Mathematical Biosciences and Engineering

MBE, 19(2): 1659–1676.
DOI: 10.3934/mbe.2022078
Received: 17 September 2021
Accepted: 29 November 2021
Published: 14 December 2021

http://www.aimspress.com/journal/MBE

Research article

Byzantine-robust federated learning via credibility assessment on non-IID data

Kun Zhai, Qiang Ren, Junli Wang and Chungang Yan*

Key Laboratory of Embedded System and Service Computing (Tongji University), Ministry of Education, Shanghai 201804, China

* **Correspondence:** Email: yanchungang@tongji.edu.cn.

Abstract: Federated learning is a novel framework that enables resource-constrained edge devices to jointly learn a model, which solves the problem of data protection and data islands. However, standard federated learning is vulnerable to Byzantine attacks, which will cause the global model to be manipulated by the attacker or fail to converge. On non-iid data, the current methods are not effective in defensing against Byzantine attacks. In this paper, we propose a Byzantine-robust framework for federated learning via credibility assessment on non-iid data (*BRCA*). Credibility assessment is designed to detect Byzantine attacks by combing adaptive anomaly detection model and data verification. Specially, an adaptive mechanism is incorporated into the anomaly detection model for the training and prediction of the model. Simultaneously, a unified update algorithm is given to guarantee that the global model has a consistent direction. On non-iid data, our experiments demonstrate that the *BRCA* is more robust to Byzantine attacks compared with conventional methods.

Keywords: Byzantine robust; federated learning; adaptive anomaly detection; non-iid; computer vision

1. Introduction

In recent years, the abundance of data generated from many distributed devices with the popularity of smartphones, wearable devices, intelligent home appliances, and autonomous driving. These data are usually concentrated in the data center for effective use. However, a crucial issue

arises that the concentrated data store causes leakage of personal privacy [1]. Simultaneously, as the computing power of these mobile devices increases, it is attractive to store data locally while completing related computing tasks. Federated learning is a distributed machine learning framework that allows multiple parties to collaboratively train a model without sharing raw data [2,3], which has attracted significant attention from industry and academia recently. [4] summarizes and discusses in the application of federated learning in big data and its future direction. Although federated learning has essential significance and advantages in protecting user privacy, it also faces many challenges.

First of all, due to the distributed nature of federated learning, it is vulnerable to Byzan- tine attacks. Notably, it has been shown that, with just one Byzantine client, the whole federated optimization algorithm can be compromised and fail to converge [5]. Especially when the training data is not independent and identically distributed (non-iid), the difficulty of defense against Byzantine attacks is increased and it is difficult to guarantee the convergence of the model [6].

Methods for defending against Byzantine attacks in federated learning have been exten- sively studied, including coordinate-wise trimmed mean [9], the coordinate- wise median [7,8], the geometric median [10,11], and distance-based methods Krum [12], BREA [6], Bulyan [5]. In addition to the above methods based on statistical knowledge, [14] proposes a new idea based on anomaly detection to complete the detection of Byzantine clients in the learning process. [13] discusses the challenges and future directions of federated learning in real-time scenarios in terms of cybersecurity.

The above methods can effectively defend against Byzantine attacks to some extent, but there are also some limitations. First, the methods based on statistical knowledge have high computational complexity, and also their defense abilities are weakened due to the non-iid data in federated learning. Second, for the anomaly detection algorithm [14], there is a premise that the detection model should be trained on the test data set. Obviously, the premise hypothesis cannot be realized in practical applications because it is difficult for us to get such a data set, which can cover almost all data distributions. Therefore, it necessary for the anomaly detection model to get pre-training without relying on test dataset and update dynamically on non-iid data.

In this paper, we propose a new method that each client needs to share some data with the server, which makes a trade-off between client privacy and model performance. Unlike FedAvg [2], we use credibility score as the weight of model aggregation, not the sample size. The credibility score of each client is obtained by integrating the verification score and the detection score. The former is calculated by sharing data.

The main contributions of this paper are:

▪We propose a new federated learning framework (BRCA) which combines credibility assessment and unified update. BRCA not only effectively defends against Byzantine attacks, but also reduces the impact of non-iid data on the aggregated global model.

▪The credibility assessment combing anomaly detection and data verification effectively detects Byzantine attacks on non-iid data.

▪By incorporating an adaptive mechanism and transfer learning into the anomaly detection model, the anomaly detection model can dynamically improve detection performance. Moreover, its pre-training no longer relies on the test data set.

▪We customize four different data distributions for each data set, and explore the influence of data distribution on defense methods against Byzantine attacks.

2. Related work

FedAvg is firstly proposed in [2] as an aggregation algorithm for federated learning. The server updates the global model by a weighted average of the clients' model updates, and the aggregation weight is determined based on its data sample size. Stich [15] and Woodworth et al. [16] analyze the convergence of FedAvg on strongly-convex smooth loss functions. However, they assume that the data is iid, which is not suitable for federated learning [17,18]. And Li et al. [19] makes the first convergence analysis of FedAvg when the data is non-iid. [20] uses clustering to improve federated learning in non-iid data. Regrettably, the ability of naive FedAvg is very weak to resist Byzantine attacks.

In the iterative process of federated aggregation, honest clients send the true model updates to the server, wishing to train a global model by consolidating their private data. However, Byzantine clients attempt to perturb the optimization process [21]. Byzantine attacks may be caused by some data corruption events in the computing or communication process such as software crashes, hardware failures and transmission errors. Simultaneously, they may also be caused by malicious clients through actively transmitting error information, in order to mislead the learning process [21].

Byzantine-robust federated learning has received increasing attention in recent years. Krum [12] is designed specially to defend Byzantine attacks in the federated learning. Krum generate the global model by a client's model update whose distances to its neighbors is shortest. GeoMed [10] uses the geometric median which is a variant of the median from one dimension to multiple dimensions. Unlike the Krum, the GeoMed uses all client updates to generate a new global model, not just one client update. Trimmed Mean [9] proposes that each dimension of its global model is obtained by averaging the parameters of clients' model updates in that dimension. But before calculating the average, the largest and smallest part of the parameters in that dimension are deleted, Xie et al. [22] and Mhamdi et al. [5] are all its variants. BREA [6] also considers the security of information transmission, but its defense method is still based on distance calculation. Zero [23] based on Watermark detection approach detect attacks such as malware and phishing attacks and cryptojacking. [24] surveys intrusion detection techniques in mobile cloud computing environment.

