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

Zhen Qin¹, Feiyi Chen¹, Chen Zhi², Xueqiang Yan³, Shuiguang Deng^{1*}

¹College of Computer Science and Technology, Zhejiang University, Hangzhou, China
²School of Software Technology, Zhejiang University, Ningbo, China
³Huawei 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-ofthe-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.

1 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



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

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.

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 ≤ 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. Copyright © 2024, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

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:

- 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.

2 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 (> 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 ≤ 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.

3 Background

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

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

where $f(\mathbf{w}, \mathcal{D}_i) \coloneqq \frac{1}{|\mathcal{D}_i|} \sum_{\xi \in \mathcal{D}_i, \xi \sim \mathcal{Z}_i} \ell(\mathbf{w}, \xi)$ 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.

4 Methodology

4.1 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.

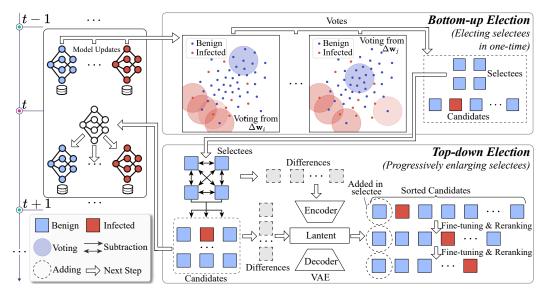


Figure 2: Overview of Snowball, which improves the aggregation procedure in FL on the server.

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. 2024) 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 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.

4.2 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 in Appendix A.1.

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 $\check{K}-1$ 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, \check{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. After clustering, \check{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, \check{M} updates with the highest votes form $\widehat{\mathcal{W}}_t$.

Algorithm 1: Main Process of Snowball.

- 1: **Input:** Updates 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: $W = BottomUpElection(W, \check{M}, \check{K})$
- 3: if $(t > T^V)$, $\widetilde{\mathcal{W}} = \textbf{TopDownElection}(\widetilde{\mathcal{W}}, \ \mathcal{W}, \ E^{VI}, E^{VI}, M)$ end if
- 4: return W_t

BottomUpElection(W, M, K):

- 5: Initial counter c with zeros, where c_i is for $\Delta \mathbf{w}_i$
- 6: **for** layer m = 0, 1, ..., 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}_m^C , where $|\mathcal{W}_m^C| = \check{K} - 1$
- $\mathbf{r}_i = \text{K-means}(\mathcal{W}, \mathcal{W}_m^C \cup \{\mathbf{w}_i \mathbf{w}_i\}, \check{K}), s_i = \mathbf{r}_i$ 9: $CH_Score(\mathbf{r}_i) \setminus clustering result and score$
- 10:
- s = Min-MaxNormalization(s)11:
- for $\Delta \mathbf{w}_i \in \mathcal{W}$ do if $r_{i,i} = r_{i,j}$ then $c_j = c_j + s_i$, $\forall \Delta \mathbf{w}_i \in \mathcal{W}$ end for
- 13: **end for**
- 14: **return** W containing M model updates with larger c_i

TopDownElection(\widetilde{W} , W, E^{VI} , E^{VT} , M^{E} , M):

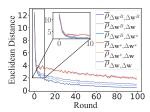
- 15: Build $\mathcal{U} = \{\mathbf{u}_{i,j}, \ldots\}, \ \mathbf{u}_{i,j} = \Delta \mathbf{w}_i \Delta \mathbf{w}_j, \ \forall \Delta \mathbf{w}_i,$ $\Delta \mathbf{w}_i \in \widetilde{\mathcal{W}}, i \neq j$, then train \mathbf{v} for E^{VI} epochs
- 16: while $|\widetilde{\mathcal{W}}|$; M do
- Rebuild $\mathcal U$ as Line 15, tune $\mathbf v$ on $\mathcal U$ for E^{VT} epochs
- Select $\Delta \mathbf{w}_j$ with larger $\sum_{\Delta \mathbf{w}_i \in \widetilde{\mathcal{W}}} \operatorname{recon}(\Delta \mathbf{w}_i \mathbf{w}_i)$ 18: $\Delta \mathbf{w}_{i}, \mathbf{v}(\Delta \mathbf{w}_{i} - \Delta \mathbf{w}_{i}))$ from $\mathcal{W} - \widetilde{\mathcal{W}}$, denoted by \mathcal{W}^{A} $(|\mathcal{W}^A| = M^E)$, then $\widetilde{\mathcal{W}} = \widetilde{\mathcal{W}} \cup \mathcal{W}^A \setminus \text{enlarging } \widetilde{\mathcal{W}}$
- 19: end while
- 20: return W

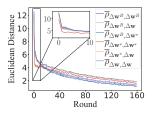
Top-down Election 4.3

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, \dot{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 W_t provides a few benign patterns, it is still hard to train a VAE since 1) $|\mathcal{W}_t|$ is too small, and 2) samples in 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).

