Information-agnostic Model Poisoning Attacks against Byzantine-robust Federated Learning

Yan Zhang[1], Yueyao Chen[2], Xiao Tan[1], Dian Shen,[1] Meng Wang[3], and Beilun Wang[1] (✉)

[1] Southeast University, Nanjing, China
{zhang.yan,xtan,dshen,beilun}@seu.edu.cn
[2] DongFang Boiler Co.,Ltd., China
cyueyao@foxmail.com
[3] Tongji University, Shanghai, China
mengwangsd@outlook.com

Abstract. Federated learning (FL), as a popular distributed machine learning paradigm in various applications, is faced with Byzantine failure caused by malicious clients. To overcome this issue, Byzantine-robust FL uses robust aggregation algorithm (AGR) to defend against adversaries. Current attacks often assume prior knowledge of the AGR and benign updates, which is impractical. In this paper, we propose a novel information-agnostic model poisoning attack named GGO to break down prevailing robust AGRs. With GGO, the attackers require no knowledge of the central server or the benign clients, and need only manipulate local updates of controlled clients. The GGO attack is designed based on the scheme of gap-based group obfuscating to manipulate the local updates in a fine-grained manner. We provide theoretical guarantee of the effectiveness of GGO towards representative state-of-the-art robust AGR method Zeno. Extensive experimental results further show that GGO attack can defeat five widely-used defence methods, Zeno, Krum, Trimmed Mean, Median and Centered Clipping. Specifically, on the CIFAR-10 dataset with 27 clients, our attack achieves an average drop in accuracy of 47.7%.

Keywords: Federated learning · Byzantine attacks · Byzantine-robust methods · Machine learning

1 Introduction

Federated Learning (FL) [9] has gained significant attention across various contexts, such as learning tasks on mobile and wireless devices [4, 26, 35], vehicle networks [19,21,25] and COVID-19 diagnosis [34]. This paradigm facilitates collaborative training of a global model, under the orchestration of a central server and numerous local clients. FL offers an advantage of preserving clients privacy by keeping their data locally. However, FL is known to be prone to Byzantine poisoning attacks [13]: malicious attackers can attempt to degrade the utility (e.g., model accuracy) of the global model by manipulating either local data or

local parameters. Current studies have developed some successful defence strategies against Byzantine attacks [2,8,10,20,28,32]. These defences focus on either filtering unreliable local updates or dropping some coordinates of the updates via the robust aggregation algorithm (AGR). For instance, score-based robust AGRs devise different score functions to select updates with high credibility, then directly discard others [2,32]. Element-based robust AGRs apply statistical approaches to exclude unreliable dimensions and retain the remaining [28,36].

Targeted at the robust AGRs, emerging attacks have evolved to break through such defences. For instance, 1) in [1], PCA attackers target Krum [2] and Trimmed Mean [36] by limiting the difference between benign and poisoned updates. They constantly add undetectable slight changes to the local updates and accumulatively break down the model. 2) IPM attackers in [30] target Krum and Median [36] by using a negative mean of benign local updates to derail the crowds direction. 3) in [7], LMPA attackers also target Krum and Trimmed Mean. They formulate the attack as optimization problems. For different problems, attackers solve them via adaptive methods, such as stochastic gradient descent and binary search. 4) in [23], Min-Max attackers target Krum, Trimmed Mean, Median and some other defences. The attacker limits the maximum distance of malicious gradient from any other gradients within the maximum distance between any two benign gradients.

However, the key issue of existing methods is that most of them *assume* the information, e.g., the AGR on the central server and benign updates of local devices, to be known in prior. For instance, in order to control the difference between benign updates and poisoned updates, PCA attackers have access to benign updates. Similarly, IPM attackers also have to know about benign updates for computing the negative average of them. Moreover, LMPA attackers need to know the defence method on the server and be adaptively different for different target defences. In this paper, we question the validity of this assumption, and point out that, in real-world FL scenarios, such information is either difficult to obtain, or may even be unavailable. For example, every local client keeps their data locally and privately, and an attacker can not have access to their dataset or updates. The AGRs on the server are also not visible to attackers. There are also a few existing attacks that claim to require no prior knowledge of benign clients or servers [23]. However, the effectiveness of these attacks can't be guaranteed and has very high variations in practice.

