BARFED: Byzantine Attack-Resistant Federated Averaging Based on Outlier Elimination

Ece Isik-Polat  
Graduate School of Informatics  
Middle East Technical University  
eceisik@metu.edu.tr

Gorkem Polat  
Graduate School of Informatics  
Middle East Technical University  
gorkem.polat@metu.edu.tr

Altan Kocyigit  
Graduate School of Informatics  
Middle East Technical University  
kocyigit@metu.edu.tr

Abstract

In federated learning, each participant trains its local model with its own data and a global model is formed at a trusted server by aggregating model updates coming from these participants. Since the server has no effect and visibility on the training procedure of the participants to ensure privacy, the global model becomes vulnerable to attacks such as data poisoning and model poisoning. Although many defense algorithms have recently been proposed to address these attacks, they often make strong assumptions that do not agree with the nature of federated learning, such as Non-IID datasets. Moreover, they mostly lack comprehensive experimental analyses. In this work, we propose a defense algorithm called BARFED that does not make any assumptions about data distribution, update similarity of participants, or the ratio of the malicious participants. BARFED mainly considers the outlier status of participant updates for each layer of the model architecture based on the distance to the global model. Hence, the participants that do not have any outlier layer are involved in model aggregation. We perform extensive experiments on many grounds and show that the proposed approach provides a robust defense against different attacks.

1 Introduction

In federated learning (FL), each participant trains its local model independently without sharing its own data and only learned parameters of these local models are sent to a trusted server to form the aggregated global model. Then, the trusted server sends the parameters of the aggregated global model back to these participants and they continue local training with these new parameters. This process is iterated until a stopping criterion, which is generally convergence of the global model, is met [15]. Several aggregation approaches and optimization algorithms that are mainly variations of the Stochastic Gradient Descent (SGD) have been proposed [6, 11, 15, 16, 26]. Although these methods are based on the assumption that all participants are trusted, recent studies show that the learning performance can be greatly reduced in the presence of attacks.

Since privacy concerns are one of the main motivations of the FL, the trusted server that forms the global model has no control over the participants’ local data and training process [3]. Thus FL becomes vulnerable to attacks such as data poisoning, model poisoning, and backdoor attacks. In data poisoning attacks, malicious participants manipulate or modify data, e.g., by adding noise to the training data or label flipping [3, 9, 18, 19]. In model poisoning attacks, model updates of the
participant are altered. Byzantine attack in which malicious participants send arbitrary updates is one of the prevalent model poisoning attacks [3, 4, 7, 8, 18]. Backdoor attacks aim to affect the global model adversarially on a particular sub-task, for example, by making the model classify “trucks” as “planes” by adding small visual artifacts to the training set [1, 9, 10, 18, 21].

Although many defense algorithms incorporated into aggregation rules are proposed to handle these attacks and prevent performance degradation [1, 4, 5, 23–25], many studies have shown that they make assumptions that are usually not met in practical FL setting [2, 8, 10, 17]. In particular, Non-IID datasets and organized (coordinated) attacks bring serious issues to the learning problem and invalidate assumptions of previous works. Moreover, defense strategies that require examining the local datasets and utilize partial or full knowledge of the training process (defense against backdoor attacks and approaches using data sanitization) are not appropriate in practical FL settings. Thus, an important area of study is to analyze and develop these approaches in realistic FL environments.

Contributions. In this paper, we propose an attack-resistant federated averaging method called BARFED. BARFED does not make any unrealistic assumptions about data distributions, update similarities of participants or having information about the malicious participant ratios. We conducted comprehensive experiments to evaluate the performance of BARFED on three different datasets under different attack types, organized/independent attacks, and IID/Non-IID data distributions. We also compared the performance of BARFED with recently proposed defense algorithms. We showed that although a recently proposed robust defense algorithm performs well on IID settings, it did little or no defense in Non-IID settings. In contrast, the proposed approach BARFED stabilizes the convergence and protects the global model from various attacks in any setting.

The rest of the paper is organized as follows. The related work in the literature is reviewed in Section 2. In Section 3, the methodology of the proposed approach is introduced. In Section 4, the details of experimental design are presented. The results and effectiveness of the approach are presented in Section 5. The conclusion of this study is given in Section 6. Finally, the broader impact of this study is discussed in Section 7. Complementary experimental details are given in Appendix as supplementary material.

2 Related Work

Attack-robust FL is a heavily studied topic in recent years. [25] introduced two different approaches instead of solely averaging gradients. The first method was the coordinate-wise median and the second was the coordinate-wise trimmed mean that excludes the highest and smallest values with the given percentage. [4] proposed a Byzantine fault-tolerant SGD algorithm called Krum that combines the majority-based and square-distance methods. [7] introduced a method that combines Krum and trimmed mean, called Bulyan. These methods presume IID data and the ratio of malicious participants should be known in each communication round, which usually does not hold in FL settings.