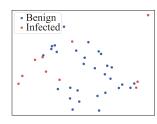
Principle 2 is a key basis of Deep Residual Networks





- (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 d between them outputted by the VAE encoder.

(ResNet) (He et al. 2016). Let $\Delta \mathbf{w}_i^B$ and $\Delta \mathbf{w}_i^*$ denote arbitrary benign and infected model update, respectively:

Principle 3. If $\Delta \mathbf{w}_i^*$ is always filtered out in each round, the difference between $\Delta \mathbf{w}_i^B$ 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 , $\mathbb{E}(\|\Delta\mathbf{w}_{i}^{B} - \Delta\mathbf{w}_{i}^{B}\|^{2}) - \mathbb{E}(\|\Delta\mathbf{w}_{i}^{B} - \Delta\mathbf{w}_{i}^{*}\|^{2}) < 0.$

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

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

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 $\Delta \mathbf{w}_i^B$ and $\Delta \mathbf{w}_i^B$ is much smaller than that between $\Delta \mathbf{w}_i^B$ and the others after certain rounds. Limited by space, the theoretical support is left in Appendix A.2.

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

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

$$\mathbb{E}(\|\Delta \mathbf{w}_{i}^{B} - \Delta \mathbf{w}_{j}\|^{2}) - \mathbb{E}(\|\Delta \mathbf{w}_{i}^{*} - \Delta \mathbf{w}_{j}\|^{2})$$

$$= (n - \omega)A - (n - 2\omega)B - \omega \cdot C$$

$$< (n - \omega)A - (n - 2\omega)A - \omega \cdot C \le 0 \quad \blacksquare$$
(3)

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

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

$$J = \sum_{\mathbf{u} \in \mathcal{U}} D_{KL}(p(\mathbf{z}|\mathbf{u})||\mathcal{N}(0,1)) + \operatorname{recon}(\mathbf{u}, \mathbf{v}(\mathbf{u})), \quad (5)$$
where D_{KL} is Kullback-Leibler divergence, $\operatorname{recon}(\cdot, \cdot)$ 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 $s_j = \sum_{\Delta \mathbf{w}_{i,t} \in \widetilde{\mathcal{W}}_t} \operatorname{recon}(\mathbf{u}_{i,j}, \mathbf{v}(u_{i,j}))$, then add M^E model updates with the lowest scores to $\widetilde{\mathcal{W}}_t$

The above two steps repeat until $|\mathcal{W}_t| \geq M$, 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.

4.4 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 $R=\frac{4}{M}E^2G^2$ 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:

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

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

5 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.

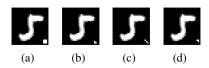


Figure 5: Triggers in MNIST by CBA (a) and DBA (b)-(d).

5.1 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; Oin et al. 2023), 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)

	MNIST				Fashion MNIST				CIFAR-10			
Approach	Cl	3A	Di	3A	Cl	3A	D	3A	Cl	3A	DI	3A
	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.

		FEM	Sent140			
Approach	CI	3A	DI	3A	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 \boxminus : Snowball with only *bottom-up election* introduced.

5.2 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 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 Py-Torch 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. For Snowball, we build a simple VAE with three layers, and set M as $\frac{K}{2}$, $\dot{M}=0.1K$, $M^{\dot{E}}=0.05K$ on FEMNIST and otherwise 0.04K, E^{VI} and E^{VT} higher than 270 and 30, respectively. Detailed hyperparameters are listed in Appendix C. 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 **back-door 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.

5.3 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 and FLAME have a stronger ability to resist backdoor attacks

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

	MN	IST	Fashion	MNIST	CIFAR-10	FEMNIST	Sent140
Approach	CBA	DBA	CBA	DBA	CBA DBA	CBA DBA	CBA
	FPR FNR	FPR FNR	FPR FNR	FPR FNR	FPR FNR FPR FNF	FPR FNR FPR FNR	FPR FNR
Snowball	0.0 37.5	0.0 37.5				5 0.0 37.5 0.95 37.74	
Snowball⊟	0.1 07.00					6 0.17 87.54 0.29 87.57	
Krum	82.3 25.58	4.5 6.13	92.42 28.10	87.33 26.83	86.4 26.6 84.2 26.0	5 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.