Table 1. The summary of the contributions and limitations of the related papers.

Reference	Contributions	Limitations
[12] [10] [9] [5] [22]	Krum, GeoMed and Trimmed Mean complete the Byzantine defense based on statistical knowledge. Easy to deploy applications.	The assumption is that the data of the clients is iid. High computational complexity.
[25]	The auto-encoder anomaly detection model is firstly applied to detect Byzantine attacks.	The pre-training of the anomaly detection model is completed on test dataset. The anomaly detection model is static.
[6]	Cryptography is used to protect the security of information transmitted between clients and server.	Defense against Byzantine attacks is still based on distance to find outliers, and had limited defenses capabilities.

All of the above defense methods based on statistical knowledge and distance are not effective in defending against Byzantine attacks in non-iid settings. Abnormal [25] uses an anomaly detection model to complete the detection of Byzantine attacks.

The concept of independent and identically distributed (iid) of data is clear, but there are many meanings of non-iid. In this work, we only consider label distribution skew [17]. The categories of samples may vary across clients. For example, in the face recognition task, each user generally has their face data; for mobile device, some users may use emojis that do not show up in others' devices.

We summarize the contributions and limitations of the existing works in Table 1.

In this paper, we propose a method that combine credibility assessment and unified update to robust federated learning against Byzantine attacks on non-iid data.

3. Byzantine-robust federated learning on non-iid data

We utilize a federated setting that one server communicates with many clients. For the rest of the paper, we will use the following symbol definitions: A is the total client set, $|A|$ = n; S is the selected client set in every iteration, $|S|$ = k; among them, B is Byzantine client set, $|B|$ = b, and H is honest client set, $|H| = h$. w_i^t is the model update sent by the client i to the server at round t, Byzantine attack rate $\xi = \frac{b}{k}$. w^t is the global model at round t, D_P = {D , ..., D_n} is clients' private data, Ds = {Ds, ..., Ds } is the clients' shared data, and data-sharing rate $\gamma = \frac{|D_s|}{|D_P|+|D_s|}$ ($|\cdot|$ represents the sample size of the data set).

3.1. BRCA: Byzantine-robust federated learning via credibility assessment

In order to enhance the robustness of federated learning against Byzantines attacks on non-iid data, *BRCA* combines credibility assessment and unified update, Figure 1 depicts the architecture of *BRCA*.

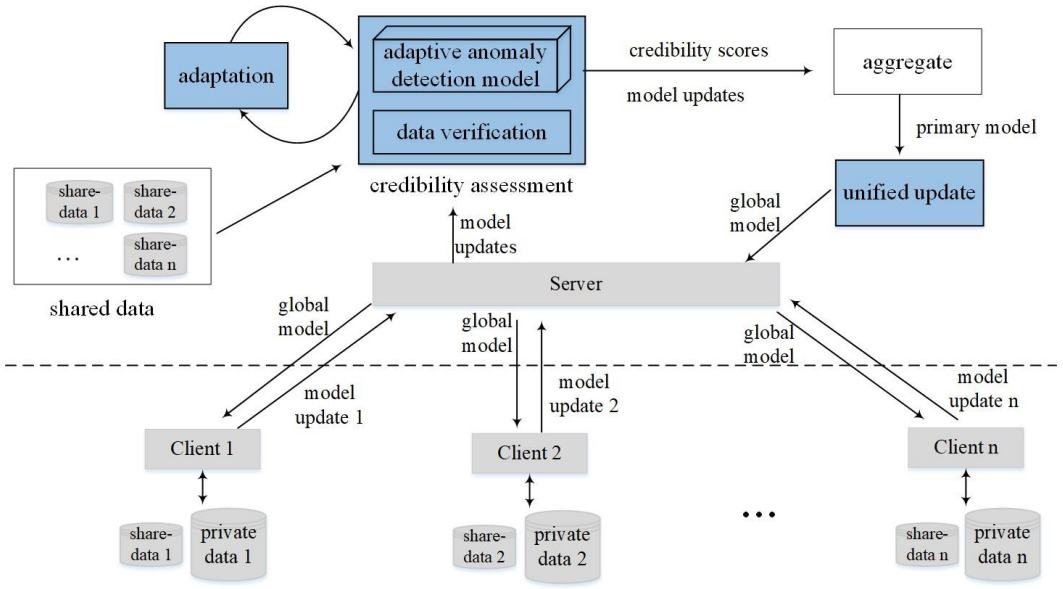


Figure 1. The frame diagram of the BRCA.

Before training, each client needs to share some private data to the server. In each iteration, the server randomly selects some clients and sends the latest global model to them. These clients use their

private data to train the model locally and send the model updates to the server. After receiving model updates, the server conducts a credibility assessment for each model update and calculates their credibility scores. Momentum is an effective measure to improve the ability of federated learning to resist Byzantine attacks [26]. So our aggregation Eq (1) is as follow:

$$w^t + 1 = \alpha W^t + (1 - a) \sum_{i \in s} r_i^t w_i^t \tag{1}$$

where r_i^t is the credibility score of client i at round t and α $(0 < \alpha < 1)$ is a decay factor. Last, unified update uses shared data to update the primary global model to get the new global model for this round

Algorithm 1 is the description of *BRCA*, which contains *Credibility Assessment* in line 22, and *Unified Update* in line 28. The crucial of *BRCA* to defend against Byzantine attacks is credibility assessment. On non-iid data, the data distributions of different clients are immense, and it is difficult to judge whether the difference is caused by Byzantine attacks or the non-iid data. However, the model update of the honest client should have a positive effect on its private data, which is not affected by other clients. Simultaneously, anomaly detection model can effectively detect Byzantine attacks [25]. Thus, we combine the above two ideas to detect Byzantine attacks. In order to solve the shortcomings of the existing anomaly detection models, we propose an adaptive anomaly detection model. In this paper, the shared data is randomly selected by each client based on the sample category. Of course, other sampling methods could also be used, such as clustering. In addition, it must be pointed out that the shared data will only be used on the server, not on the clients. That effectively protect the clients' privacy.