There are also clustering or similarity metrics based methods that work under certain conditions and assumptions. [9] assumes that the trusted participants have a unique distribution and as a result, their gradient updates vary. Since malicious participants have a common goal, their gradient updates tend to be more similar. Based on this assumption, they proposed the FoolsGold algorithm that identifies participants who make similar gradient updates with a method based on cosine similarity and reduces the learning rates of these participants. [18] proposed a method that clusters the participants based on the pairwise cosine dissimilarities between their updates and considers the elements of the largest cluster as benign. [20] presented a method based on identifying malicious participant clusters with a visualization that is obtained by applying Principal Component Analysis to the parameters of the last layer of the participants’ local models. Unlike [9], [18, 20] worked with benign participants that have similar updates on IID data.

The data distribution and update similarities of participants are two essential factors that should be examined in detail. Most of the recent studies proposed methods for IID case ([4, 7, 18, 20, 25]). However, Non-IID distribution of participants’ data is one of the key properties of FL and it was emphasized that existing techniques for Byzantine tolerant distributed learning do not perform well when data of participants are Non-IID [1, 10, 15]. Although the proposed method in [9] addresses Non-IID data distribution, it only covers a very specific case where trusted participants have unique updates and malicious participants have similar updates.
Although there have been notable new studies proposing aggregation methods for distributed learning that ensure the convergence of the global model, they sacrificed classification performance in exchange for convergence, resulting in ineffective strategies that are not useful for FL settings [3, 4, 7, 25].

3 Proposed Method

The proposed model is based on Federated Averaging (FedAvg) [15], which is one of the most widely used aggregation algorithms in FL [14]. In FedAvg, participants receive the latest global model from the server, update the model using their local data and the resulting models are transferred to the server. Then, the server updates the global model by averaging the weights received from the participants, i.e., the server and the participants collectively train a neural network without sharing local datasets.

In a standard FL process, when there is no attack and the global model converges, it is expected that the local models of participants are unlikely to be far from the global model. In contrast, in the presence of malicious participants, their models drift apart from the trusted ones and the global model [3]. The main idea of our method is determining outliers by evaluating layerwise distances to the global model and utilizing the remaining reliable participants. At any round $t$, when the updates $(w_{t+1}^1, w_{t+1}^2, ..., w_{t+1}^n)$ are received at the server, BARFED calculates the distance1 of each participant’s $i^{th}$ layer to the global model’s $i^{th}$ layer, $d^i = \|w_t^i - w_{t+1}^i\|$. Then, for each layer, outliers $d^i$ are determined according to the boxplot outlier elimination technique. Let $OS(d^i) = 1^2$, if $i^{th}$ layer of participant $p$ is not an outlier. So, if $OS(d^1) \land OS(d^2) \land ... \land OS(d^n) = 1$ (or True), we mark that participant as reliable $w_{t+1}^{rel}$. Finally, BARFED determines the average of reliable participants’ weights as the new weight of the global model, $w_{t+1} = \frac{1}{n_{rel}} \sum_{n=1}^{n_{rel}} w_{t+1}^{rel}$. Complete pseudocode is given in Algorithm 1.

Algorithm 1 BARFED. $T$ is the number of rounds, $P$ is the number of participants, $i$ is used to index layers where $I$ is the total number of layers, and $J$ is the number of local epochs. $Q_1^i$, $Q_3^i$, and $IQR^i$ are first quartile, third quartile and interquartile range for the layer $i$, respectively.

1: procedure SERVERUPDATE
2: initialize weights
3: for each round $t = 1, 2, ..., T$ do
4: send $w_t$ (main model weights) to the participants
5: for each participant $p = 1, 2, ..., P$ do in parallel
6: $w_{t+1}^p \leftarrow$ PARTICIPANTUPDATE($p$, $w_t$) ▷ Local Training
7: for each layer of the model $i = 1, 2, ..., I$ do
8: for each participant $p = 1, 2, ..., P$ do
9: $d^i \leftarrow \|w_t^i - w_{t+1}^i\|$ ▷ Calculate the distance of each participant
10: $lower_{thhr}^i \leftarrow Q_1^i - (1.5 \times IQR^i)$ ▷ Calculate the lower bound for each layer
11: $upper_{thhr}^i \leftarrow Q_3^i + (1.5 \times IQR^i)$ ▷ Calculate the upper bound for each layer
12: for each participant $p = 1, 2, ..., P$ do
13: if $lower_{thhr}^i < d^i < upper_{thhr}^i$, $\forall i$ then
14: mark participant $p$ as reliable, $P_{rel}$
15: $w_{t+1} \leftarrow \frac{1}{n_{rel}} \sum_{n=1}^{n_{rel}} w_{t+1}^{rel}$ ▷ FedAvg with reliable participants
16: procedure PARTICIPANTUPDATE($p$, $w$)
17: for each step $j = 1, 2, ..., J$ do
18: $w \leftarrow w - \eta \nabla l(x^j, y^j; w)$
19: return $w$ to the server

