Towards Attack-tolerant Federated Learning via Critical Parameter Analysis

Sungwon Han[*1] Sungwon Park[*1] Fangzhao Wu[2] Sundong Kim[3]
Bin Zhu[2] Xing Xie[2] Meeyoung Cha[4,1]
[1] KAIST [2] Microsoft Research Asia [3] GIST [4] Institute for Basic Science

Abstract

Federated learning is used to train a shared model in a decentralized way without clients sharing private data with each other. Federated learning systems are susceptible to poisoning attacks when malicious clients send false updates to the central server. Existing defense strategies are ineffective under non-IID data settings. This paper proposes a new defense strategy, FedCPA (Federated learning with Critical Parameter Analysis)*. Our attack-tolerant aggregation method is based on the observation that benign local models have similar sets of top-k and bottom-k critical parameters, whereas poisoned local models do not. Experiments with different attack scenarios on multiple datasets demonstrate that our model outperforms existing defense strategies in defending against poisoning attacks.*

1. Introduction

The proliferation of computing devices like mobile phones has led to an increase in proprietary user data. The abundance of user data offers the opportunity to create numerous applications but also raises concerns about data privacy. Federated learning (FL) is a cutting-edge collaborative technique that addresses the privacy challenge by enabling machine learning on decentralized devices without exchanging locally stored data [26]. For example, a prominent FL model, FedAvg [17], works as follows: Given a central server and multiple clients, the central server selects a random subset of clients and sends the global model to them. Then, each selected client uses its own data to optimize the local model and sends back the model update to the central server. The central server takes the *average* of these received updates to construct a new global model. This FL framework enables a decentralized system to train a globally shared model via aggregating updates from local models while preserving data privacy.

However, the averaging operation used in the central server leaves room for *poisoning attacks* [2, 16] when malicious clients pose as ordinary clients and submit fraudulent model updates. Attackers can not only impede the convergence of model training and degrade performance [23] (which is called *untargeted attacks*) but they can also manipulate model updates by injecting a backdoor into the resulting global model without substantially degrading its performance [25] (which is called *targeted attacks*).

Several defense strategies have been proposed to eliminate false updates from potentially malicious clients and maintain benign updates on FL systems. For instance, one idea is to use outlier-resistant statistics such as the median or trimmed mean [31, 32] rather than the average in model aggregation. Blanchard et al. [3] proposed Krum, which removes atypical model updates with low local density compared to their k-nearest neighbors. Fung et al. [8] and Fu et al. [7] proposed weighted averaging of local updates in proportion to each update's normality level. Nevertheless, these defense strategies cannot detect adversaries in so-called non-IID (non-independent, identically distributed) situations, where data distributions vary substantially among clients. Existing defense strategies project model updates as individual Euclidean vectors and evaluate their abnormality based on their distances from other model updates. Meanwhile, the non-IID property leads to diverse benign updates, which makes malicious and benign updates indistinguishable in Euclidean space. As a result, existing defense strategies become ineffective [2, 19].

This paper presents FedCPA (Federated learning with Critical Parameter Analysis), an attack-tolerant aggregation method for FL under non-IID data settings. Inspired by a recent observation that not all model parameters contribute equally to optimization [6, 28], we assess the importance of the model parameters in every client's update. Our analysis shows that benign model updates share similar sets of top-k and bottom-k important parameters, even under non-IID data. However, this pattern is not observed for malicious model updates. Based on this observation, we propose a new defense strategy tailored for FL systems to measure model similarity, which extends beyond the extant Euclidean-based similarity and provides an efficient way to discard updates from clients that are likely malicious.

[*]Equal contribution to this work.

FedCPA consists of two steps: (1) computing the *normality* score of each client's model concerning parameter importance and (2) aggregating local updates via a weighted average to remove the effect of likely-malicious updates. In the first step, the importance of each parameter is computed by multiplying its value by its change after local training. The resulting parameters are then ranked in order of importance. Top-k and bottom-k most important parameters are extracted for each client's model and used to compute the similarity among clients' models. Then, we define the normality of the model update to measure its similarity with other model updates. Model updates that differ from other updates are considered malicious. In the second step, outlier local updates are filtered out by adjusting their weights regarding their normality scores.

Our evaluation demonstrates that FedCPA protects against both untargeted and targeted attacks better than existing methods such as Multi-Krum [3], FoolsGold [8], and ResidualBase [7]. We make the following contributions:

- We empirically show that benign local models in federated learning exhibit similar patterns in how parameter importance changes during training. The top and bottom parameters have smaller rank order disruptions than the medium-ranked parameters.
- Based on the data observation that holds over non-IID cases, we present a new metric for measuring model similarity (Eq. 5). With this measure, FedCPA can efficiently assess the normality of each local update, enabling attack-tolerant aggregation.
- Extensive experiments demonstrate the superiority of FedCPA in terms of defense performance. For example, FedCPA reduces the success rate of targeted attacks by a factor of 3 (from 51.4% to 21.9%) on CIFAR-10 and by a factor of 2 (from 74.6% to 43.2%) on TinyImageNet.

The proposed model can be used in various federated learning contexts as a more robust and attack-tolerant decentralized computing framework. Codes are available at https://github.com/Sungwon-Han/FEDCPA.

2. Related Work

2.1. Model Poisoning Attacks

Due to its decentralized nature, federated learning is susceptible to model poisoning attacks and allows malicious clients to send harmful updates to the central server without supervision [27]. As local training data is not shared, malicious participants launch attacks without a full understanding of the entire dataset [5]. Model poisoning attacks can be categorized into untargeted and targeted attacks.

In an *untargeted attack* scenario, attackers aim to indiscriminately degrade the model's overall performance across all classes [22]. Simple and widely used methods of untargeted attack include label-flipping and adding Gaussian noise, which can be executed without prior knowledge of the entire training data distribution [23]. A label-flipping attack involves malicious clients sending false update signals by randomly altering the class label of the training data [29]. On the other hand, Gaussian noise attacks send random noise with the same distribution as the local model prior to the attack in place of the benign client updates [5].

In a *targeted attack*, the objective of a malicious client is to deliberately introduce a backdoor into the global model, which predicts a specific target label for any input overlaid with the backdoor trigger but otherwise behaves like a normal model with a similar overall performance [1, 9, 30]. The backdoor trigger can be a small square to be blended into the original image or a fixed watermark on the image [4, 15].