To summarize, *BRCA* has five steps. First: the server pre-train an anomaly detection model by source data and initialize a global model. Second: every client share little private data with the server. Three: every client download the newest global model from the server, and complete model updates by private data. Then, every client send the model update to the server. Four: the server update the global model and complete the adaptation of the anomaly detection model by model updates from clients. Five: the server update the primary global model with unified update, after that, the new global model is completed. Repeating steps three to five until the global model converges

Our work is different from the recent state of the art. First, *Krum*, *GeoMed* and *TrimmedMean* are the representative methods based on geometric knowledge, but their premise is that the data of clients is dependent and identically distributed (iid). The hypothesis of our method is based on the actual application background of FL, aiming at non-iid data. Second, *Abnormal* is the first method to detect Byzantine attacks by auto-encoder anomaly detection model. However, the training of the anomaly detection model in the method is based on the test dataset and the abnormal detection model in the method is static. For both of the problems, our method has made improvement: 1) we pre-train the anomaly detection model with related but different source data without relying on the test dataset. 2) we introduce adaptive mechanism to the anomaly detection model, which help the detection model get update during federated iteration dynamically.

3.2. Credibility assessment

Algorithm 2(*Credibility Assessment*) is the key part of *BRCA*, which assigns a credibility score for each client model update. A Byzantine client would be given much lower credibility score than an honest client. To guarantee the accuracy of the credibility score, *Credibility Assessment* integrates adaptive anomaly detection model and data verification.

Algorithm 1: BRCA

Input: total clients A; total number of iterations T; learning rate η_{server}, η_{client}, $\eta_{detection}$; Byzantine attack rate ξ; epoch E_{server}, E_{client}; initial global model w^0; clients' private data $D_P = \{D_1^P, \ldots, D_N^P\}$; clients' shared data $D_s = \{D_1^s, \ldots, D_n^S\}$; initial anomaly detection model θ^0; β; α; d;k

Output: global model W^{T+1}, anomaly detection model θ^{T+1}

$R = \emptyset$: the credibility score set.

$H = \emptyset$: the honest client set.

Function Add Attack(w)**:**

  return w attacked by Byzantine client.

End Function

for *each round* t=*0 to* T **do**

  Clients:

  for *client* $i \in S$ ***parallel*** **do**

    for *each epoch* $e = 0$ *to* E_{client} **do**

      $w_i^t = w^t - \eta_{client} \nabla_l \left(D_i^P, W^t\right)$,where l is the loss function.

    end

    if $i \epsilon$ *top* ξ *percent of* S **then**

      W_i^t = Add Attack (W_i^t)

    end

    send W_i^t back to the server

  end

  Server:

  sample $S \in A$ randomly

  broadcast latest global model w^t to each client $j \in S$

  receive model updates from clients $Q = \{W_1^t, \ldots, W_j^t \ldots, W_k^t\}$, client $j \in S$

  R, H, θ^{T+1} = ***Credibility Assessment*** $\{Q, D_s, \theta^t, \beta, S, \eta_{detection}, d, k\}$

  if $H > \frac{1}{2}k$ **then**

    $$\boldsymbol{w^{t+1} = \alpha w^t + (1-\alpha) \sum_{client\ j \in S} r_j^t * w_j^t}\ ,\ \boldsymbol{r_j^t \in R}$$

  else

    $\boldsymbol{w^{t+1} = w^t}$

  end

  $\boldsymbol{w^{t+1}}$ = ***Unified Update*** $(\boldsymbol{w^{t+1}}, D_s, E_{server}, \eta_{server}, H)$

end

return global model $\boldsymbol{w^{t+1}}$, anomaly detection model θ^{T+1}

In Algorithm 2, line 4 is the *data verification*, which calculates the verification score f_i for the model update of client i. And line 5 is the *get-anomaly-score()* of the adaptive anomaly detection model, which calculates detection score e_i. Subsequently, the credibility r_i of the model update is $r_i = \beta e_i + (1 - \beta) f_i$, $R = \{r_1, \ldots r_i \ldots, r_k\}$, client $i \in S$. The *make-adaption ()* in line 24 implements the adaption of the anomaly detection model.

In this paper, we judge the model update with a credibility score lower than the mean of R as a Byzantine attack, and set its credibility score as zero. Finally, normalizing the scores to get the final credibility scores.

3.2.1 Adaptive anomaly detection model

In the training process, we cannot predict the type of attacks, but we can estimate the model update of the honest client. Therefore, we can adopt a one-class classification algorithm to build the anomaly detection model with normal model updates. Such technique will learn the distribution boundary of the model updates to determine whether the new sample is abnormal. Auto-encoder is an effective one-class learning model for detecting anomalies, especially for high-dimensional data [27].

In practical applications, we cannot get the target data to complete the pre-training of our anomaly detection model. Therefore, the initialized anomaly detection model will be pre-trained on the source data with the idea of transfer learning.

At round t, the detection score e_i^t of client i:

$$e_i^t = \exp\left(\frac{Mse(C_i^t - \theta^t(C_i^t)) - \mu(E)}{\sigma(E)}\right) \tag{2}$$

Our anomaly detection model is different from the one in Abnormal: 1) Abnormal uses the test set of the data set to train the anomaly detection model. Although the detection model obtained can complete the detection task very well, in most cases the test data set is not available. Therefore, based on the idea of transfer learning, we complete the pre-training of the anomaly detection model in the source domain. 2) *Abnormal* 's anomaly detection model will not be updated after training on the test set. We think this is unreasonable, because the test set is only a tiny part of the overall data. Using a small part of the training data to detect most of the remaining data, and the result may not be accurate enough. Therefore, pre-training of the anomaly detection model is completed in the source domain. Then we use the data of the target domain to fine-tune it in the iterative process to update the anomaly detection model dynamically, as *make-adaption* shown in Algorithm 3.