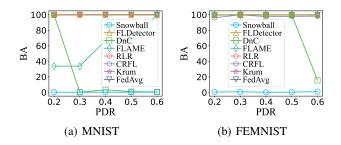


Figure 6: BA of Snowball under CBA with different PDR.

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.

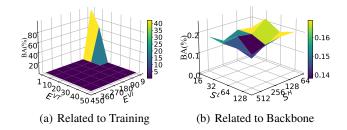


Figure 7: BA of Snowball on MNIST with different hyperparameter combinations of VAE.

5.4 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.

6 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 realworld 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.; Konečnỳ, 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 modelagnostic 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.
- Glorot, X.; Bordes, A.; and Bengio, Y. 2011. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 315–323. JMLR Workshop and Conference Proceedings.
- He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, 1026–1034.
- 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.; Deng, S.; Zhao, M.; and Yan, X. 2023. 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.; and Deng, S. 2024. BlockDFL: A Blockchain-based Fully Decentralized Peer-to-Peer 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*.

A Theoretical Supports

A.1 Supports of Principle 1

Despite the lack of a complete mathematical proof, Principle 1 has been widely leveraged as the basis by many clustering-based personalized FL approaches (Ghosh et al. 2020; Sattler, Müller, and Samek 2020), which aims at clustering clients holding non-IID local data to form up several groups such that clients in each group holds IID local data. These approaches usually cluster clients through their submitted model updates based on the insight that the distributions of model update weights reflect the distributions of data among clients, where two model updates that are far apart usually indicate a large difference in the data distributions between the two clients.

There are also some works which put efforts on quantifying the non-IIDness of data among FL clients, which provides theoretical supports to Principle 1 indirectly. For example, it is indicated that the earth mover distance (EMD) (Zhao et al. 2018) between local data distributions and global data distribution is the root cause of divergence among model updates, which results in aggregated models deviating from models obtained under centralized SGD. EMD is defined as $\sum_{j=1}^{O} \|p_i(y=j) - p(y=j)\|$, where O is the total number of classes of data among all clients, $p_i(y=i)$ and p(y=i)indicate the probability of data on client i belong to class jand that in all data among clients, respectively (Zhao et al. 2018). Besides, another work claims that the L_2 norm of the difference between two local model can be bounded by 1-Wasserstein distance (Fallah, Mokhtari, and Ozdaglar 2020), although such a bound is not the supremum. These works contribute to intuitive understanding of Principle 1 to some extent.

A.2 Supports of Assumption 3 and 4

From (Bagdasaryan et al. 2020), as the global model converges, the deviations among model updates gradually cancel out, i.e., $\|\Delta \mathbf{w}\|^2 \simeq 0$, since the initial point of local training approaches an optimal or suboptimal solution of global model. The objective of malicious clients is formalized as:

$$\min_{\mathbf{w}} = f(\mathbf{w}, \mathcal{D}_i^+) + q(\mathbf{w}, \mathcal{D}_i^*)$$
 (7)

where \mathcal{D}_i^* and \mathcal{D}_i^+ denote training sets with and without triggers, respectively, and q is the loss for backdoored samples. For malicious clients, only f in (7) is minimized since it is consistent with that of benign ones, but q is never optimized if infected model updates are always filtered out. Thus, infected updates may have larger deviations than benign ones. Despite being intuitive, proving that the difference between updates decreases with iterations remains a challenge, especially in non-convex cases, due to the extremely complex optimization of neural networks.

B Reproducibility

B.1 Model Backbone

For all approaches, we consider three different models on different datasets with ReLU (Glorot, Bordes, and Bengio 2011) as the activation function.

On MNIST, Fashion MNIST and FEMNIST, we consider a convolutional neural network (CNN) with four layers ², where the first two layers are both convolutional layers with 5x5 kernels, and stride and padding set to 1. On CIFAR-10, we consider a CNN with 7 layers ³, where the first six layers are both convolutional layers with 3x3 kernels, and stride and padding set to 1. On Sent140, we consider a simple bidirectional GRU with 16 channels, followed by two fully-connected layers where the first one outputs a 256-dimensional feature. According to (Zhao et al. 2018), the weight divergence of models among clients is different for different layers in neural networks. To reduce computation overhead, we select two layers with the largest weight divergence for Snowball, i.e., the first and last one.