2.2. Defense Strategies in Federated Learning

Operation based strategy. The main objective of defense strategies is to screen harmful updates from malicious clients. The first representative line of work involves dimension-wise aggregation, which employs outlier-resilient operations instead of a simple average. For example, *Median* aggregates local updates by computing the median value for each dimension of the updates [31]. *Trimmed Mean* is another aggregation method that eliminates a specified percentage of the smallest and largest values, then computes the average of the remaining values [32].

When the training data is of a non-IID distribution, the median aggregation method becomes less effective because it overlooks underrepresented updates. To tackle this limitation, *RFA* suggests using an approximate geometric median operation [20]. *ResidualBase*, on the other hand, introduces residual-based aggregation to determine parameter confidence after calculating the residual of each model parameter via a median estimator [7].

Anomaly detection based strategy. The next line of work involves using anomaly detection to identify and remove malicious updates during aggregation. One representative work is *Krum*, which uses the Euclidean norm space to identify updates far from benign as malicious [3]. In Krum, a local model update that shows the highest similarity to $n - m - 2$ of its neighboring updates is identified as benign, with m denoting the anticipated number of malicious clients. Multi-Krum extends this idea by selecting multiple benign local updates iteratively using Krum. Another approach, *FoolsGold*, identifies the coordinated actions of targeted attacks [8]. Operating under the assumption that malicious clients engaged in a targeted attack exhibit similar update patterns, Foolsgold adjusts the learning rate of

model updates, scaling it in proportion to the diversity of the updates. *Norm Bound* excludes clients whose local updates exceed a certain threshold for the norm, as malicious clients tend to produce updates with larger norms [24].

3. Problem Statement

Federated Learning. Suppose a set of N clients in total in a federated learning system as $\mathcal{C}$ and a set of training sample data in the i-th client as $\mathcal{D}_i$ ($i \in \{1, ..., N\}$). FL aims to train a single global model parameterized as ϕ without directly sharing the local dataset $\mathcal{D}_i$ with others. Given loss objective $\mathcal{L}_i$ in the i-th client and its empirical loss l_i, the main objective for optimizing ϕ can be expressed as

$$\begin{aligned} &\arg\min_{\phi} \mathcal{L}(\phi) = \mathbb{E}_{i\in[1..N]}[\mathcal{L}_i(\phi, \mathcal{D}_i)], \\ &\text{where } \mathcal{L}_i(\phi, \mathcal{D}_i) = \mathbb{E}_{(\mathbf{x},y)\in\mathcal{D}_i}[l_i(\mathbf{x}, y; \phi)]. \end{aligned} \tag{1}$$

Following the literature [8], we choose FedAvg [17] as the default setting to optimize Eq. 1 in the following way. FedAvg divides each training iteration into multiple steps. At the beginning of the t-th iteration ($t \geq 0$), the central server randomly selects a subset of clients and distributes its global model ϕ^t. Then, selected clients update their local model weights θ_i^t with their dataset $\mathcal{D}_i$ and send these updates as $\Delta_i^t = \theta_i^t - \phi^t$ to the central server. Then, the central server aggregates receive local model updates and modifies the global model weight ϕ^{t+1} as follows (hereafter called the *central aggregation process*):

$$\phi^{t+1} = \phi^t + \frac{\sum_{i\in[1..N]} |\mathcal{D}_i| \cdot \Delta_i^t}{\sum_{i\in[1..N]} |\mathcal{D}_i|}. \tag{2}$$

This process repeats until the global model converges.

Threat model. Consider a scenario where M malicious clients ($M < N$) infiltrate the FL system to disturb or manipulate the central aggregation process by transmitting false local updates. Because FL systems are decentralized, attackers cannot access updates from benign users and hence have a limited view of the entire data distribution. We consider two different types of poisoning attacks. One is untargeted attacks, in which attackers may send Gaussian noise to the central server or train the local model with randomly swapped labels. Such tampering can harm the global model's performance. The other type is targeted attacks, in which attackers send model updates containing a backdoor with a carefully designed backdoor trigger. This will cause the global model to incorrectly classify test samples under a specific target label.

Attack-tolerant central aggregation. Most FL systems assume that all participants are benign and that their local updates are reliable. This assumption leaves the system vulnerable to attacks that try to alter or manipulate updates for malicious purposes. Attack-tolerant central aggregation methods have been proposed to mitigate the impact of malicious updates [7, 31, 32].

Let $\mathcal{C}_m$ denote a set of malicious clients and $\mathcal{C}_b$ a set of benign clients, $\mathcal{C} = \mathcal{C}_m \cup \mathcal{C}_b$. Then, the objective of attack-tolerant central aggregation is to design the aggregation function $g^*(\cdot)$, which can be defined as follows,

$$\begin{aligned} \phi^{t+1} &= \phi^t + \frac{\sum_{i\in[1..N]} \mathbf{1}(i \in \mathcal{C}_b) \cdot \Delta_i^t}{N - M} \\ &= \phi^t + \sum_{i\in[1..N]} g^*(i) \cdot \Delta_i^t, \end{aligned} \tag{3}$$

where $\mathbf{1}(i \in \mathcal{C}_b)$ is an indicator function that becomes one if client $i \in \mathcal{C}_b$ and zero if client $i \notin \mathcal{C}_b$. The term $|\mathcal{D}_i|$ in Eq. 2 is omitted here to prevent magnifying the effect of false updates by attackers with increased sizes of their datasets.