Algorithm 2: Credibility Assessment

Input: local model updates Q; clients' shared data $D_s = \{D_1^s, \ldots, D_n^s\}$; anomaly detection model θ^t; β; selected clients S; $\eta_{detection}$; d;k

Output: credibility score of clients R; honest client set H; anomaly detection model θ^{t+1}

$R = \emptyset$: credibility score set; $H = \emptyset$: the honest client set; $sum = 0$; $sum_e = 0$; $sum_f = 0$

$C = \{C_1^t, \ldots, C_i^t, \ldots C_k^t\}$, client$i \in S$, $c_i^t$is the weight of the last convolutional layer of W_i^t

for *each client* $i \in S$ **do**

  Data Verification: **compute** f_i^t with $W_i^t$and D_i^s, client $i \in S$ base on equation 3 and equation 4

  e_i^t= ***AADM. get-anomaly-score*** (θ^t, C)

end

$sum_e = \sum_{client\ i \in S} e_i^t$; $sum_f = \sum_{client\ i \in S} f_i^t$

for *each client* $i \in S$ **do**

  $e_i^t = e_i^t / sum_e$; $f_i^t = f_i^t / sum_f$

  $r_i^t = \beta e_i^t + (1-\beta) f_i^t$; $R = R \cup \{r_i^t\}$

end

$M(R)$ is the mean of R

for *each* $r_i^t \in R$ **do**

  if $r_i^t < M(R)$ **then**

    $r_i^t = 0$

  else

    $H = H \cup \{i\}$

  end

  $sum += r_i^t$

end

for each $r_i^t \in R$ **do**

  $r_i^t = r_i^t / sum$

end

θ^{t+1} = ***AADM. make-adaption*** $(H, \theta^t, \eta_{detection}, C, d, k)$

return R, H, θ^{t+1}.

3.2.2 Data verification

The non-iid of client data increases the difficulty of Byzantine defense. However, the performance of the updated model of each client on its shared data is not affected by other clients, which can be effectively solved this problem. Therefore, we use the clients' shared data $\{D_S = D_1^s, \ldots D_i^s, \ldots, D_k^s\}$

Algorithm 3: AADM adaptive anomaly detection model

Input: anomaly detection model θ^t; weights of the last convolutional layer of the local model C; $\eta_{detection}$; credibility score R; honest client set H;d;k

Output: updated anomaly detection model θ^{t+1}

Function get-anomaly-score (θ^t, C_i^t)**:**
  compute e_i^t with θ^t and c_i^t, client $i \in S$ based on Eq (2)
  return e_i^t.
End Function

Function make-adaption (H, θ^t, $\eta_{detection}$,C, d)**:**
  if $|H| < 1/2$ k **then**
    $\theta^{t+1} = \theta^t$
  else
    H_0 is a subset of H obtained by removing the clients whose credibility score is in the largest and smallest d fraction.
    for *client* $i \in H_0$ **do**
      $\theta^{t+1} = \theta^t - \eta_{detection} \nabla(\theta^t, c_i^t)$
    end
  end
  return updated anomaly detection model θ^{t+1}
End Function

client $i \in S$ to calculate the verification score of their updated model:

$$f_i^t = \left(exp\left(\frac{l_i^t - \mu(l)}{\sigma(L)}\right)\right)^{-2} \tag{3}$$

where l_i^t is loss of client i calculated on model w_i^t using the shared data D_i^s at round t:

$$l_i^t = \frac{1}{|D_i^s|}\sum_{j=0}^{|D_i^s|} l\left(D_i^{s(j)}, W_i^t\right) \tag{4}$$

where $D_i^{s(j)}$ is the $j-th$ sample of D_i^s and $\mu(L)$, $\sigma(L)$ are the mean and variance of set $L = \{l_1, \ldots, l_k\}$ respectively.

3.3. Unified update

After getting the credibility score r_t^k in Algrithm 2 with the anomaly score e_t^k and the verification score f_t^k, we can complete the aggregation of the clients' local model updates in Eq (1) and get a preliminary updated global model. However, due to the non-iid of client data, the knowledge learned by the local model of each client is limited, and the model differences between two clients are also significant. Therefore, to solve the problem that the preliminary aggregation model lacks a clear

and consistent goal, we introduce an additional *unified update* procedure with shared data on server, details can be seen in Algorithm 4.

Because the data used for the *unified update* is composed of each client's data, it can more comprehensively cover the distribution of the overall data. The goal and direction of the *unified update* are based on the overall situation and will not tend to individual data distribution.

Algorithm 4: Unified update

Input: global model w^{t+1}; clients' shared data $D_s = \{D_1^s, \ldots, D_n^s\}$; E_{server};η_{server}; honest client set H

Output: global model w^{t+1}.

```
for each epoch e = 0 to E_server do
    for i ∈ H do
        w^{t+1} = w^{t+1} − η_server ∇l(D_i^s, w^{t+1})
    end
end
return global model w^{t+1}.
```

4. Experiments

To verify the effectiveness of *BRCA*, we structure the client's data into varying degrees of non-iid, and explore the impact of different amounts of shared data on the global model. At the same time, we also compare the performance of our anomaly detection model with the Abnormal 's and explore the necessity of unified update.

4.1. Experimental steup

4.1.1. Datasets

Mnist and Cifar10 are the two most commonly used public data sets in image classification, and most of the benchmark methods in our work also use these two data sets for experiments. Using these two data sets, it is easier to compare with other existing methods.

We do the experiments on Mnist and Cifar10, and customize four different data distributions: (a) non-iid-1: each client only has one class of data. (b) non-iid-2: each client has 2 classes of data. (c) non-iid-3: each client has 5 classes of data. (d) iid: each client has 10 classes of data.

For Mnist, using 100 clients and four data distributions: (a) non-iid-1: each class of data in the training dataset is divided into 10 pieces, and each client selects one piece as its private data. (b) non-iid-2: each class of data in the training dataset is divided into 20 pieces, and each client selects 2 pieces of different classes of the data. (c) non-iid-3 each class of data in the training dataset is divided into 50 pieces, and each client selects 5 pieces of different classes of the data. (d) iid: each class of data in the training dataset is divided into 100 pieces, and each client selects 10 pieces of different classes of the data. As for the source domains used for the pre-training of the anomaly detection model, we randomly select 20,000 lowercase letters in the Nist dataset.

For Cifar10, there are 10 clients and the configuration of four data distributions is similar to that of the Mnist. We select some classes of data in Cifar100 as source domain, which are as follows: lamp

(number:40), lawn mower (41), lobster (45), man (46), forest (47), mountain (49), girl (35), Snake (78), Rose (70) and Tao (68), these samples do not exiting in Cifar10.