We take one step back and ask: *without prior information of benign clients and servers, how to conduct provably effective attacks against current Byzantine-robust FL?*

Motivated by the above question, we propose a novel information-agnostic model poisoning attack. In such a scenario, attackers have no knowledge of benign updates and the AGR algorithms on the server. Our attack, called **GGO**, follows the scheme of Gap-based Group Obfuscating process. A GGO attacker manipulates the adversarial updates by selecting key elements of the update and utilizing the gap before and after local updating, finds important dimen-

sions and conducts grouped poisoning. We provide theoretical guarantee of our GGO attack towards three existing Byzantine-tolerant defence, Zeno.

Besides, our experimental evaluations show that GGO can easily and effectively destruct five state-of-the-art defences, reaching an averagely 47.7% drop in accuracy with 27 clients on the CIFAR-10 dataset. Additionally, under the same realistic black-box conditions, GGO outperforms four prevailing attacks. We believe that our proposed attack scheme could provide inspirations for future defences designed against Byzantine attacks in FL settings.

Our contributions can be summarized below.

- **An information-agnostic attack method.** We propose GGO attack that works without any prior knowledge of benign clients and the AGRs of the server. It is more pragmatic than previous poisoning attacks, and easy to implement in real-world federated learning scenarios.
- **Theoretical guarantee of attack success.** We formally prove that our GGO attack is Byzantine-disturbing against state-of-the-art defence, Zeno.
- **Experimental evaluations.** We evaluate the performance of GGO attack on five different defence settings and compare it with four existing attacks. Experimental results show significant performances of our attack.

2 Background and Related Work

2.1 Federated Learning

Federated learning is a distributed training paradigm where n clients collaboratively train a model under the orchestration of a central server, while keeping the training data decentralized [9]. FL aims to find the optimal parameter $\mathbf{w}^* \in \mathbb{R}^d$ by solving the following empirical risk minimization problem:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} F(\mathbf{w}) = \arg\min_{\mathbf{w}} \frac{1}{n} \sum_{i=1}^{n} f_i(\mathbf{w}). \tag{1}$$

The function $f_i(\mathbf{w}) = \mathbb{E}_{\mathbf{X}_i \sim \mathcal{D}_i}[\mathcal{L}_i(\mathbf{w}; \mathbf{X}_i)]$ denotes the local objective function of the i-th client. $\mathcal{L}_i$ denotes the corresponding loss function of heterogeneous local dataset $\mathcal{D}_i, i \in [n]$. In each communication round t, the server sends global parameter $\mathbf{w}^t$ to selected clients $\mathcal{S} \subseteq [n]$. Then, every client i conducts local updating through parallelly running ξ_i iterations of SGD with mini-batch $\mathbf{X}_i$:

$$\mathbf{w}_i^{t,l+1} = \mathbf{w}_i^{t,l} - \eta \nabla \mathcal{L}_i(\mathbf{w}_i^{t,l}; \mathbf{X}_i), l \in [\xi_i] \tag{2}$$

where η is the learning rate. Then, selected clients upload the local model updates to the server. Finally, the server updates the global parameter:

$$\mathbf{w}^{t+1} = \mathbf{w}^t + \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} (\mathbf{w}_i^{t,\xi_i} - \mathbf{w}^t). \tag{3}$$

2.2 Existing Defence Methods

Recently, many Byzantine-robust defences have been developed, such as [3,5,14, 28,29,31,36]. The key of these defences is to use Byzantine-robust aggregation algorithms (AGRs) to filter adversarial updates.

Score-based defences. This kind of defences usually devise score functions to characterize the credibility of local updates from different clients. The server updates the global model with those of high credibility. For instance, Zeno proposed by [32] leverages a stochastic descendant score to accomplish the suspicion-based aggregation. Besides, [2] proposes a defence method named Krum. For each local client, Krum computes its score elementwisely for all updates through a distance score function, then outputs the one with the lowest score. In addition, iterative Centered Clipping proposed by [10] uses a minimization function to control the updating scale on the server. There are also other different score functions proposed by [16,24,33]. In general, score-based defences use score functions to find outliers that are far away from others.