4. Critical Parameter Analysis

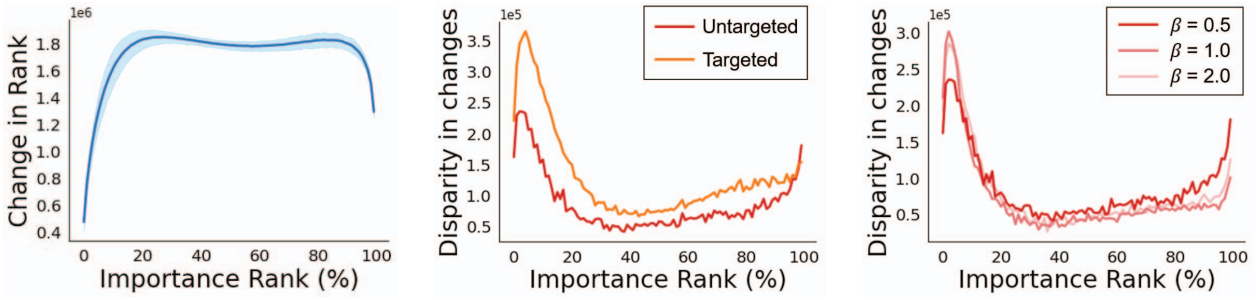
Given the problem statement, our goal, as formulated in Eq. 3, is to determine which updates are malicious and neutralize their impact during the central aggregation process. Prior studies used L_2 distance-based similarity, assuming that false updates are positioned far from benign updates in the Euclidean space [3, 7]. Such an approach performs poorly in the non-IID setting [2], where benign updates become diverse enough to be separated from malicious updates. Motivated by a recent study that demonstrated parameters play diverse roles in model training [6, 13, 28], we adopt an alternative approach to examine parameter importance and identify common patterns among benign updates distinguishable from malicious updates. Our new defense strategy is robust under non-IID data distributions.

Let θ_i^t denote the model parameters of client i at communication round t. After the local training, the model update is defined as $\Delta_i^t = \theta_i^t - \phi^t$. As originally used in [28], we evaluate the importance p_i of the model parameters of client i with the following equation:

$$p_i[n] = |\Delta_i[n] \cdot \theta_i[n]|, \tag{4}$$

where the notation $[n]$ represents the n-th component value of a given vector.

The role of Eq. 4 is two-fold. First, the magnitude of the update provides information about the intensity of the learning signal imposed on each parameter for optimization [13]. Second, the magnitude of the weight represents how much the parameter contributes to the model's prediction [6]. By considering both the update and the weight, we can comprehensively assess the importance of each model parameter. Specifically, when the value of $p_i[n]$ is large, the parameter is considered critical and can significantly impact



(a) Changes of importance in benign clients (b) Disparity in changes by attacks (c) Disparity in changes with different β

Figure 1: Analysis of importance-rank changes of parameters in federated learning: (a) Averaged change in importance-ranks of parameters in benign local models after one training round with the standard deviation area shaded. (b) Comparison of change patterns under two different poisoning attack scenarios, where the disparity is measured by the difference in importance-rank changes between benign and poisoned models after one training round. (c) The disparity in change patterns of the untargeted attack under varying data heterogeneity determined by β.

the optimization process. If $p_i[n]$ is small, the parameter is considered non-critical and is rarely used for training.

Given a federated learning system with multiple clients and parameter importance information of each local model, we conduct an analysis to answer the questions below.

- **Q1.** *Do benign local models exhibit similar patterns of changing parameter importance during training?*
- **Q2.** *Are there any differences in the change of parameter importance between the training of normal and malicious objectives?*
- **Q3.** *If any patterns are discovered, are they persistent across different non-IID settings and datasets?*

The first question asks about the common change pattern in the importance-ordering of local model parameters among benign clients. To answer this question, we conducted multiple rounds of communication in the FL system using the CIFAR-10 dataset. For each round $t > 1$, the central server shares its global model, ϕ^t, with clients. Clients then record the parameter importance of the shared global model ϕ^t via $p^t_{global}[n] = |\Delta^t[n] \cdot \phi^t[n]|$, where $\Delta^t[n] = \phi^t[n] - \phi^{t-1}[n]$ is the change of the global model made from the previous $t-1$ round. Note that p^t_{global} is identical for all clients since they receive the same model. The parameters were then ordered according to the global model's parameter importance p^t_{global} (x-axis in Figure 1). After the local training, each client i computes the model's importance again, expressed as p^t_i. We analyzed the changes in orderings between p^t_{global} and p^t_i for each round $t > 1$, and the averaged results are displayed in Fig. 1[1].

Figure 1a shows experimental results of importance-rank changes in benign clients. We can see that most rank changes concentrate on parameters of medium importance, whereas the top-importance parameters tend to remain stable and the bottom-importance parameters tend to change less in importance ranks. This finding suggests different roles for parameters in the model; The top-importance parameters may be less susceptible to changes due to their significant role in shaping the model's predictions. On the other hand, the bottom-importance parameters have only a small effect on the prediction, and hence they may be neglected, resulting in fewer importance-rank changes during the optimization process. A similar observation is also made in [28] on the role of model parameters.

The experiment was then repeated in the presence of a poisoning attack. We prepared two models derived from the same global model: one was trained with a normal objective and the other with a malicious objective. The disparity in importance-rank changes between the two models was then computed for both targeted and untargeted attack scenarios. The results are shown in Figure 1b. We can see that the poisoned models for both attack scenarios cause greater perturbations in the top- and bottom-importance parameters. This phenomenon may be explained by the fact that a poisoning attack seeks to alter the most critical parameters for disrupting model optimization and injecting malicious information by awakening the unused parameters (i.e., less important parameters) to cause overfitting to noise.

Finally, we examined if this phenomenon holds for various non-IID data settings and datasets. The level of non-IIDness was adjusted by the beta hyper-parameter (β) in the Dirichlet distribution of clients. The experimental results are shown in Figure 1c and Figure 4 in the Appendix. They both confirm that the pattern persists across varying levels of non-IIDness (adjusted by the β value) and multiple datasets. These observations can be summarized to answer the initial questions as follows:

[1]Note that the scale of the y-axis in Fig. 1 lies within [0, 1.1E7], as we used the ResNet18 model with 1.1E7 trainable parameters.

- **A1.** *When it comes to importance-rank changes of parameters, benign local models in FL systems tend to have similar top and bottom parameters in terms of importance ranks.*

- **A2.** *Poisoned local models in FL systems tend to have different sets of parameters with top and bottom importance compared with benign models, which can either degrade optimization or induce overfitting.*

- **A3.** *The above importance-rank change patterns of parameters persist for different levels of data heterogeneity and datasets.*

5. Main Defense Approach: FedCPA

We present an effective defense method against poisoning attacks, called FedCPA. Our key idea is to define a new model similarity metric through critical parameter analysis and measure the normality of each local update based on this similarity. The model then attempts to filter out and reduce the impact of potentially malicious updates using attack-tolerant central aggregation. We describe each procedure in detail.