4.1.2. Models

We use logistic regression on Mnist dataset. η_{server}= 0.1, η_{client} = 0.1, $\eta_{detection}$= 0.02, E_{client}= 5, E_{server}= 1, n = 100, k = 30, ξ = 20%. Two convolution layers and three fully connected layer on Cifar10, η_{server} = 0.05, η_{client} = 0.05, $\eta_{detection}$= 0.002, E_{client}= 10, E_{server} = 10, n = 10, k = 10, ξ = 20%. The structure of models are the same as [10].

4.1.3. Benchmark byzantine attacks

Same-value attacks: A Byzantine client i sends the model update $\omega_i = c1$ to the server (1 is all-ones vectors, c is a constant), we set c = 5. Sign-flipping attacks: In this scenario, each client i computes its true model update ω_i, then Byzantine clients send ω_i = aω_i (a < 0) to the server, we set $a = -5$. Gaussian attacks: Byzantine clients add Gaussian noise to all the dimensions of the model update $\omega_i = \omega_i + \epsilon$, where s follows Gaussian distribution N (0, g^2) where g is the variance, we set g = 0.3.

4.1.4. Benchmark defense methods

Defenses: *Krum*, *GeoMed*, *Trimmed Mean*, *Abnormal* and *No Defense*. *No Defense* does not use any defense methods.

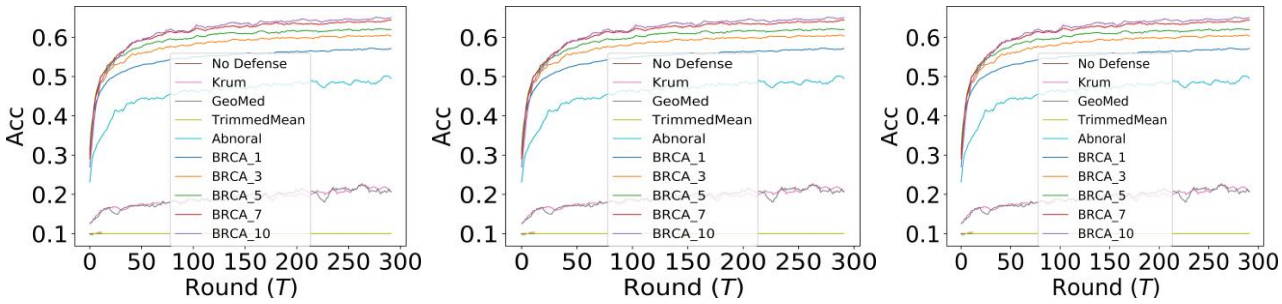
4.2. Result and discussion

4.2.1. Impact of shared data rate

In the first experiment, we test the influence of the shared data rate γ in our algorithm, and do the experiment with the data distribution of non-iid-2. We implement it on five different values [1, 3, 5, 7 and 10%]. Figures 2 and 3 are the accuracy and loss for Cifar10. It is found that: 1) In all cases of Byzantine attacks, our algorithm is superior to the three benchmark methods. 2) Only 1% of the data shared by the client can significantly improve the performance of the global model. For three Byzantine attacks, *Krum*, *GeoMed*, *Trimmed Mean*, *No Defense* are all unable to converge. This also shows that when the model is complex, such methods would be less able to resist Byzantine attacks.

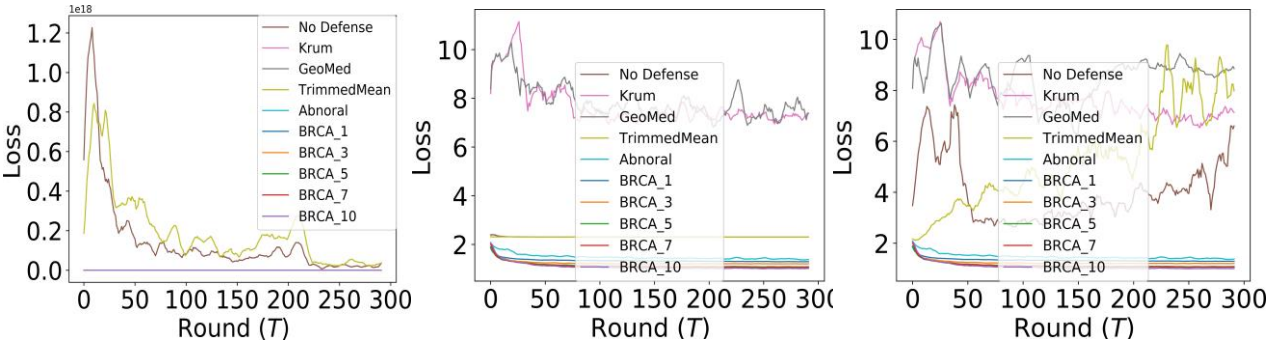
With the increase in the client data sharing ratio, the performance of the global model has become lower. When the client shares the data ratio from 1 to 10%, the average growth rate with the three Byzantine attacks are: 1.8→1.41→0.97→0.92%. The clients only share one percent of the data, and the performance of the global model can be greatly improved.

Figure 4 clearly demonstrates the impact of different shared data rates on the loss value of the global model on Cifar10.



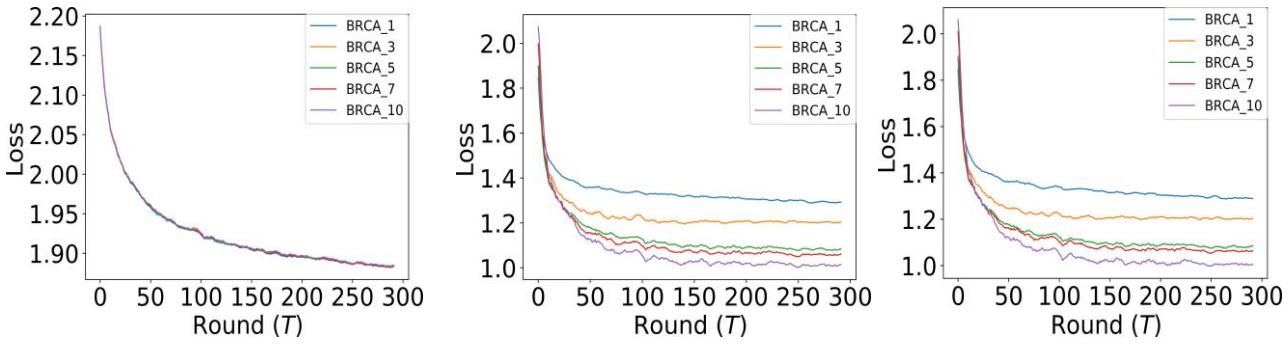
(a) Same value (b) Sign flipping (c) Gaussian noisy

Figure 2. The Accuracy of Cifar10. Byzantine attack types from (a) to (c) are as follows: *Same value*, *Sign flipping* and *Gaussian noisy*. Six defense methods are adopted for each type of attack, in order: *No defense*, *Krum*, *GeoMed*, *Trimmed Mean*, *Abnormal* and *BRCA*. For Ours, there are five different shared data rate (1, 3, 5, 7 and 10%), which correspond accordingly: *BRCA 1*, *BRCA 3*, *BRCA 5*, *BRCA 7*, *BRCA 10*.