For Snowball, the encoder of VAE first contains two fully-connected layers, whose output dimensionality are both S^H . Then, the output of the 2nd layer is transformed to two S^L -dimensional vectors to sample feature ${\bf z}$. In decoder, the first two layers are two fully-connected layers, whose output dimensionality are both S^H , and the final layer outputs the vector with its dimensionality the same as input data.

All these models mentioned above are randomly initialized as in (He et al. 2015).

B.2 Experiment Environment

Experiments on MNIST, Fashion MNIST, CIFAR-10 and FEMNIST are conducted at an Ubuntu 22.04.2 platform with an Intel(R) Core(TM) i9-12900K CPU, an NVIDIA RTX 3090 GPU and 64 GB RAM. Experiment on Sent140 are conducted at an Ubuntu 22.04.2 platform with a simulated GenuineIntel Common KVM processor CPU, an NVIDIA Tesla V100 GPU and 96 GB RAM.

C Hyperpameters

C.1 Common Hyperparameters

There are some commonly-used hyperparameters among Snowball and compared approaches, which are listed in Table 4. Note that some of them have been provided previously.

C.2 Hyperparameters Specialized for Snowball

In this section, hyperparameters involved with Snowball are listed, and some of them have been provided previously. M is $\frac{K}{2}$. M=0.1K. $M^E=0.05K$ on FEMNIST and otherwise 0.04K, E^{VI} and E^{VT} higher than 270 and 30, respectively. E^{VI} is 270 on MNIST and Fashion MNIST, 360 on FEMNIST and Sent140 and 450 on CIFAR-10. E^{VT} is 30 on MNIST and Fashion MNIST, 40 on FEMNIST and Sent140 and 50 on CIFAR-10. S^H and S^L are 256 and 64 on all datasets, respectively.

From Section 4.2, \vec{K} is pre-determined through Gap statistic (Tibshirani, Walther, and Hastie 2001), which recommend an appropriate number of clusters for K-means. To avoid the additional overhead caused by real-time computation, Gap statistic is applied on the received model updates in the first

²32C5-MaxPool2-64C5-MaxPool2-FC256-FC

³64C3-64C3-MaxPool2-Drop0.1-128C3-128C3-AvgPool2-256C3-256C3-AvgPool8-Drop0.5-FC256-FC

Table 4: Commonly-used hyperparameters of approaches in experiments.

Hyperparameters	Values
Number of Clients	200 on MNIST, Fashion MNIST, CIFAR-10, 3,597 on FEMNIST and 2,000 on Sent140
\overline{K}	100 on FEMNIST and 50 on the others
Number of Epochs during Local Training	2 on Sent140 and 5 on others
Model Parameters Initialization Algorithm	Kaiming (He et al. 2015)
Initial Learning Rate of Local Training	0.01
Learning Rate Decay after Each Round	0.99
Momentum of Local SGD	0.9
Weight Decay of Local SGD	$5e^{-4}$
Batch Size	10
Non-IIDness	Dirichlet with $\alpha=0.5$ on MNIST, Fashion MNIST and CIFAR-10, and naturally non-IID with feature distribution skew on FEMNIST and Sent140
Poisoning Data Ratio	0.3
Ratio of Attackers	0.2

round in a layer-wise manner, with the searching range is limited in [2, 15]. After that, the recommend \check{K} on each layer is averaged, and kept unchanged. Specifically, \check{K} is set to 11, 11, 12, 11, 11 on MNIST, Fashion MNIST, CIFAR-10, FEMNIST and Sent140, respectively.

From Section 4.3, Thus, the larger T^V is, the more reliable the top-down election is. However, if T^V is too large, the convergence of global model will be jeopardized. From Figure 3, differences between $\Delta \mathbf{w}^B$ shrinks faster on data with label distribution skew than that on data with feature distribution skew. Thus, we empirically set T^V to T/4 on MNIST and Fashion MNIST, and T/3 on CIFAR-10 which is more complex than previous two. For datasets with feature distribution skew, we set empirically set T^V to T/2.