1Unless otherwise stated, all distances in this study refer to the Euclidean distance.
2Outlier Status of the distance of $i^{th}$ layer of a participant $p$. 3
Figure 1 shows the layerwise distances of a randomly selected trusted and a malicious participant to the global model for each communication round. The distance of the trusted participant to the global model for each layer remains inside the aggregation interval (highlighted for each layer, \([Q_1 \sim 1.5 \times IQR, Q_3 + 1.5 \times IQR]\)). In contrast, the distances of the malicious participants are in the outlier region. BARFED is mainly based on making this separation utilizing distances.

One of the main differences of the BARFED from previous defense strategies is that a participant is labeled as malicious and not included in the aggregation even if it has a single outlier layer. Most of the defense strategies proposed in the literature rely on including a model’s parameters partially. Since each parameter within the model is evaluated individually, while a group of parameters is discarded in the aggregation rule, a different group of parameters of the same participant can be included. The main motivation of the proposed all-or-nothing approach is that each participant is evaluated in a holistic approach. Parameters of the neural network architectures are highly dependent on each other; therefore, evaluating these parameters independently may lead to inaccurate inferences. If any layer of the model updates is considered to be malicious, it is a sign for a malicious participant; therefore, there is no need to include it (or rest) in the aggregation step. In the proposed approach, for a participant’s model to be included in the calculation of global model aggregation, each layer must fall within the safe range calculated for that layer, i.e., a consensus should be ensured among all layers of the local model. Interestingly, experiments show that ratio of unreliable participants determined in the proposed approach is very similar to the ratio of actual malicious participants (see Figures 11 and 12 in Appendix).

**Lemma 1.** The expected time complexity of the BARFED\((V_1, V_2, \ldots, V_n)\) is \(O(l \cdot n \log n)\) where \(V_1, V_2, \ldots, V_n\) has \(l\) layer and each layer is \(d\)-dimensional vector.

**Proof.** For each layer in \(V_i\), server computes \(n\) distances for \(d\)-dimensional vectors (\(O(nd)\)). Then, server finds the outliers by sorting the distances of each participant, \(O(n \log n)\). Finally, each layer is checked if it is outlier or not (\(O(n)\)). As a result, a time complexity of \(O(l \cdot (nd + n \log n + n))\) is obtained. Since \(O(n)\) is negligible relatively to \(O(n \log n)\), time complexity can be reduced to \(O(l \cdot n \log n)\).

The computational complexity of the BARFED is much more efficient than Krum and its variant Bulyan, which are quadratic (\(O(d \cdot n^2)\)). BARFED is slightly more efficient than the coordinate-wise median and trimmed mean as they require sorting of all individual parameters (BARFED only makes sorting as many as the number of layers).

![Figure 1](image_url)

Figure 1: Distances of randomly selected trusted (top row) and malicious (bottom row) participants to the global model under label flipping attack on MNIST dataset. Circles refer to the distance of the participant to the same layer of the global model for that round. Shaded areas show upper and lower bounds to determine outliers calculated among all participants for that communication round.
4 Experimental Design

We conducted experiments on MNIST [13], CIFAR10 [12], and Fashion-MNIST [22] datasets. MNIST dataset contains 28×28 grayscale images with 50,000 training images and 10,000 testing images. CIFAR10 dataset contains 32×32 color images with 50,000 training images and 10,000 testing images, and Fashion-MNIST contains 28×28 grayscale images with 60,000 training images and 10,000 testing images.

One of the dimensions of our experiments is the data distribution of the participants that can be either IID or Non-IID. For IID cases, training datasets are distributed to the participants randomly and uniformly, i.e., each participant has each class equally. On the other hand, for Non-IID cases, two different labels for MNIST and Fashion-MNIST and five different labels for CIFAR10 are distributed to the participants.

Another dimension is attack types. There are two attack types examined, namely, label flipping attacks and Byzantine attacks. In label flipping attacks, malicious nodes flip their ground truth labels with a target class label. In Byzantine attacks, malicious participants send random weight updates from a standard normal distribution with zero mean and unit standard deviation.