Element-based defences. These defences select updates by measuring the distance between update elements. For instance, Median and Trimmed Mean are widely used and discussed [5,10,28]. Median outputs the median value for every coordinate of local update vectors. Trimmed Mean calculates the mean value after trimming m-largest and m-smallest elements of every coordinate [36]. They both require that the number of honest clients is larger than that of Byzantine clients. Generally, element-based defences use statistical approaches to discard unreliable elements.

Hybrid defences. These defences combine the above two kinds of defences. For instance, [8] proposes Bulyan. It uses the score-based defence Krum to preselect some local updates, then use the element-based defence Median to calculate the final result. An obvious shortcoming of Bulyan is that it has a strict requirement for the number of Byzantine clients, i.e., $n \geq 4f + 3$. It means when there are 10 local clients in total, Bulyan can only tolerate 1 of them as a Byzantine client. Therefore, this method can hardly be applied in real world applications and isn't included in our experimental part.

2.3 Existing Attacks against Byzantine-robust FL

Against current Byzantine-robust FL, emerging attacks have been proposed. They usually have to access some prior knowledge of honest clients or the server, e.g., benign updates or the defence method. Here summarizes the assumptions on adversary's knowledge of existing representative attacks against Byzantine-robust defences: LMPA (Local Model Poisoning Attack) [7], PCA (Preventing Convergence Attack) [1], IPM (Inner Product Manipulation) [30], Min-Max updates-only (Min-Max-U) [23]. Specifically, LMPA attackers formulate their attacks as optimization problems with full or partial knowledge. Their optimization problems vary for different defence methods, so the attackers should know the defence method on the server. This assumption is impractical in privacy-preserving FL cases. PCA attackers destruct the convergence of the global model

by consistently applying small changes to model parameters. However, it assumes that local data are all normally distributed, which is not suitable for FL settings. Moreover, without information of the number of total clients, this attack would fail. IPM attackers compute the negative average of benign updates as the poisoned result. Additionally, Min-Max (updates-only) computes the poisoned result such that its maximum distance from any other local update is upper bounded by the maximum distance between any two benign updates. Note that the prior knowledge of these adversaries is either difficult to obtain, or may even be unavailable in practice. For example, every local client keeps their data locally and privately, and an attacker can not have access to their dataset or updates. The AGRs on the server are also not visible to attackers.

We are also aware of a few existing attacks that claim to require no prior knowledge of benign clients or servers, such as Min-Max agnostic (Min-Max-A) [23]. However, the effectiveness of Min-Max-A highly relies on the choice of perturbation vectors.

3 Gap-based Group Obfuscating Attack

Attack and defence settings. In our setting, GGO attackers only have access to local updates, without knowledge including benign updates, the defence on the server, benign data distributions and the number of total clients. In real federated systems, there might be multiple attackers with the ability to collaborate and communicate [11]. Previous research by [9] proposed the collusion assumption of Byzantine attackers, which was followed by latter studies [7, 30]. We join the consensus with them and assume that GGO attackers can collude with each other.

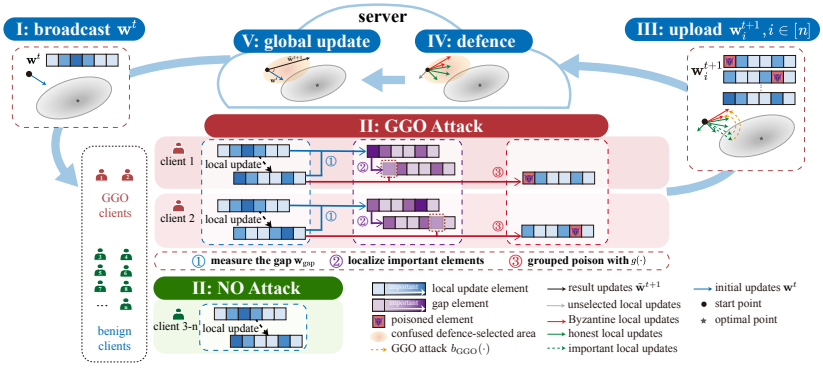


Fig. 1. The pipeline of GGO attack in the t-th communication round.