Defining local model similarity. Given two local models and their parameter importance computed by Eq. 4, we measure the local model similarity with two criteria: top/bottom-k critical sets similarity and importance rank correlation. First, we extract the indices of the top-k and bottom-k important parameters from each client (i.e., Θ_i^{top}, Θ_i^{bottom} in client i) and compare them by calculating the Jaccard similarity between each pair of parameter sets. Second, to further assess the similarity of the parameter importance pattern, we compute the Spearman correlation of the importance values between two models for both top-k and bottom-k parameter sets. The correlation is calculated over the parameter sets that are common in the two models (i.e., $\Theta_{i\cap j}^{\text{top}} = \Theta_i^{\text{top}} \cap \Theta_j^{\text{top}}$, $\Theta_{i\cap j}^{\text{bottom}} = \Theta_i^{\text{bottom}} \cap \Theta_j^{\text{bottom}}$). These criteria are derived from our observations that the benign local model tends to have similar sets of parameters with top and bottom importance, while poisoned models do not. The similarity measure between local models θ_i and θ_j is defined as the following equation:

$$\begin{aligned}\text{sim}(\theta_i, \theta_j) =& J(\Theta_i^{\text{top}}, \Theta_j^{\text{top}}) + J(\Theta_i^{\text{bottom}}, \Theta_j^{\text{bottom}}) \\ &+ \rho(r_i(\Theta_{i\cap j}^{\text{top}}), r_j(\Theta_{i\cap j}^{\text{top}})) \\ &+ \rho(r_i(\Theta_{i\cap j}^{\text{bottom}}), r_j(\Theta_{i\cap j}^{\text{bottom}})),\end{aligned} \tag{5}$$

where $J(\cdot,\cdot)$ denotes the Jaccard similarity and $\rho(\cdot,\cdot)$ denotes the Spearman correlation between two inputs, which is normalized to [0, 1] to align the scale. Here, r_i and r_j represent the functions that map indices to their ranks in terms of parameter importance for clients i and j, respectively.

Algorithm 1 Central aggregation process with FedCPA

Input: Global model weight ϕ^t, global model weight from previous round ϕ^{t-1}, a set of local clients $\mathcal{C}^t$ with their models θ_i^t, and updates Δ_i^t, given index i.

```
// Computing parameter importance
```
for *each client* $i \in \mathcal{C}^t$ **do**
  $p_i^t[n] = |\Delta_i^t[n] \cdot \theta_i^t[n]|$ in Eq. 4
  $\Theta_i^{\text{bottom}}, \Theta_i^{\text{top}} = \arg\text{sort}(p_i^t)[:k],\ \arg\text{sort}(p_i^t)[-k:]$
end
```
// Measuring normality score
```
for *each client* $i \in \mathcal{C}^t$ **do**
  for *each client* $j \in \mathcal{C}^t, j \neq i$ **do**
    $\Theta_{i\cap j}^{\text{top}} = \Theta_i^{\text{top}} \cap \Theta_j^{\text{top}},\ \Theta_{i\cap j}^{\text{bottom}} = \Theta_i^{\text{bottom}} \cap \Theta_j^{\text{bottom}}$
    $s_{i,j} = \text{sim}(\theta_i^t, \theta_j^t)$ in Eq. 5
  end
  $\mathcal{N}(\theta_i^t) = \text{sim}(\theta_i^t, \phi^{t-1}) + \frac{1}{|\mathcal{C}^t|}\sum_{j\in\mathcal{C}^t} s_{i,j}$ in Eq. 7
  $\tilde{\mathcal{N}}(\theta_i^t) = \text{Scale}(\mathcal{N}(\theta_i^t))$
  $\lambda_i^t = \text{Clip}_{0\sim 1}(\ln \frac{\tilde{\mathcal{N}}(\theta_i^t)}{1-\tilde{\mathcal{N}}(\theta_i^t)} + 0.5)$ in Eq. 8
end
```
// Attack-tolerant update
```
$\phi^{t+1} \leftarrow \phi^t + \frac{1}{\sum_{i\in\mathcal{C}^t} \mathbb{1}(\lambda_i^t > 0)} \sum_{i\in\mathcal{C}^t} \lambda_i^t \cdot \Delta_i^t$ in Eq. 9

Normality score for local model. Assuming that adversarial models would have dissimilar patterns of parameter importance from other benign models, we regard a model with low similarity to others as likely malicious. Given the set of clients $\mathcal{C}^t$ participating in communication round t, normality score $\mathcal{N}(\theta_i^t)$ of the local model θ_i^t can be defined as follows:

$$\mathcal{N}(\theta_i^t) = \frac{1}{|\mathcal{C}^t|} \sum_{j\in\mathcal{C}^t} \text{sim}(\theta_i^t, \theta_j^t). \tag{6}$$

However, relying solely on similarities among local models is susceptible to a Sybil attack, where most clients selected at the beginning of the round are malicious by chance [8]. In this scenario, the normality score for adversarial models can be overestimated, as their updates tend to be similar. To enhance the stability of the defense, we also compare the local model with the global model ϕ^{t-1} from the previous $t-1$ round, resulting in the following normality score,

$$\mathcal{N}(\theta_i^t) = \text{sim}(\theta_i^t, \phi^{t-1}) + \frac{1}{|\mathcal{C}^t|} \sum_{j\in\mathcal{C}^t} \text{sim}(\theta_i^t, \theta_j^t). \tag{7}$$

Attack-tolerant central aggregation. We aggregate local updates through a weighted average, with the weight λ_i^t determined by the normality score $\mathcal{N}(\theta_i^t)$. This allows us to filter out the effect of likely malicious updates, while preserving the knowledge gained from likely benign clients' updates. To convert normality scores into weights, we scale

Method ($\gamma_p = 0.5$)	CIFAR-10 ACC(↑)	CIFAR-10 ASR(↓)	SVHN ACC	SVHN ASR	TinyImageNet ACC	TinyImageNet ASR
No Defense	72.1	71.0	93.0	22.2	39.5	96.6
Median	65.6	77.8	90.7	23.0	32.5	96.1
Trimmed Mean	70.1	51.4	92.2	20.9	39.3	97.2
Multi Krum	69.9	63.8	92.1	21.4	37.1	74.6
FoolsGold	45.5	54.3	79.6	23.5	24.3	92.4
Norm Bound	68.2	61.2	93.1	20.8	36.6	96.7
RFA	**72.8**	56.4	92.3	20.8	37.1	93.9
ResidualBase	70.6	59.9	93.1	21.1	**39.6**	96.9
FedCPA	68.8	**21.9**	**93.3**	**20.6**	30.1	**43.2**