Lastly, attacker types are investigated. Independent attackers are the malicious participants incapable of coordinating with each other, acting individually, and sending random updates to the server. Organized or coordinated attackers are the malicious participants that carry out the attack in an organized or coordinated manner and send similar updates to the server. For example, in independent label flipping attacks, malicious participants flip their ground truth labels with an arbitrary target label, e.g., if there are two malicious participants with label 7 in their data sets, one flips 7 to 1, while the other flips to 4. On the other hand, in organized label flipping attacks, the malicious participants flip ground truth labels with consistent target labels, e.g., all malicious participants that have 7 in their datasets flip the label as 1. Similarly, for independent Byzantine attacks, malicious participants send different random weights while they send the same random weights in the organized setting.

In order to increase the success of the attacks and reduce the likelihood of malicious participants being caught, the replaced classes were chosen as semantically similar as possible. The replaced classes in the organized setting for each data set are presented in Table 1.

| Table 1: Replaced classes for organized label flipping attack. |
|-----------------|-----------------|-----------------|
| **MNIST**       | **Fashion-MNIST** | **CIFAR10**     |
| Original | Replaced | Original | Replaced | Original | Replaced |
| 0     | 9 | T-shirt/Top | Shirt | Plane | Bird |
| 1     | 7 | Trouser | Dress | Car | Truck |
| 2     | 5 | Pullover | Coat | Bird | Plane |
| 3     | 8 | Dress | Trousers | Cat | Dog |
| 4     | 6 | Coat | Pullover | Deer | Horse |
| 5     | 2 | Sandal | Sneaker | Dog | Cat |
| 6     | 4 | Shirt | T-shirt/Top | Frog | Ship |
| 7     | 1 | Sneaker | Ankle Boot | Horse | Deer |
| 8     | 3 | Bag | Sandal | Ship | Frog |
| 9     | 0 | Ankle Boot | Sneaker | Truck | Car |

There are 100 participants in all experiments. There are two model architectures for MNIST, and only one architecture for CIFAR10 and Fashion-MNIST. The experiments were run on NVIDIA Tesla P100 16GB for 436.7 hours and on NVIDIA Tesla V100 16GB for 525.5 hours. The details of run times for trainings, the model architectures, and the hyperparameters are given in Appendix. For CIFAR10 experiments, data augmentation techniques such as horizontal random flipping and random cropping, and training strategies like learning rate scheduling and gradient clipping were applied to enhance the model performance.

5 Experimental Results & Discussion

Fang et al. [8] has shown that trimmed mean and coordinate-wise median are more robust to attacks than Krum and its variant Bulyan. Unlike coordinate-wise median, trimmed mean requires a malicious
A robust defense strategy should not cause any noticeable performance loss when there is no attack in the FL system. Table 2 and Figure 2 show the results of experiments when all participants are trusted (when there is no attack on any participant). Incorporating both strategies into FedAvg does not cause any performance degradation in the IID setting. Although the performance of the median strategy is slightly worse in Fashion-MNIST and CIFAR10, it can be tolerable in an FL setting. On the other hand, when local datasets are Non-IID, the median strategy causes significant performance degradation, which points to the questionability of the method.

Table 2: Accuracies on test sets when all participants are trusted. The worst results are bold.

|          | IID         | Non-IID     |
|----------|-------------|-------------|
|          | No Defense  | BARFED      | Median | No Defense  | BARFED      | Median |
| MNIST-2NN| 97.7 97.7   | 97.6 97.6   | 97.6   | 96.4 96.6   | 96.2 96.4   | 80.7   |
| MNIST-CNN| 98.9 98.9   | 98.9 98.9   | 98.9   | 98.8 98.8   | 98.8 98.8   | 96.9   |
| Fashion MNIST | 90.4 90.5 | 90.4 90.5 | 90.1   | 86.8 87.4   | 82.4 84.3   | 79.1   |
| CIFAR10  | 78.9 79.0   | 78.4 78.4   | 76.7   | 77.5 77.7   | 77.8 77.9   | 64.3   |

Figure 2: Accuracy curves of different strategies when all participants are trusted.

5.2 Label Flipping Attacks

Table 3 and Table 4 show that as the ratio of malicious participants increases, vanilla FedAvg cannot avoid performance degradation, which requires that a defense mechanism should be incorporated.
When malicious participants are organized, degradation becomes more severe and oscillation increases (see Figure 3). The most degradation occurs when attacks are organized in Non-IID setting.

For IID cases of MNIST-2NN, MNIST-CNN and Fashion-MNIST, BARFED achieves a slightly higher accuracy score most of the time, but differences between BARFED and median are negligible. When comparing with all-trusted performance, both strategies can recover the negative effects of the label flipping attack. For IID cases of CIFAR10 experiments, BARFED achieves noticeably better performance than the median.