GGO attack is a three-step obfuscation scheme, and its whole pipeline under FL settings is shown in Fig. 1. In the t-th iteration, a GGO attacker first measures

the gap $\mathbf{w}_{\text{gap}}$ between the local parameter $\mathbf{w}^{t+1}$ and $\mathbf{w}^t$ elementwisely. Then, with the intuition that local update elements which change more prominently outweigh others, the attacker localizes important elements with an importance recorder $\mathbb{1}(\mathbf{w}^{t+1}, \mathbf{w}^t)$ through certain approaches, which would be introduced later. Here, $\mathbb{1}(\mathbf{w}^{t+1}, \mathbf{w}^t)$ denotes a binary vector that elementwisely records the importance of the gap between $\mathbf{w}^{t+1}$ and $\mathbf{w}^t$. $(\mathbb{1}(\mathbf{w}^{t+1}, \mathbf{w}^t))_j = 1$ denotes that the j-th dimension of the parameter is important, otherwise the vector element equals 0. Finally, the attacker uses a function $g(\cdot)$ to poison these important elements through group collusion. In this case, a group of GGO attackers can share their local updates, then collaboratively disturb the global model. Generally, in the t-th iteration, with local update result $\mathbf{w}^{t+1} \in \mathcal{G}$, a GGO attacker completes its attack via

$$\begin{aligned} b_{\text{GGO}}(\mathbf{w}^{t+1}) = \mathbb{1}(\mathbf{w}^{t+1}, \mathbf{w}^t) \odot g(\mathbf{w}^{t+1}) \\ + (1 - \mathbb{1}(\mathbf{w}^{t+1}, \mathbf{w}^t)) \odot \mathbf{w}^{t+1}. \end{aligned} \tag{4}$$

Here, the symbol $\odot$ denotes the Hadamard product, and $g(\cdot)$ denotes a group function for attack. The detailed three steps of GGO are as follows.

Step 1 (measure the gap) We straightforwardly use Equation (5) to obtain the gap before and after local updating. To facilitate reading, we use $\mathbf{w}^t$ to denote the local update vector of the i-th client in the t-th iteration, and use $\mathbf{w}_{\text{gap}}$ to denote the update gap of attacker i in the t-th iteration.

$$\mathbf{w}_{\text{gap}} = \mathbf{w}^{t+1} - \mathbf{w}^t. \tag{5}$$

We have $\mathbb{1}(\mathbf{w}^{t+1}, \mathbf{w}^t) = \mathbb{1}(\mathbf{w}_{\text{gap}})$ and we use $\mathbb{1}(\mathbf{w}_{\text{gap}})$ as the gap of the i-th client hereafter for easier reading.

Step 2 (localize important elements) Every attacker uses the gap $\mathbf{w}_{\text{gap}}$ to localize elements that have more significant influence on the local update. We let $(\mathbb{1}(\mathbf{w}_{\text{gap}}))_j = 1$ denote that the corresponding j-th element of $\mathbf{w}^{t+1}$ is important and would be poisoned by the attacker then. Here, we propose three approaches for localizing elements. They measure the importance of update elements from different perspectives.

- Approach 1: threshold-based.

$$(\mathbb{1}(\mathbf{w}_{\text{gap}}))_j = \begin{cases} 1, \text{ if } |\frac{(w_{\text{gap}})_j}{w_j^t}| > \gamma, \ j \in [d] \\ 0, \text{ otherwise,} \end{cases} \tag{6}$$

  where $(w_{\text{gap}})_j$ and w_j^t denote the j-th dimension of $\mathbf{w}_{\text{gap}}$ and $\mathbf{w}^t$ respectively, γ denotes the threshold and d denotes the dimension of an update vector. This approach selects elements whose variations have reached the threshold.
- Approach 2: sign-based.

$$(\mathbb{1}(\mathbf{w}_{\text{gap}}))_j = \begin{cases} 1, \text{ if } \text{sign}(w_j^t) = -\text{sign}(w_j^{t+1}), \ j \in [d] \\ 0, \text{ otherwise,} \end{cases} \tag{7}$$

where sign$(\cdot)$ is a sign function. This approach simply selects elements whose direction have changed.