Method ($\gamma_p = 0.8$)	CIFAR-10 ACC(↑)	CIFAR-10 ASR(↓)	SVHN ACC	SVHN ASR	TinyImageNet ACC	TinyImageNet ASR
No Defense	69.3	50.9	92.5	22.0	**38.8**	96.1
Median	62.4	70.6	90.0	23.6	31.5	96.2
Trimmed Mean	71.4	19.0	91.7	21.4	37.9	97.0
Multi Krum	69.0	40.4	90.7	23.4	36.3	19.0
FoolsGold	49.1	46.8	69.8	32.3	28.5	69.1
Norm Bound	64.9	53.1	92.7	20.9	35.7	97.1
RFA	70.1	44.8	91.8	22.1	36.3	11.4
ResidualBase	69.9	54.0	92.5	21.9	38.6	96.2
FedCPA	**72.3**	**12.5**	**93.1**	**20.8**	38.7	**4.8**

Table 1: Comparison of defense performance over three datasets under targeted attack scenarios with different levels of pollution ratio $\gamma_p = 0.5, 0.8$. ACC and ASR refer to the final accuracy and the attack success rate, respectively. The symbol (↑) indicates that a higher value is preferable, while (↓) represents the opposite. The best results are marked bold.

Method ($\gamma_p = 0.8$)	CIFAR-10	SVHN	TinyImageNet
No Defense	69.8	90.6	33.0
Median	59.8	89.9	28.7
Trimmed Mean	72.9	91.0	34.1
Multi Krum	72.7	92.6	35.9
FoolsGold	18.6	47.6	4.6
Norm Bound	64.9	90.8	29.3
RFA	72.6	92.7	36.5
ResidualBase	73.6	92.1	36.0
FedCPA	**74.9**	**93.2**	**36.8**

Method ($\gamma_p = 1.0$)	CIFAR-10	SVHN	TinyImageNet
No Defense	63.8	86.1	24.4
Median	56.8	89.6	21.2
Trimmed Mean	66.2	87.9	27.2
Multi Krum	73.0	92.6	35.9
FoolsGold	24.9	41.9	1.3
Norm Bound	63.5	86.6	24.1
RFA	71.5	92.4	**36.3**
ResidualBase	70.3	91.8	30.5
FedCPA	**74.4**	**93.2**	34.9

Table 2: Comparison of defense performance over three datasets under label flipping attack scenarios with different levels of pollution ratio $\gamma_p = 0.8, 1.0$. The best results are marked bold.

each score to the range from 0 to 1 with Min-Max normalization, i.e., $\tilde{\mathcal{N}}(\theta_i^t) \leftarrow \text{Scale}(\mathcal{N}(\theta_i^t))$. Following the literature [8], we apply the inverse sigmoid function to a normalized score to enhance the differentiation of weight values and avoid over-penalization of low, non-zero similarity values on benign clients, resulting in the following weight,

$$\lambda_i^t = \text{Clip}_{0\sim 1}(\ln \frac{\tilde{\mathcal{N}}(\theta_i^t)}{1-\tilde{\mathcal{N}}(\theta_i^t)} + 0.5). \tag{8}$$

where $\text{Clip}_{0\sim 1}(\cdot)$ denotes a function that rounds and clips any values exceeding the 0-1 range. Given the local update from client i as Δ_i, the global model at communication round t is updated as follows:

$$\phi^{t+1} \leftarrow \phi^t + \frac{1}{\sum_{i\in\mathcal{C}^t} \mathbf{1}(\lambda_i^t > 0)} \sum_{i\in\mathcal{C}^t} \lambda_i^t \cdot \Delta_i^t, \tag{9}$$

where $\mathbf{1}(\lambda_i^t > 0)$ is an indicator function that produces one if λ_i^t is larger than zero and zero otherwise. The overall procedure of FedCPA is described in the Algorithm 1.

6. Experiments

We evaluate the effectiveness of FedCPA in defending against several attack scenarios over multiple datasets. Component analyses are conducted to confirm the contribution of each component to robustness under varying simulation hyper-parameters.

6.1. Defense Performance Evaluation

Data. Three benchmark datasets on image classification tasks are utilized in our experiment: (1) CIFAR-10 [11] includes 60,000 samples of 32x32 pixels with 10 classes; (2) SVHN [18] includes 73,257 training samples and 26,032 test samples of 32x32 sized digits; (3) TinyImageNet [12] contains 100,000 samples from 200 classes.

In our experiments, the non-IID property of federated learning in the three datasets is simulated using the Dirichlet distribution, following previous works [10, 14]. The Dirichlet distribution can be denoted as $Dir(N, \beta)$, where N is the total number of clients and β refers to the parameter that adjusts the level of heterogeneity in the decentralized data distributions. A lower value of β results in greater non-IIDness. We set N and β to 20 and 0.5 as default values, respectively.

Method	CIFAR-10	SVHN	TinyImageNet
No Defense	32.7	47.8	2.1
Median	67.8	91.5	28.8
Trimmed Mean	55.6	72.5	12.1
Multi Krum	52.8	68.4	15.0
FoolsGold	13.9	6.7	0.5
Norm Bound	28.2	42.9	1.2
RFA	72.0	92.2	35.8
ResidualBase	74.6	**93.7**	**37.0**
FedCPA	**74.8**	93.6	36.1

Table 3: Accuracy (%) under the Gaussian noise attack over three datasets. The best results are marked bold.