(a) Same value (b) Sign flipping (c) Gaussian noisy

Figure 3. The loss of Cifar10. The legends are the same as Figure 2.



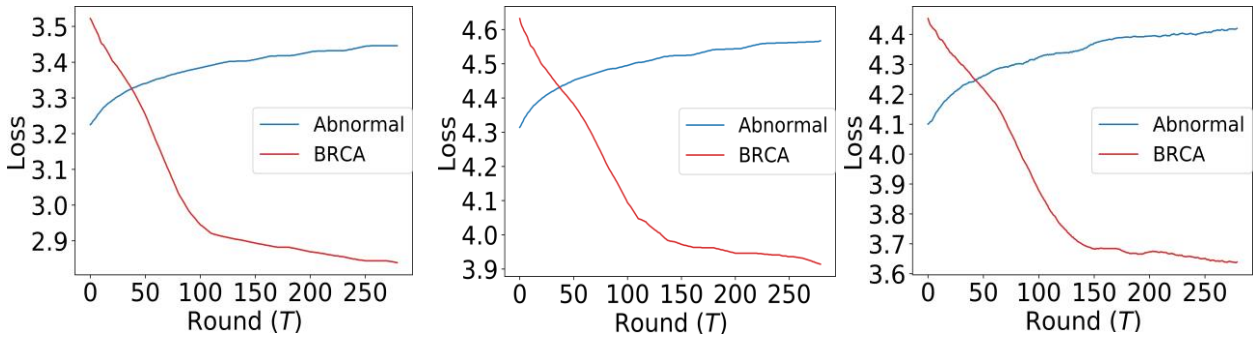
(*a*) Same value (*b*) Sign flipping (*c*) Gaussian noisy

Figure 4. The loss of BRCA on Cifar10 with five different shared rate.

4.2.2. Performance of anomaly detection model

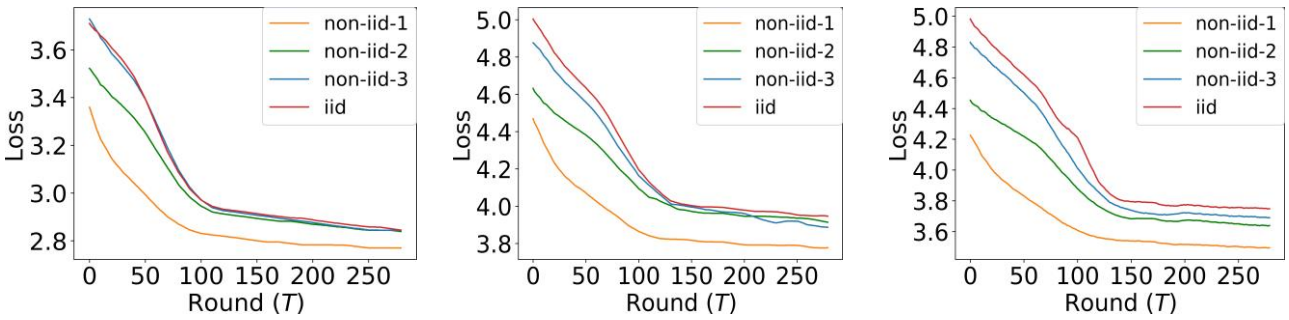
In this part, the purposes of our experiment are: 1) Compare anomaly detection model between ours and *Abnormal*. 2) Explore the robustness of the anomaly detection model to data that are non-iid. The shared data rate γ is 5%, Sections 4.2.3 and 4.2.4 are the same.

In order to compare the detection performance of the anomaly detection model against Byzantine attacks between *BRCA* and *Abnormal*, we use the cross-entropy loss as the evaluation metric which is calculated by the detection score. Firstly, we get detection scores $E = \{e_1, ..., e_i, ..., e_k\}$ based on model update ω_i and θ, client $i \in S$. Then, we set $P = Sigmoid(E - \mu(E))$ represents the probability that the client is honest and $1 - P$ is the probability that the client is Byzantine. Lastly, we use P and true label $Y\ (y^i = 0,\ i \in B\ and\ y^i = 1,\ j \in H)$ to calculate the cross-entropy loss $l = \Sigma_{i=1}^{k} y_i\, ln(P_i)$



(a) Same value (b) Sign flipping (c) Gaussian noisy

Figure 5. the cross-entropy loss of our and Abnormal anomaly detection model, on Cifar10 with non-iid-2. (a)–(c) are the performance for three Byzantine-attacks.



(*a*) Same value (*b*) Sign flipping (*c*) Gaussian noisy

Figure 6. (a)–(c) are our anomaly detection model's performance on four different data distribution (iid, non-iid-1, non-iid-2, non-iid-3) against Byzantine attacks (*Gaussian noisy*, *sign flipping*, *same value*).

Figure 5(a)–(c) compare the loss of the anomaly detection model between *BRCA* and the *Abnormal*. From the figures, we can see that our model has a greater loss than *Abnormal* in the initial stage, mainly due to the pre-training of the anomaly detection model using the transfer learning. The

initial pre-trained anomaly detection model cannot be used well in the target domain. As the adaptation progress, the loss of our model becomes decreases and gradually outperforms the Abnormal. Although *Abnormal* has a low loss in the initial stage, as the training progresses, the loss gradually increases, and the detection ability becomes degenerate.

Figure 6(a)–(c) show the influence of different data distributions on our detection model. For different data distributions, the detection ability of the model is different, but it is worth pointing out that: as the degree of non-iid of the data increases, the detection ability of the model also increases.