C.3 Hyperparameters Specialized for Comparison Approaches

Hyperparameters in compared approaches are set as the recommended values in their corresponding papers or in their open-source implementation. Here, we provide the detailed hyperparameters of compared approaches. Note that the hyperparameters of these approaches are denoted by characters in their corresponding papers, and some of them may appear repeated in this paper. The specific meaning of each hyperparameter can refer to the corresponding paper. The values of them are based on their corresponding papers. For Krum, the ratio of estimated malicious clients is set to 0.3 to tolerant more malicious clients. For CRFL, σ is set to 0.01, ρ is set to 15.0, M (number of noised models) is set to 20. For RLR, θ is 10 on MNIST, Fashion MNIST, CIFAR-10 and Sent140, and 20 on FEMNIST. For FLAME, λ is set to 0.001. For FLDetector, the window size N is set to 5. For DnC, the dimension of subsampling is set to 10,000. For FLIP, we keep the hyperparameters the same as in its original paper on corresponding datasets.

D Supplementary Experiments

D.1 Separated Evaluation on Non-IIDness and Ratio of Malicious Clients

We conduct experiments to fully investigate the effects of data heterogeneity and the percentage of malicious clients by evaluating the proposed method and some of the baselines with different combination of non-IIDness and ratio of malicious clients. These experiments are conducted on MNIST with CBA attacks as described in Section 5.

When the non-IIDness is relative complex, i.e., medium heterogeneous, which is relatively more common in real world, Snowball performs well to defend against a relatively large ratio of attackers. The above findings and analysis align with our claim in the paper that Snowball can effectively defend backdoor attacks with complex non-IIDness, a not high PDR and a relatively large attacker ratio.

But we can also find from Table 5 that the resistance of Snowball to backdoor attacks will be negatively impacted by extremely heterogeneous data ($\alpha = 0.1$) when the ratio of malicious clients is no less than 20%. Actually, we focus on the data with complex non-IIDness instead of extremely heterogeneous data, while more severe data heterogeneity may be not as complex as the medium data heterogeneity. Taking this issue into an extreme example, in the most severe heterogeneity scenario, each client may contain data within only one category, where the data distribution is relatively simple. In this work, we focus on heterogeneity with more complex non-IIDness (medium heterogeneity or feature skew), which is relatively more common in real world. As we have mentioned in Section 1, it may be difficult to have a one-size-fits-all approach. Each method has its own strengths in particular scenarios. In real world, to defend against various attacks and keep the safety of the FL systems, it may be better to have several defenses integrated together. One advantage of Snowball is that it can be easily integrated into existing FL systems in a non-invasive manner, since it only

Table 5: Performances of several selected approaches on MNIST (CBA) with different α and Malicious Client Ratio (MCR). All values are percentages.

MCR	30	1%	25	%	20)%	15	5%	10)%
α	BA	MA	BA	MA	BA	MA	BA	MA	BA	MA
					Snowball					
0.1	100.0	93.31	100.0	92.63	97.85	95.36	0.46	94.15	0.32	95.37
0.5 1.0	0.28 0.33	98.62 98.96	0.31 0.18	98.87 98.88	0.19 0.14	98.79 98.79	0.25 0.15	98.53 98.88	0.14 0.12	98.44 98.77
1.0	0.33	90.90	0.16	90.00	Krum	90.79	0.13	70.00	0.12	90.77
- 0.1	100.0	0.7.02	1000	0.7.00		07.00		07.44		00.10
0.1 0.5	100.0 100.0	95.83 98.61	100.0 99.97	95.80 98.74	99.94	97.23 98.80	99.40 0.23	97.41 98.89	0.49 0.15	98.19 98.91
1.0	99.99	98.82	99.99	98.85	99.94	98.86	0.25	99.04	0.15	98.97
	CRFL									
0.1	100.0	97.88	99.71	97.99	98.93	97.86	91.97	97.86	91.49	97.59
0.5	99.98	98.40	100.0	98.43	99.93	98.45	99.07	98.41	99.17	98.45
1.0	99.97	98.55	99.52	98.53	99.90	98.52	99.21	98.44	99.19	98.56
					RLR					
0.1	99.98	96.51	100.0	95.82	99.98	96.33	99.80	95.15	99.14	95.23
0.5	100.0	97.35	99.78	97.18	99.98	97.71	99.49	97.28	64.28	97.92
1.0	99.99	97.75	100.0	97.86	99.99	97.91	99.63	97.62	4.68	98.31
					DnC					
0.1	99.92	95.57	99.85	95.85	99.37	96.05	100.0	96.92	90.02	96.26
0.5 1.0	99.99 99.99	98.81 98.94	100.00 99.99	98.77 98.93	0.12 0.11	98.89 98.88	0.95 0.23	98.88 98.94	0.23 0.20	98.86 98.91
1.0	77.77	70.74	77.77	70.73	I	70.00	0.23	70.74	0.20	90.91
					FLAME					
0.1 0.5	100.0	96.17	100.0 99.98	96.51	100.0	96.47 98.77	0.42	96.90	0.65 0.20	96.37
1.0	100.0 99.99	98.46 98.70	99.98	98.46 98.85	0.48 0.23	98.77 98.84	0.19 0.19	98.45 98.87	0.20	98.60 98.82
1.0	FLDetector								70.02	
0.1	100.00	97.82	99.54	97.19	98.92	98.06	96.30	96.99	89.37	97.65
0.5	100.00	98.95	99.99	98.96	100.0	98.84	99.97	98.97	99.92	98.95
1.0	99.99	99.07	99.98	98.97	99.99	98.93	99.90	99.04	99.89	99.00