– Approach 3: deviation-based.

$$(\mathbb{1}(\mathbf{w}_{\text{gap}}))_j = \begin{cases} 1, \text{ if } |(w_{\text{gap}})_j| > \sigma, \ j \in [d] \\ 0, \text{ otherwise,} \end{cases} \tag{8}$$

where σ is the standard deviation of $w_j^t, \ j \in [d]$. This approach selects elements which hold changes over the standard deviation.

Above three approaches are applied in different scenarios. The threshold-based approach works well for most cases. The sign-based approach works in the update process where directions change choppily. The deviation-based approach relates to standard deviation of updates. When the distributions of local datasets are heavily non-IID, it might not be applicable. Generally, we recommend using the threshold-based approach.

Step 3 (grouped poison) There have been several classical ways to attack the model, e.g., sign-flipping attack [6] and Gaussian attack [7]. However, these attacks usually change the parameter elements markedly, e.g., sign-flipping attacker i simply sends $-10\mathbf{w}^t$ rather than $\mathbf{w}^t$ to the server. These kinds of attacks evade elements markedly and make them easily filtered out by defences on the server. In order to make the poisoned elements less noticeable, with the idea of peer collusion, we use a group function $g(\cdot)$ to poison selected important elements via Equation (9).

$$\tilde{\mathbf{w}}^{t+1} = g(\mathbf{w}^{t+1}) = -\frac{\delta}{f}\sum_{i=1}^{f}(\mathbf{w}_i^{t+1} \odot \mathbb{1}(\mathbf{w}_{\text{gap}})), \tag{9}$$

where δ is the perturbation weight and indicates the degree of poisoning.

4 Theoretical Guarantee of GGO attack

4.1 Definition of Byzantine Disturbance

Inspired by Xie's work [30] about Byzantine tolerance, the definition of Byzantine disturbance that measures successful Byzantine attacking with the following intuition: In FL, the central server usually uses a robust aggregation defence method $d(\cdot)$ to filter n received local parameters $\mathcal{G} = \{\mathbf{w}_1, \cdots, \mathbf{w}_n\}$ before global model updating [17, 18, 27]. When there exist no attacks, the aggregated result is supposed to guarantee the descent of the loss. On the contrary, an effective Byzantine attack $b(\cdot)$ usually leads the average parameter $\bar{\mathbf{w}}$ to a poisoned $\tilde{\mathbf{w}}$, and changes the updating direction on the server. Therefore, the inner product of aggregated parameters with and without attacks can indicate the effectiveness of Byzantine attacks [30]. Based on this intuition, we devise the following definition:

Definition 1. *(Byzantine Disturbance). Without loss of generality, we suppose that in a certain communication round, the server originally receives n local updates $\mathcal{G} = \{\mathbf{w}_1, \cdots, \mathbf{w}_n\}$, and $\mathbb{E}[\mathbf{w}_i] = \bar{\mathbf{w}}, \forall i \in [n]$. Assume that after Byzantine attack $b(\cdot)$, the server leverages the defence $d(\cdot)$ to filter Byzantine updates. Let $\tilde{\mathbf{w}} = d(b(\mathcal{G}))$ denote the defence result of received updates after attacking. Thus, the attack $b(\cdot)$ is Byzantine-disturbing towards $d(\cdot)$ if*

$$\langle \bar{\mathbf{w}},\ \tilde{\mathbf{w}} \rangle \leq 0. \tag{10}$$

With the above definition, we then provide theoretical guarantee of our GGO attack towards the prevailing defence.

4.2 Targeted Defence and Assumptions

Definition of targeted defence Zeno [32] is listed as follows.

Zeno. This method leverages the stochastic descendant score to accomplish the suspicion-based aggregation. For any local update vector $\mathbf{w}_i, i \in [n]$ and current parameter $\mathbf{w}$, Zeno selects local updates with the first $n-b$ ($n \geq b \geq f$) highest stochastic descendant scores via the following score function:

$$\begin{aligned} \text{Score}(\mathbf{w}_i, \mathbf{w}) &= F(\mathbf{w}) - F(\mathbf{w}_i) - \rho \|\mathbf{w} - \mathbf{w}_i\|^2, \\ d_{\text{Zeno}}(\mathbf{w}_1, \cdots, \mathbf{w}_n) &= d_{\text{Zeno}}(\mathbf{v}_1, \cdots, \mathbf{v}_{n-b}) = \frac{1}{n-b} \sum_{i=1}^{n-b} \mathbf{v}_i, \end{aligned} \tag{11}$$

where $\rho > 0$ is a hyper-parameter, and $\mathbf{v}_1, \cdots, \mathbf{v}_{n-b}$ denote parameters with the first $n-b$ highest scores. Zeno is claimed to be able to tolerate arbitrary number of Byzantine clients.