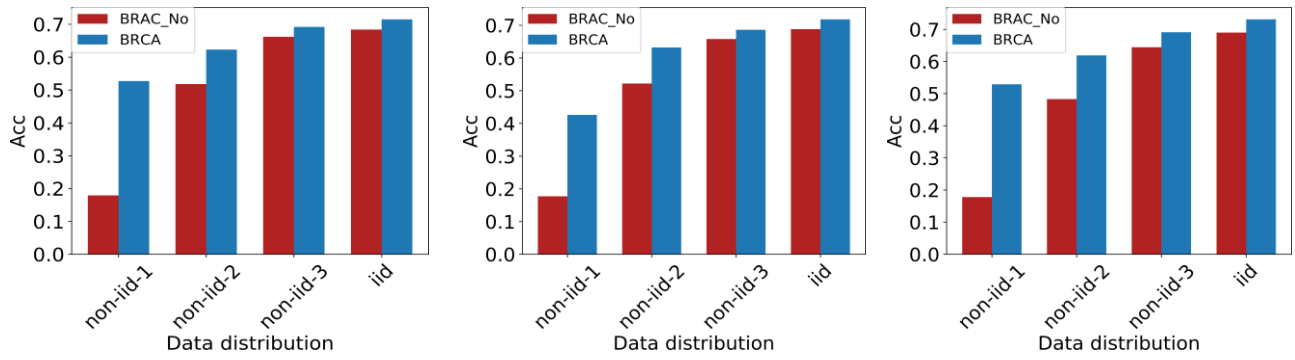
4.2.3. Impact of unified update

In this part, we study the impact of the unified update on the global model. Figure 7 shows the accuracy of the global model with and without unified update on Cifar10.

From non-iid-1 to iid, the improvement of the global model's accuracy by unified update is as follows: 35.1→13.6→4.7→2.3% (*Same value*), 34.8→10.5→3.0→3.1% (*Gaussian noisy*), 24.9→9.9→2.8→3.0% (*Sign flipping*). Combined with Figure 7, it can be clearly found that the more simple the client data is, the more obvious the unified update will be to the improvement of the global model.

When the data is non-iid, the directions of the model updates between clients are different. The higher the degree of non-iid of data, the more significant the difference. The global model obtained by weighted aggregation does not fit well with the global data. Unified update on the shared data can effectively integrate the model updates of multiple clients, giving the global model a consistent direction.

Therefore, it is necessary to implement a unified update to the primary aggregation model when data is non-iid.



(a) Same value (*b*) Sign flipping (*c*) Gaussian noisy

Figure 7. The accuracy of BRCA and BRCA No on Cifar10. BRAC No is based on BRCA with unified update removed.

4.2.4. Impact of non-iid

Tables 2 and 3 show the accuracy and loss of each defense method under different data distributions on Cifar10. It can be seen that our method is the best, and the performance is relatively

stable for different data distributions. The higher the degree of non-iid of data, the more single the data of each client, the lower the performance of the defense method.

Table 2. The accuracy of the six defenses under four different data distributions on Cifar10, against three attacks.

Attacks	Defenses / Distri	No	Krum	GeoMed	Abnormal	TrimmedMean	BRCA
Same value	Non-iid-1	0.1	0.1	0.1	0.178	0.1	**0.529**
	Non-iid-2	0.101	0.207	0.205	0.480	0.1	**0.619**
	Non-iid-3	0.1	0.398	0.398	0.634	0.1	**0.691**
	iid	0.098	0.696	0.705	0.698	0.101	**0.713**
Gaussian noisy	Non-iid-1	0.1	0.1	0.1	0.178	0.1	**0.529**
	Non-iid-2	0.191	0.204	0.205	0.513	0.059	**0.623**
	Non-iid-3	0.0409	0.398	0.394	0.660	0.171	**0.692**
	iid	0.1	0.697	0.694	0.710	0.120	**0.715**
Sign flipping	Non-iid-1	0.1	0.101	0.1	0.177	0.1	**0.426**
	Non-iid-2	0.1	0.192	0.214	0.5131	0.1	**0.621**
	Non-iid-3	0.1	0.397	0398	0.651	0.1	**0.686**
	iid	0.1	0.697	0.703	0.711	0.1	**0.718**

Table 3. The loss of the six defenses under four different data distributions on Cifar10, against three attacks.

Attacks	Defenses / Distri	No	Krum	GeoMed	Abnormal	TrimmedMean	BRCA
Same value	Non-iid-1	$2.84e^{16}$	11.72	9.61	2.29	$6.05e^{17}$	**2.09**
	Non-iid-2	$6.99e^{16}$	7.29	8.01	2.06	$3.63e^{16}$	**2.09**
	Non-iid-3	$4.48e^{16}$	2.35	2.38	1.893	$3.37e^{16}$	**0.691**
	iid	$1.51e^{16}$	0.794	0.774	1.837	$3.17e^{16}$	**1.79**
Gaussian noisy	Non-iid-1	$8.635e^{4}$	8.41	9.37	2.29	936.17	**1.54**
	Non-iid-2	9.51	7.57	8.37	1.34	7.98	**0.623**
	Non-iid-3	8.22	2.01	2.31	0.94	6.07	**0.692**
	iid	8.09	0.81	0.79	0.82	3.12	**0.76**
Sign flipping	Non-iid-1	2.30	10.72	9.91	2.29	2.30	**1.54**
	Non-iid-2	2.31	7.77	7.10	1.34	2.30	**0.621**
	Non-iid-3	2.31	2.36	2.13	0.94	2.30	**0.686**
	iid	2.31	0.79	0.80	0.81	2.31	**0.76**

Our analysis is as follows: 1) The non-iid of data among clients causes large differences between clients' models. And it is difficult for the defense method to judge whether the anomaly is caused by the non-iid of the data or by the Byzantine attacks, which increases the difficulty of defending the Byzantine attack. 2) *Krum* and *GeoMed* use statistical knowledge to select the median or individual client's model to represent the global model. This type of method can effectively defend against Byzantine attacks when the data is iid. However when the data is non-iid, each client's model only focuses on a smaller area, and its independence is high, cannot cover the domain of other clients, and obviously cannot represent the global model. 3) *Trimmed Mean* is based on the idea of averaging to defend against Byzantine attacks. When the parameter dimension of the model is low, it has a good performance. But as the complexity of the model increases, the method can not stably complete convergence.

