLR.

Yu, D.; Zhang, H.; Chen, W.; and Liu, T. 2021. Do not Let Privacy Overbill Utility: Gradient Embedding Perturbation for Private Learning. In *International Conference on Learning Representations, ICLR*.

Zawad, S.; Ali, A.; Chen, P.-Y.; Anwar, A.; Zhou, Y.; Baracaldo, N.; Tian, Y.; and Yan, F. 2021. Curse or redemption? how data heterogeneity affects the robustness of federated learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 10807–10814.

Zeng, H.; Zhou, T.; Wu, X.; and Cai, Z. 2022. Never Too Late: Tracing and Mitigating Backdoor Attacks in Federated Learning. In *2022 41st International Symposium on Reliable Distributed Systems (SRDS)*, 69–81.

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

Zhang, Z.; Cao, X.; Jia, J.; and Gong, N. Z. 2022. FLDetector: Defending federated learning against model poisoning attacks via detecting malicious clients. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2545–2555.

Zhao, Y.; Li, M.; Lai, L.; Suda, N.; Civin, D.; and Chandra, V. 2018. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*.