When the data of participants is Non-IID, median strategy performed worse than even the vanilla FedAvg. On the other hand, BARFED successfully defends against malicious participants and gets an accuracy score very close to when all participants are trusted. Performance of accuracy curves for MNIST-2NN can be examined in Figure 3.

| Organized | Independent |
|-----------|-------------|
| m%        | No Defense  | BARFED | Median | No Defense  | BARFED | Median |
| 10        | 92.9        | 97.6   | 97.7   | 97.4        | 97.4   | 97.2   | 97.6   | 97.6   | 97.4   | 97.5   |
| 2NN       | 72.4        | 97.3   | 97.4   | 97.5   | 97.2   | 97.3   | 97.1   | 97.4   | 97.5   | 97.0   | 97.1   |

| 10        | 94.2        | 99.0   | 98.9   | 98.9   | 98.9   | 98.9   | 98.9   | 98.9   | 98.9   | 98.9   |
| 20        | 75.4        | 99.0   | 99.9   | 98.8   | 98.8   | 96.2   | 98.9   | 98.9   | 98.9   | 98.9   |

Table 3: Accuracies under label flipping attacks at different attacker ratios in IID settings. The best results are bold.

| Organized | Independent |
|-----------|-------------|
| m%        | No Defense  | BARFED | Median | No Defense  | BARFED | Median |
| 10        | 92.4        | 95.8   | 96.1   | 96.3   | 75.1   | 83.8   | 93.7   | 95.3   | 96.0   | 96.3   | 80.8   | 83.3   |
| 2NN       | 83.8        | 88.4   | 95.6   | 96.1   | 67.7   | 75.0   | 89.3   | 94.0   | 95.5   | 96.1   | 67.8   | 75.9   |

| 10        | 95.5        | 98.0   | 98.6   | 98.7   | 96.4   | 96.7   | 97.0   | 98.3   | 98.7   | 98.8   | 95.4   | 95.9   |
| 20        | 83.4        | 91.0   | 95.6   | 96.6   | 91.4   | 93.1   | 95.3   | 96.6   | 98.6   | 98.7   | 93.5   | 94.2   |

| 10        | 83.0        | 85.7   | 87.7   | 88.6   | 79.9   | 80.6   | 84.3   | 86.7   | 87.5   | 88.5   | 78.7   | 79.6   |
| 20        | 73.7        | 79.2   | 84.2   | 87.6   | 76.0   | 76.7   | 81.5   | 84.6   | 85.1   | 87.6   | 77.4   | 78.5   |

| 10        | 75.2        | 75.3   | 78.0   | 78.1   | 55.0   | 55.9   | 74.4   | 74.4   | 77.3   | 77.4   | 56.7   | 57.2   |
| 20        | 70.8        | 70.9   | 76.8   | 76.9   | 54.5   | 54.9   | 70.2   | 70.3   | 76.2   | 76.3   | 52.0   | 52.7   |

Table 4: Accuracies under label flipping attacks at different attacker ratios in Non-IID settings. The best results are bold.

5.3 Byzantine Attacks

[1, 8, 18] show that Byzantine attacks are more effective than data poisoning attacks. Our experiments, resulting in dramatic performance degradation under Byzantine attacks, are in line with the previous studies. The level of performance degradation caused by organized attackers is more than independent attackers.

When the data distributions of the participants are Non-IID, the performance scores get worse compared to IID cases and Non-IID attacks with organized attackers are the most harmful case. Again, as the number of malicious participants increases, the performance degradation increases, too.

For IID cases, median and BARFED are able to recover the negative effects of Byzantine attacks as if there has been no malicious participant (Table 5, Figure 4). For Non-IID cases, the median is able to prevent performance degradation up to a degree; however, it can not eliminate the performance degradation caused by attacks as well as BARFED. BARFED gets better scores than median for all data sets. Only for Non-IID experiments of MNIST-CNN median can catch the BARFED (Table 6, Figure 4).
6 Conclusion

This study proposes BARFED, an assumption-free attack-resistant federated averaging algorithm based on outlier elimination, and conducts comprehensive experiments in various FL settings. These experiments reveal that Byzantine attacks are dramatically more severe than label flipping attacks. Moreover, attacks in the Non-IID cases are more effective than IID cases and organized attackers can severely compromise the performance of the main model more compared to independent attackers. Although both baseline and BARFED recover performance loss in the presence of attacks in IID cases, the baseline method performs poorly in Non-IID cases, it may even perform worse than the vanilla FedAvg. We put forward experimental evidence to show that BARFED recovers performance loss even under organized attacks and in Non-IID cases. There are many attack-robust aggregation methods and mechanisms for FL in the literature, but they mainly focused on ensuring the convergence under some assumptions such as data distribution, knowledge of malicious participant ratio, and update similarity of participants. Our work highlights the shortfall in current theoretically convergence guaranteed methods and presents a broader research goal to create aggregation mechanisms that work in harmony with Non-IID data, which is one of the key properties of FL.