5. Conclusions

In this work, we propose a robust federated learning framework against Byzantine attacks when the data is non-iid. *BRCA* detects Byzantine attacks by *credibility assessment.* Meanwhile, it makes the *unified updating* of the global model on the shared data, so that the global model has a consistent direction and its performance is improved. *BRCA* can make the global model converge very well when facing different data distributions. And for the pre-training of anomaly detection models, transfer learning can help the anomaly detection model get rid of its dependence on the test data set. Experiments have proved that *BRCA* performs well both on non-iid and iid data, especially on non-iid data. In the future, we will improve our methods by studying how to protect the privacy and security of shared data.

Acknowledgments

This work was partially supported by the Shanghai Science and Technology Innovation Action Plan under Grant 19511101300.

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

1. L. Zhou, K. H. Yeh, G. Hancke, Z. Liu, C. Su, Security and privacy for the industrial internet of things: An overview of approaches to safeguarding endpoints, *IEEE Signal Process. Mag.*, **35** (2018), 76–87. doi: 10.1109/MSP.2018.2846297.
2. B. McMahan, E. Moore, D. Ramage, S. Hampson, B. Aguera, Communication-efficient learning of deep networks from decentralized data, in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, **54** (2017), 1273–1282.
3. Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, V. Chandra, Federated learning with non-iid data, preprint, arXiv:1806.00582.

4. T. R. Gadekallu, Q. Pham, T. Huynh-The, S. Bhattacharya, P. K. R. Maddikunta, M. Liyanage, Federated learning for big data: A survey on opportunities, applications, and future directions, preprint, arXiv:2110.04160.
5. E. M. E. Mhamdi, R. Guerraoui, S. Rouault, The hidden vulnerability of distributed learning in Byzantium, in *International Conference on Machine Learning*, (2018), 3521–3530.
6. J. So, B. Guler, A. S. Avestimehr, Byzantine-resilient secure federated learning, *IEEE J. Sel. Areas Commun.*, 2020. doi: 10.1109/JSAC.2020.3041404.
7. X. Chen, T. Chen, H. Sun, Z. S. Wu, M. Hong, Distributed training with heterogeneous data: Bridging median-and mean-based algorithms, preprint, arXiv:1906.01736.
8. H. Yang, X. Zhang, M. Fang, J. Liu, Byzantine-resilient stochastic gradient descent for distributed learning: A lipschitz-inspired coordinate-wise median approach, in *2019 IEEE 58th Conference on Decision and Control (CDC)*, (2019), 5832–5837. doi: 10.1109/CDC40024.2019.9029245.
9. D. Yin, Y. Chen, R. Kannan, P. Bartlett, Byzantine-robust distributed learning: Towards optimal statistical rates, in *International Conference on Machine Learning*, (2018), 5650–5659.
10. Y. Chen, L. Su, J. Xu, Distributed statistical machine learning in adversarial settings: Byzantine gradient descent, in *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, **1** (2017), 1–25. doi: 10.1145/3154503.
11. K. Pillutla, S. M. Kakade, Z. Harchaoui, Robust aggregation for federated learning, preprint, arXiv:1912.13445.
12. P. Blanchard, E. M. E. Mhamdi, R. Guerraoui, J. Stainer, Machine learning with adversaries: Byzantine tolerant gradient descent, in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, (2017), 118–128.
13. M. Alazab, S. P. R M, P. M, P. Reddy, T. R. Gadekallu, Q. Pham, Federated learning for cybersecurity: Concepts, challenges and future directions, *IEEE Trans. Ind. Inf.*, (2021). doi: 10.1109/TII.2021.3119038.
14. S. Li, Y. Cheng, W. Wang, Y. Liu, T. Chen, Learning to detect malicious clients for robust federated learning, preprint, arXiv:2002.00211.
15. S. U. Stich, Local sgd converges fast and communicates little, preprint, arXiv:1805.09767.
16. B. Woodworth, J. Wang, A. Smith, B. McMahan, N. Srebro, Graph oracle models, lower bounds, and gaps for parallel stochastic optimization, preprint, arXiv:1805.10222.
17. P. Kairouz, H B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, et al., Advances and open problems in federated learning, preprint, arXiv:1912.04977.
18. T. Li, A. K. Sahu, A. Talwalkar, V. Smith, Federated learning: Challenges, methods, and future directions, *IEEE Signal Process. Mag.*, **37** (2020), 50–60. doi: 10.1109/MSP.2020.2975749.
19. X. Li, K. Huang, W. Yang, S. Wang, Z. Zhang, On the convergence of fedavg on non-iid data, preprint, arXiv:1907.02189.
20. S. Agrawal, S. Sarkar, M. Alazab, P. K. R. Maddikunta, T. R. Gadekallu, Q. Pham, Genetic CFL: Optimization of hyper-parameters in clustered federated learning, preprint, arXiv:2107.07233.
21. Y. Chen, S. Kar, J. M. Moura, The internet of things: Secure distributed inference, *IEEE Signal Process. Mag.*, **35** (2018), 64–75. doi: 10.1109/MSP.2018.2842097.
22. C. Xie, O. Koyejo, I. Gupta, Generalized byzantine-tolerant sgd, preprint, arXiv:1802.10116.
23. C. Iwendi, Z. Jalil, A. R. Javed, T. R. G, R. Kaluri, G. Srivastava, et al., Keysplitwatermark: Zero watermarking algorithm for software protection against cyber-attacks, *IEEE Access*, **8** (2020), 72650–72660. doi: 10.1109/access.2020.2988160.

24. S. Shamshirband, M. Fathi, A. T. Chronopoulos, Computational intelligence intrusion detection techniques in mobile cloud computing environments: Review, taxonomy, and open research issues, *J. Inf. Sec. Appl.*, **55** (2020), 102582. doi: 10.1016/j.jisa.2020.102582.
25. S. Li, Y. Cheng, Y. Liu, W. Wang, Ti. Chen, Abnormal client behavior detection in federated learning, preprint, arXiv:1910.09933.
26. E. El-Mhamdi, R. Guerraoui, S. Rouault, Distributed momentum for byzantine-resilient learning, preprint, arXiv:2003.00010.
27. M. Sakurada, T. Yairi, Anomaly detection using autoencoders with nonlinear dimensionality reduction, in *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, (2014), 4–11. doi: 10.1145/2689746.2689747.

AIMS Press