Implementation details. We set the number of communication rounds to 100, with one epoch of local training per round. Following the literature [10, 14], we use ResNet18 as the default backbone network. The SGD optimizer is employed. The learning rate, momentum, and weight decay parameter for the optimizer are set to 0.01, 0.9, and 1e-5. The batch size is set to 64. The hyper-parameter k for top and bottom-k parameter sets is set to 0.01 (1%). To simulate a more realistic federated setting, half of the clients (i.e., $N/2$) are randomly chosen in each round of training. Data augmentation techniques such as random crop, horizontal flip, and color jitter are applied during the local training. In the case of the targeted attack, we follow the original literature [9] and generate a noise input pattern called backdoor. The size of the backdoor is set to 5×5, and its location is in the bottom-right corner of the images. For the untargeted Gaussian attack, we set the standard deviation of the Gaussian noise to 0.05.

Baselines. A total of eight baselines are compared: (1) No Defense represents the classical FedAvg algorithm without any consideration of attack scenarios; (2) Median and (3) Trimmed Mean [31, 32] utilize the outlier-resistant statistics, mean and trimmed mean of local updates, for aggregation; (4) Multi Krum [3] iteratively selects a likely-benign local update with the lowest Euclidean distance from other updates; (5) FoolsGold [8] identifies grouped actions of attacks by inspecting similarity among local updates; (6) Norm Bound [24] filters out the updates whose norm is above a predefined threshold; (7) RFA [20] applies the geometric median operation for robust aggregation; (8) Residual Base [7] introduces a repeated median estimator to compute the confidence of each update.

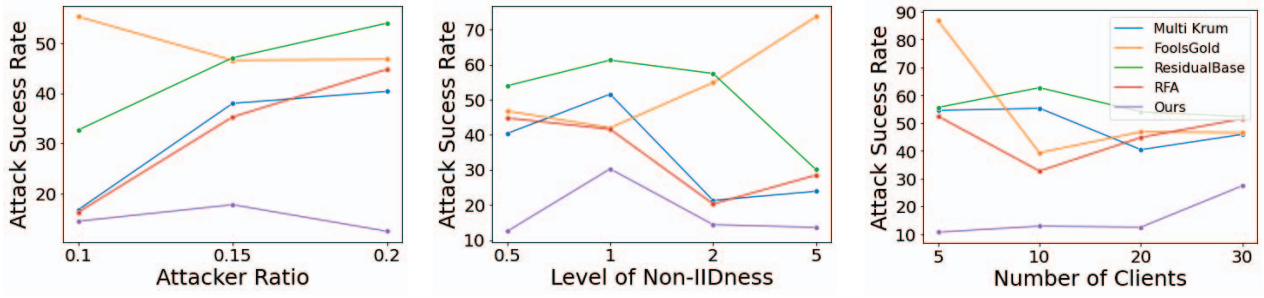
For all baselines, we followed the original implementations and hyper-parameter settings. The confidence interval and clipping threshold in the ResidualBase algorithm are set to 2.0 and 0.05, respectively. In RFA, we set the smoothing parameter to 1e-6 and the maximum number of Weiszfeld iterations to 100.

Setup	Targeted ACC	Targeted ASR	Label flipping ACC	Gaussian ACC	Total
No Defense	2.8	6.2	6.5	7.0	5.6
Median	7.8	7.5	7.5	4.0	6.7
Trimmed Mean	4.2	4.7	4.8	5.3	4.8
Multi Krum	5.7	4.7	2.8	5.7	4.7
FoolsGold	9.0	5.5	9.0	9.0	8.1
Norm Bound	5.2	5.5	6.8	8.0	6.4
RFA	3.8	3.7	2.7	3.0	3.3
ResidualBase	2.7	6.0	3.5	1.3	3.4
FedCPA	3.2	1.0	1.3	1.7	**1.8**

Table 4: Performance comparison summaries among defense strategies. Averaged rank for each evaluation metric under different attack scenarios, including both untargeted and targeted attacks, is reported. Our FedCPA presents superb defense performance.

Evaluation. All methods are assessed under the same experimental settings (e.g., β, the number of clients, communication rounds, and epochs for local training). Given a total of N clients, we set 20% of the clients to play an adversarial role as default. Three attack scenarios are evaluated, one for targeted and two for untargeted attacks. The targeted attack injects a crafted backdoor trigger pattern into some training images and changes their labels to the target class to manipulate the model training. The untargeted attacks consist of the label flipping attack, which randomly alters the labels to generate false update signals [29], and the Gaussian noise attack, which sends Gaussian noise as an update [5]. For both the targeted and label flipping attacks, experiments were conducted with two different levels of pollution ratio (γ_p), representing the fraction of poisoned samples added to the dataset. In the targeted attack experiments, we use a pollution ratio of 0.5 and 0.8, while a pollution ratio of 0.8 and 1.0 is used in the label flipping attack experiments. As an evaluation metric, the final accuracy on the test set is reported for the untargeted attack scenarios, while both the attack success rate and the final accuracy are reported for the targeted attack scenario. All measures are calculated by averaging the last ten rounds of results.

Results. Tables 1-4 present the evaluation results and their summaries for different attack scenarios. FedCPA shows the best or comparable classification accuracy and attack success rate against other defense strategies over all datasets. Our method consistently performs satisfactorily against all types of attacks, whereas some baselines may struggle against specific attacks (e.g., ResidualBase in the targeted attack scenario). Notably, FedCPA reduces the success rate of targeted attacks by a factor of 2 to 4 compared to other baselines on the CIFAR-10 and TinyIm-



(a) Effect of the ratio of attackers (b) Effect of the level of non-IIDness (c) Effect of the number of clients

Figure 2: Robustness test results against targeted attacks on CIFAR-10 with varying experimental settings. The results demonstrate that FedCPA consistently achieves the best defense performance (lowest ASR) compared with the baselines.

Setup	Targeted ACC	Targeted ASR	Untargeted ACC
All components	**72.3**	**12.5**	**74.9**
without topk	70.4	18.9	74.0
without bottomk	60.5	36.8	67.6
without global	68.8	24.1	74.8
without local	65.0	20.2	72.4

Table 5: Ablation study results of FedCPA on CIFAR-10. The best results are marked bold. Our method with full components reports the best defense performance against both targeted and untargeted attacks.

Top/bottom-k ratio	Targeted ACC	Targeted ASR	Untargeted ACC
k = 0.005 (0.5%)	61.0	63.4	71.4
k = 0.01 (1%)	72.3	12.5	74.9
k = 0.02 (2%)	67.7	15.6	74.0
k = 0.05 (5%)	60.2	51.6	74.3

Table 6: Hyper-parameter analysis under both targeted and untargeted attacks on CIFAR-10 with different values of k.

ageNet datasets. These results highlight the effectiveness of our method in providing robustness for FL systems.