Nevertheless, our method is mainly based on the outlier elimination and it may tolerate up to a certain number of malicious participants in the system. As the ratio of attackers increases in the FL setting, they will have a high impact on the distribution of distances. Distance distributions also depend on some other parameters such as the severity of the Non-IID data and coordination of attackers; therefore, to what extent BARFED can handle malicious participants is beyond the scope of this study and reserved as future work.

Table 5: Accuracies under Byzantine attacks at different attacker ratios in IID settings. The best results are bold.

|        | Organized |           | Independent |           |
|--------|-----------|-----------|-------------|-----------|
|        | m%        | No Defense| BARFED      | Median     | No Defense| BARFED      | Median     |
| MNIST  | 10        | 57.0      | 68.9        | **97.5**   | 97.2      | 97.2        | 86.0       | 90.7      | **97.5**   | 97.2      | 97.3       |
|        | 20        | 31.9      | 43.9        | **97.5**   | 97.6      | 97.3        | 76.2       | 86.1      | **97.5**   | 97.6      | 97.3       |
| 2NN    | 10        | 54.1      | 79.3        | **98.9**   | 98.9      | 98.9        | 91.9       | 95.0      | **98.9**   | 99.0      | 98.9       |
|        | 20        | 10.0      | 17.0        | **98.8**   | 98.8      | **98.9**    | 70.8       | 87.2      | **98.8**   | 98.8      | **98.8**   |
| MNIST  | 10        | 15.3      | 48.2        | **90.2**   | 90.3      | 89.9        | 90.1       | 79.2      | **90.2**   | 90.3      | 90.0       | 90.2       |
|        | 20        | 9.8       | 21.1        | **90.4**   | **90.5**  | 90.1        | 90.2       | 36.8      | **90.3**   | **90.4**  | 90.0       | 90.1       |
| Fashion| 10        | 6.6       | 13.8        | **74.2**   | **78.0**  | 73.5        | 75.9       | 6.1       | **73.3**   | 77.2      | **73.6**   | **77.8**   |
|        | 20        | 7.2       | 13.9        | **74.8**   | **77.6**  | 70.8        | 75.4       | 6.1       | 16.6       | **73.4**  | **77.5**   | 73.0       | 75.3       |

Figure 3: Accuracy curves of different strategies for MNIST-2NN under label fling attacks at different attacker ratios.
Table 6: Accuracies under Byzantine attacks at different attacker ratios in Non-IID settings. The best results are bold.

|                  | Organized | Independent |
|------------------|-----------|-------------|
|                  | m | No Defense | BARFED | Median | No Defense | BARFED | Median |
|                  | % | min | max | min | max | min | max | min | max |
| MNIST            | 10 | 26.1 | 36.0 | 96.1 | 96.2 | 85.6 | 89.5 | 45.6 | 61.6 | 96.1 | 96.2 | 83.4 | 88.8 |
| 2NN              | 20 | 14.9 | 26.6 | 95.9 | 96.1 | 90.3 | 92.7 | 16.5 | 34.1 | 95.9 | 96.1 | 89.3 | 90.9 |
| MNIST            | 10 | 15.6 | 33.5 | 98.7 | 98.8 | 97.4 | 97.5 | 46.1 | 72.4 | 98.7 | 98.8 | 97.0 | 97.2 |
| CNN              | 20 | 9.00 | 15.7 | 98.6 | 98.7 | 97.6 | 97.7 | 17.4 | 40.2 | 98.6 | 98.7 | 96.8 | 97.1 |
| Fashion          | 10 | 5.70 | 23.6 | 82.3 | 85.6 | 79.4 | 80.8 | 19.7 | 39.6 | 80.8 | 83.8 | 79.7 | 80.8 |
| MNIST            | 20 | 8.60 | 18.0 | 85.3 | 86.5 | 78.4 | 79.6 | 10.0 | 22.2 | 84.5 | 86.3 | 77.7 | 78.3 |
| CIFAR            | 10 | 5.4  | 13.6 | 70.7 | 77.6 | 57.6 | 63.0 | 7.3  | 13.9 | 72.9 | 75.3 | 62.0 | 66.2 |
|                 | 20 | 6.7  | 13.4 | 72.5 | 77.6 | 68.7 | 72.1 | 6.1  | 13.4 | 72.3 | 77.4 | 58.0 | 67.2 |

Figure 4: Accuracy curves of different strategies for MNIST-2NN under Byzantine attacks at different attacker ratios.