filters out several model updates for aggregation.

Besides, more severe heterogeneity negatively affects MA of both Snowball and the baselines, because they are not personalized FL approaches.

D.2 The selection of M

M decides how many model updates are aggregated during one round of federated aggregation. Intuitively, a larger M will bring higher MA (main task accuracy), since the average of more model updates is stable than that of less model updates in terms of convergence. This phenomenon is more pronounced when the data is non-IID, because if too few model updates are aggregated, it will lead to significant differences in the global model's trajectory between two consecutive rounds, jeopardizing the accuracy of global model. However, a larger M may cause infected model updates wrongly selected and aggregated, since as the top-down election gradually selects model updates, the proportion of infected model updates among the remaining candidates in-

creases. Therefore, if M is set too large, there is a higher likelihood that infected models will be mistakenly chosen during the final few steps of top-down election.

Thus, the value of M is a tradeoff between MA and BA. We set M to $\frac{K}{2}$ based on the consideration that the number of malicious clients must not exceed the number of benign clients. Due to the significant consequence that higher BA may cause, we have opted for a relatively conservative strategy of M.

To experimentally demonstrate this, we run Snowball with different M on Fashion MNIST with the other settings the same as in Table 6 of our manuscript.

From the results, we can find that MA increases with the increase of M, but if M is set to 0.8K, the approach fails to defend against backdoor attacks, although there are 0.2K malicious clients. Thus, M should not be set excessively large in pursuit of higher MA.

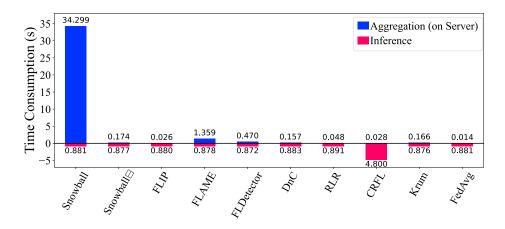


Figure 8: Time consumption of approaches, where the aggregation time is consumed by the server, and the inference time is calculated by conduct model inference on the test set.

Table 6: The performance of Snowball with different of M on MNIST (CBA)

\overline{M}	BA	MA
0.2K	0.17	89.13
0.3K	0.23	89.21
0.4K	0.39	89.23
0.5K	0.39	89.27
0.6K	0.17	89.63
0.7K	0.22	90.07
0.8K	92.02	89.95

D.3 Time Consumption

The main limitation of Snowball lies in its complexity mainly brought by the training, repeated tuning and prediction of the VAE. Fortunately, such overhead is added only at the server, which is usually equipped with powerful computation resources supporting massive parallel computing. Besides, the aggregation in FL is not very frequent. In contrast, the main time consumption of FL usually comes from training and transmitting.

We run these approaches on MNIST at one of the previously mentioned platform, i.e., an Ubuntu 22.04.2 platform with an Intel(R) Core(TM) i9-12900K CPU, an NVIDIA RTX 3090 GPU and 64 GB RAM, to have a numerical result on their time consumption. Figure 8 presents the time consumption of these approaches taken by aggregation of model updates and inference on the test set, respectively. As presented, Snowball takes more time than its peers and Snowball for aggregation. Snowball does not incur significant additional time overhead compared with the peers of Snowball. This indicates that the main consumption of Snowball lies in the training, repeated tuning and prediction of the VAE.

For inference, we can find that Snowball and its peers except CRFL take approximately the same amount of time for inference on the test set, while CRFL takes significantly more time compared to other approaches, because it relies on model ensembles for inference.

As previously mentioned, the additional time overhead brought by Snowball is added only at the server. On one hand, the server usually has powerful computational resources, and by leveraging large-scale parallel computing, significant reductions in aggregation time can be achieved. On the other hand, the aggregation in FL does not typically occur frequently because local training tends to take longer time compared to the aggregation.