6.2. Component Analysis

Ablation study. We conduct an ablation study to evaluate the contribution of each component in our full model. The following variations are compared: (1) without topk only considers and compares parameters of bottom-k importance to compute the normality score of models, while (2) without bottomk is vice-versa; (3) without global omits the similarity term in the normality score between the local model and the global model from the previous round (Eq. 6); (4) without local only utilizes global model similarity for the normality measure (i.e., $\mathcal{N}(\theta_i^t) = \text{sim}(\theta_i^t, \phi^{t-1})$).

Table 5 shows that the full model with all components performs the best against both targeted and untargeted attacks (i.e., label flipping attacks) among all variations, which implies that each component plays an important role in detecting malicious updates. Interestingly, without considering the bottom-k important parameters, the ablation study showed the greatest decrease in defense performance among all the ablations. These results support our hypothesis that poisoning attacks cause a local model to overfit maliciousness by utilizing unused parameters. Therefore, simply focusing on the bottom-k important parameters is also effective in detecting adversarial clients.

Robustness test. Next, we conduct experiments in settings with varying key experimental parameters to assess the robustness of our approach. These include (a) the number of malicious clients $|\mathcal{C}_m|$, (b) the total number of participating clients N, and (c) the degree of non-IIDness, controlled by β in the Dirichlet distribution.

The performance comparison between FedCPA and the baselines on the CIFAR-10 dataset is shown in Figure 2. Only the results for the targeted attack scenario are reported due to the space limitation. More results can be found in the appendix. We can see that, under various experimental settings, FedCPA consistently demonstrates superior defense performance.

Hyper-parameter analysis. We investigate the effect of hyper-parameter k on defense performance. Hyper-parameter k determines the proportion of model parameters selected to create the parameter sets Θ_i^{top} and Θ_i^{bottom} (i.e., the top-k and bottom-k most important parameters) for each client i. The smaller k, the fewer parameters are compared to compute the normality score of the model.

The results for various values of k are presented in Table 6. Our method demonstrates satisfactory results for most measures under both targeted and untargeted attack

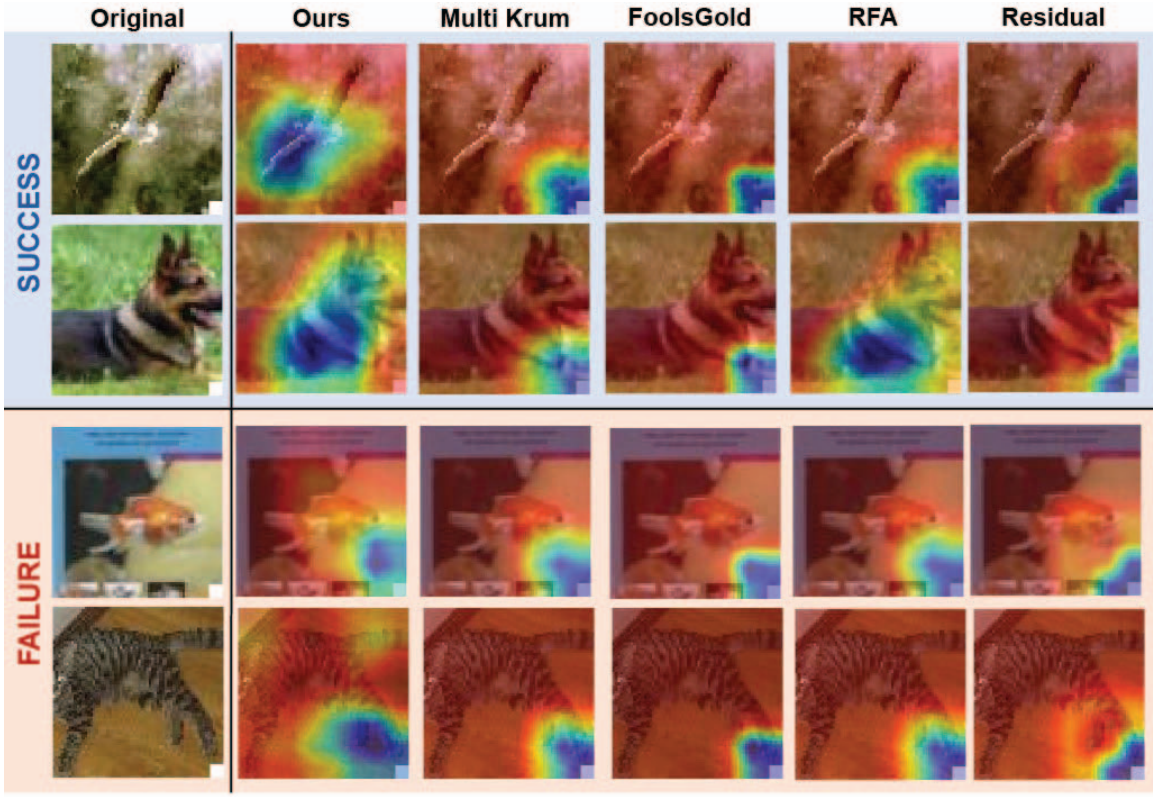


Figure 3: Qualitative analysis under a targeted attack scenario over TinyImagenet, where the highlighted part visualizes how the model recognizes class characteristics based on the Grad-CAM algorithm.

(i.e., label flipping attack) scenarios when k is within a reasonable range of 1-2%. However, setting k to a value too small or too large significantly decreases the performance. This is because the normality measure with a small k may not have enough evidence to distinguish malicious updates, while the measure with a large k can be disturbed by the importance changes caused by data heterogeneity.

Qualitative Analysis We also perform a qualitative analysis to assess how effectively FedCPA can filter out malicious knowledge during training under targeted attack scenarios. Figure 3 compares the performance of different defense strategies in interpreting class characteristics after training. To evaluate each model's interpretation, we corrupted test set images with a small patch of noise used by attackers and used the Grad-CAM algorithm [21] to visualize the model's attention for each input. Blue-framed images represent success cases randomly sampled from the dataset, while red-framed images represent failure cases. Our method tends to extract key features from the image compared to other cases where the model is contaminated by malicious knowledge and only focuses on the injected noise patch. Even in failure cases, our approach gives attention to other visual traits along with the noise, demonstrating its robustness against attacks.