7 Broader Impact

Outlier elimination-based defense methods, including BARFED, are likely to discard participants who are not necessarily malicious but have diverse model updates than the others. Although it can be considered inadvertent elimination, it may lead to the underrepresentation of minority groups and impairs diversity and fairness. Therefore, we see it as an important challenge and future research topic.
References

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

[2] G. Baruch, M. Baruch, and Y. Goldberg. A little is enough: Circumventing defenses for distributed learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[3] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo. Analyzing federated learning through an adversarial lens. In *International Conference on Machine Learning*, pages 634–643. PMLR, 2019.

[4] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 118–128, 2017.

[5] L. Chen, H. Wang, Z. Charles, and D. Papailiopoulos. Draco: Byzantine-resilient distributed training via redundant gradients. In *International Conference on Machine Learning*, pages 903–912. PMLR, 2018.

[6] M. Duan, D. Liut, X. Chen, Y. Tan, J. Ren, L. Qiao, and L. Liang. Astra: Self-balancing federated learning for improving classification accuracy of mobile deep learning applications. In 2019 *IEEE 37th International Conference on Computer Design (ICCD)*, pages 246–254. IEEE, 2019.

[7] E. M. El Mhamdi, R. Guerraoui, and S. Rouault. The hidden vulnerability of distributed learning in Byzantium. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3521–3530. PMLR, 10–15 Jul 2018.

[8] M. Fang, X. Cao, J. Jia, and N. Gong. Local model poisoning attacks to byzantine-robust federated learning. In 29th *USENIX Security Symposium* (*USENIX* Security 20), pages 1605–1622, 2020.

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

[10] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

[11] J. Konečný, H. B. McMahan, D. Ramage, and P. Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, 2016.

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

[13] Y. LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[14] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang. On the convergence of fedavg on non-iid data. In *International Conference on Learning Representations*, 2020.

[15] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017.

[16] R. Pathak and M. J. Wainwright. Fedsplit: an algorithmic framework for fast federated optimization. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 7057–7066. Curran Associates, Inc., 2020.
[17] S. Rajput, H. Wang, Z. Charles, and D. Papailiopoulos. Detox: A redundancy-based framework for faster and more robust gradient aggregation. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 32. Curran Associates, Inc., 2019.

[18] F. Sattler, K.-R. Müller, T. Wiegand, and W. Samek. On the byzantine robustness of clustered federated learning. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 8861–8865. IEEE, 2020.

[19] S. Shen, S. Tople, and P. Saxena. Auror: Defending against poisoning attacks in collaborative deep learning systems. In Proceedings of the 32nd Annual Conference on Computer Security Applications, pages 508–519, 2016.

[20] V. Tolpegin, S. Truex, M. E. Gursoy, and L. Liu. Data poisoning attacks against federated learning systems. In European Symposium on Research in Computer Security, pages 480–501. Springer, 2020.

[21] H. Wang, K. Sreenivasan, S. Rajput, H. Vishwakarma, S. Agarwal, J.-y. Sohn, K. Lee, and D. Papailiopoulos. Attack of the tails: Yes, you really can backdoor federated learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, Advances in Neural Information Processing Systems, volume 33, pages 16070–16084. Curran Associates, Inc., 2020.

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

[23] C. Xie, S. Koyejo, and I. Gupta. Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance. In International Conference on Machine Learning, pages 6893–6901. PMLR, 2019.

[24] C. Xie, S. Koyejo, and I. Gupta. Zeno++: Robust fully asynchronous sgd. In International Conference on Machine Learning, pages 10495–10503. PMLR, 2020.

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

[26] H. Yuan and T. Ma. Federated accelerated stochastic gradient descent. Advances in Neural Information Processing Systems, 33, 2020.
A Appendix

A.1 Model Architectures

The model architectures used for each datasets are shown in Table 7 (MNIST-2NN [15]), Table 8 (MNIST-CNN [15]), Table 9 (CIFAR10 [20]) and Table 10 (Fashion-MNIST). All activation functions for all models are ReLU.