7. Conclusion

We presented FedCPA, a defense strategy against poisoning attacks in federated learning systems. Our method is based on the observation that benign local models tend to have similar sets of important parameters, while adversarial models do not. To distinguish malicious updates, we propose a new normality measure that considers the pattern of important parameters in local models. Then, we aggregate local updates via a weighted average, where the weight of a local update is determined by its normality score. Extensive experiments with both targeted and untargeted attack scenarios on multiple datasets demonstrate the effectiveness of FedCPA in defending against poisoning attacks. Our work contributes to the ongoing efforts on attack-tolerant federated learning and provides new insights for future research.

Acknowledgements. This research was supported by the Institute for Basic Science (IBS-R029-C2). Sungwon Han, Sungwon Park, and Meeyoung Cha were supported by the National Research Foundation of Korea (NRF) grant (RS-2022-00165347). Sundong Kim also received the NRF grant funded by the Ministry of Science and ICT (RS-2023-00240062).

References

[1] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *Proceedings of AISTATS*, pages 2938–2948. PMLR, 2020. 2

[2] Gilad Baruch, Moran Baruch, and Yoav Goldberg. A little is enough: Circumventing defenses for distributed learning. In *Advances in NeurIPS*, volume 32, 2019. 1, 3

[3] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Advances in NeurIPS*, volume 30, 2017. 1, 2, 3, 7

[4] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017. 2

[5] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Gong. Local model poisoning attacks to {Byzantine-Robust} federated learning. In *Proceedings of USENIX Security*, pages 1605–1622, 2020. 2, 7

[6] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *Proceedings of ICLR*, 2019. 1, 3

[7] Shuhao Fu, Chulin Xie, Bo Li, and Qifeng Chen. Attack-resistant federated learning with residual-based reweighting. *arXiv preprint arXiv:1912.11464*, 2019. 1, 2, 3, 7

[8] Clement Fung, Chris JM Yoon, and Ivan Beschastnikh. The limitations of federated learning in sybil settings. In *Proceedings of RAID*, 2020. 1, 2, 3, 5, 6, 7

[9] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017. 2, 7, 11

[10] Sungwon Han, Sungwon Park, Fangzhao Wu, Sundong Kim, Chuhan Wu, Xing Xie, and Meeyoung Cha. Fedx: Unsupervised federated learning with cross knowledge distillation. In *Proceedings of ECCV*, pages 691–707, 2022. 6, 7, 11

[11] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 6

[12] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *Stanford CS 231N*, 7(7):3, 2015. 6

[13] Namhoon Lee, Thalaiyasingam Ajanthan, and Philip Torr. Snip: Single-shot network pruning based on connection sensitivity. In *Proceedings of ICLR*, 2019. 3

[14] Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *Proceedings of CVPR*, pages 10713–10722, 2021. 6, 7, 11

[15] Yunfei Liu, Xingjun Ma, James Bailey, and Feng Lu. Reflection backdoor: A natural backdoor attack on deep neural networks. In *Proceedings of ECCV*, pages 182–199. Springer, 2020. 2

[16] Lingjuan Lyu, Han Yu, and Qiang Yang. Threats to federated learning: A survey. *arXiv preprint arXiv:2003.02133*, 2020. 1

[17] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of AISTATS*, pages 1273–1282, 2017. 1, 3

[18] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011. 6

[19] Sungwon Park, Sungwon Han, Fangzhao Wu, Sundong Kim, Bin Zhu, Xing Xie, and Meeyoung Cha. Feddefender: Client-side attack-tolerant federated learning. *arXiv preprint arXiv:2307.09048*, 2023. 1

[20] Krishna Pillutla, Sham M Kakade, and Zaid Harchaoui. Robust aggregation for federated learning. *IEEE Transactions on Signal Processing*, 70:1142–1154, 2022. 2, 7

[21] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of ICCV*, pages 618–626, 2017. 9

[22] Ali Shafahi, W Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. Poison frogs! targeted clean-label poisoning attacks on neural networks. In *Advances in NeurIPS*, volume 31, 2018. 2

[23] Jacob Steinhardt, Pang Wei W Koh, and Percy S Liang. Certified defenses for data poisoning attacks. In *Advances in NeurIPS*, volume 30, 2017. 1, 2

[24] Ziteng Sun, Peter Kairouz, Ananda Theertha Suresh, and H Brendan McMahan. Can you really backdoor federated learning? *arXiv preprint arXiv:1911.07963*, 2019. 3, 7

[25] Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. *Advances in NeurIPS*, 31, 2018. 1

[26] Jianyu Wang, Zachary Charles, Zheng Xu, Gauri Joshi, H Brendan McMahan, Maruan Al-Shedivat, Galen Andrew, Salman Avestimehr, Katharine Daly, Deepesh Data, et al. A field guide to federated optimization. *arXiv preprint arXiv:2107.06917*, 2021. 1

[27] Chuhan Wu, Fangzhao Wu, Tao Qi, Yongfeng Huang, and Xing Xie. Fedattack: Effective and covert poisoning attack on federated recommendation via hard sampling. In *Proceedings of ACM SIGKDD*, 2022. 2

[28] Xiaobo Xia, Tongliang Liu, Bo Han, Chen Gong, Nannan Wang, Zongyuan Ge, and Yi Chang. Robust early-learning: Hindering the memorization of noisy labels. In *Proceedings of ICLR*, 2021. 1, 3, 4

[29] Han Xiao, Huang Xiao, and Claudia Eckert. Adversarial label flips attack on support vector machines. In *Proceedings of ECAI*, pages 870–875. IOS Press, 2012. 2, 7

[30] Chulin Xie, Keli Huang, Pin-Yu Chen, and Bo Li. Dba: Distributed backdoor attacks against federated learning. In *Proceedings of ICLR*, 2020. 2

[31] Cong Xie, Oluwasanmi Koyejo, and Indranil Gupta. Generalized Byzantine-tolerant SGD. *arXiv preprint arXiv:1802.10116*, 2018. 1, 2, 3, 7

[32] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *Proceedings of ICML*, pages 5650–5659, 2018. 1, 2, 3, 7