Table 7: Model architecture of MNIST-2NN.

| Layer          | Size          |
|----------------|---------------|
| Fully Connected| (784, 200)    |
| Fully Connected| (200, 200)    |
| Fully Connected| (200, 10)     |

Table 8: Model architecture of MNIST-CNN.

| Layer             | Size          |
|-------------------|---------------|
| Conv              | 32@5×5        |
| Max Pooling       | 2×2           |
| Conv              | 64@5×5        |
| Max Pooling       | 2×2           |
| Fully Connected   | (1024, 512)   |
| Fully Connected   | (512, 10)     |

Table 9: Model architecture of CIFAR10.

| Layer             | Size          |
|-------------------|---------------|
| Conv              | 32@3×3, pad=1 |
| Conv              | 32@3×3, pad=1 |
| Max Pooling       | 2×2           |
| Conv              | 64@3×3, pad=1 |
| Conv              | 64@3×3, pad=1 |
| Max Pooling       | 2×2           |
| Conv              | 128@3×3, pad=1|
| Conv              | 128@3×3, pad=1|
| Max Pooling       | 2×2           |
| Fully Connected   | (2048, 128)   |
| Fully Connected   | (128, 10)     |

Table 10: Model architecture of Fashion-MNIST.

| Layer             | Size          |
|-------------------|---------------|
| Conv              | 32@5×3, pad=2 |
| Max Pooling       | 2×2           |
| Conv              | 64@5×5, pad=2 |
| Max Pooling       | 2×2           |
| Fully Connected   | (3136, 500)   |
| Fully Connected   | (500, 10)     |

The FL setting parameters used for each dataset are shown in Table 11. Learning rate scheduling and clipping were not applied to MNIST and Fashion-MNIST, therefore related parameters are set as N/A (Not Applicable).

Table 11: FL setting parameters used in experiments.

| Parameters                  | MNIST-2NN | MNIST-CNN | Fashion MNIST | CIFAR10 |
|-----------------------------|-----------|-----------|---------------|---------|
| number of participant (n)   | 100       | 100       | 100           | 100     |
| communication round (t)     | 200       | 200       | 200           | 500     |
| number of label in each     | 2         | 2         | 2             | 5       |
| participant in IID setting  |           |           |               |         |
| number of label in each     |           |           |               |         |
| participant in Non-IID setting |       |           |               |         |
| batch size                  | 32        | 32        | 25            | 100     |
| number of epoch             | 10        | 10        | 10            | 10      |
| momentum                    | 0.9       | 0.9       | 0.9           | 0.9     |
| learning rate               | 0.01      | 0.01      | 0.002         | 0.0015  |
| minimum learning rate (min_lr) | N/A   | N/A       | N/A           | 0.000010|
| lr scheduler factor         | N/A       | N/A       | N/A           | 0.2     |
| best threshold              | N/A       | N/A       | N/A           | 0.0001  |
| clipping threshold          | N/A       | N/A       | N/A           | 10      |
A.2 Experiments

A.2.1 MNIST CNN Experiments

![Accuracy curves for MNIST CNN under label flipping attacks at different attacker ratios.](image)

Figure 5: Accuracy curves of different strategies for MNIST CNN under label flipping attacks at different attacker ratios.

![Accuracy curves for MNIST CNN under Byzantine attacks at different attacker ratios.](image)

Figure 6: Accuracy curves of different strategies for MNIST CNN under Byzantine attacks at different attacker ratios.
A.2.2 Fashion-MNIST Experiments

Figure 7: Accuracy curves of different strategies for Fashion-MNIST under label flipping attacks at different attacker ratios.

Figure 8: Accuracy curves of different strategies for Fashion-MNIST under Byzantine attacks at different attacker ratios.
A.2.3 CIFAR10 Experiments

Figure 9: Accuracy curves of different strategies for CIFAR10 under label flipping attacks at different attacker ratios.

Figure 10: Accuracy curves of different strategies for CIFAR10 under Byzantine attacks at different attacker ratios.
A.2.4 Number of Reliable and Outlier Participants for CIFAR10 Experiments

Figure 11: Number of participants marked as reliable and outlier in CIFAR10 label flipping attacks.

Figure 12: Number of participants marked as reliable and outlier in CIFAR10 Byzantine attacks.
## A.3 GPU Run Time

Table 12: Detailed GPU runtimes.

|                  | CIFAR10 | Fashion MNIST | MNIST | Total |
|------------------|---------|---------------|-------|-------|
| **Tesla P100-16GB** | 250.2   | 82.2          | 104.3 | 436.7 |
| Byzantine Attacks | 51.6    | 61.5          | 52.9  | 166.0 |
| Label Flipping Attacks | 198.6  | 20.7          | 51.4  | 270.7 |
| **Tesla V100-16GB** | 251.8   | 145.2         | 128.5 | 525.5 |
| Byzantine Attacks | 149.4   | 63.5          | 46.3  | 259.2 |
| Label Flipping Attacks | 102.4  | 81.7          | 82.2  | 266.3 |
| **Total**        | 502.0   | 227.4         | 232.8 | 962.2 |