Before proving the Byzantine disturbance of GGO attack, we provide some assumptions.

Assumption 1 *We assume that $F(\mathbf{w})$ has L-smoothness and μ-lower-bounded Taylors approximation (also called μ-weak convexity): $\langle \nabla F(\mathbf{w}_1), \mathbf{w}_1 - \mathbf{w}_2 \rangle + \frac{\mu}{2}\|\mathbf{w}_1 - \mathbf{w}_2\|^2 \leq F(\mathbf{w}_1) - F(\mathbf{w}_2) \leq \langle \nabla F(\mathbf{w}_1), \mathbf{w}_1 - \mathbf{w}_2 \rangle + \frac{L}{2}\|\mathbf{w}_1 - \mathbf{w}_2\|^2$.*

Assumption 2 *We assume that in any iteration, honest updates received by the server have upper-bounded variance: $\mathbb{E}\|\mathbf{w}_i - \bar{\mathbf{w}}\| \leq V, \mathbf{w}_i \in \mathcal{H}$, where $\bar{\mathbf{w}}$ follows Definition 1.*

Assumption 3 *We assume that maximum number of local updates have largest variance, i.e., in a certain communication round, every dimension of the initial n local updates holds variance V.*

Generally, Assumption 1 covers the cases of non-convexity, non-strong convexity and strong convexity by taking $\mu < 0, \mu = 0$ and $\mu > 0$ respectively. Assumption 2 bounds the variance of all local updates and Assumption 3 assumes the worst case of benign updates. With these assumptions held, we prove the Byzantine disturbance of our GGO attack.

4.3 Theoretical Guarantee for Attacking Zeno

The following theorem shows that GGO attack is Byzantine-disturbing towards Zeno d_{Zeno}. In detail, GGO could successfully defeat this defence by element-wisely controlling local updates in a degree that highly relates to the variance V and the hyper-parameter ρ of Zeno.

Theorem 1. *We suppose that in a certain communication round, the server originally receives n local updates $\mathcal{G} = \{\mathbf{w}_1, \cdots, \mathbf{w}_n\}$, where $\mathbb{E}[\mathbf{w}_i] = \bar{\mathbf{w}}, \forall i \in [n]$. We assume that there are $n = 2f$ clients, where f are* GGO *attackers. Let $\tilde{\mathbf{w}} = d_{\text{Zeno}}(b_{\text{GGO}}(\mathcal{G}))$ denote the defence result of received updates after* GGO *attack. When attacked by* GGO *with perturbation weight δ, for every dimension of $\bar{\mathbf{w}}$, if $\frac{Vf(L-2\rho)}{(f+\delta)(L+2\rho)} \leq \|\bar{w}_j\| \leq \frac{Vf}{f+\delta}, j \in [d]$, there exists Byzantine disturbance of* GGO *such that $\langle \bar{\mathbf{w}}, \ \tilde{\mathbf{w}} \rangle \leq 0$.*

Proof. Without loss of generality, we suppose that f Byzantine updates and $(n-f)$ benign updates are all scalars. To ease interpretation, we assume that in the original set $\mathcal{G}$, $w_i = \bar{w} + V, w_j = \bar{w} - V, i = 1, \cdots, f,$ and $j = f+1, \cdots, n$. We first consider the situation where GGO attackers choose the first f of them to poison, and we denote $\mathcal{B} = \{\tilde{w}_1, \cdots, \tilde{w}_f\}$, $\mathcal{H} = \{w_{f+1}, \cdots, w_n\}$, where $\mathcal{B} \cup \mathcal{H} = \text{GGO}(\mathcal{G})$. As mentioned in Section 4.1, for Byzantine and benign updates respectively, Zeno calculates their score via the stochastic descendant score function:

$$Score(\tilde{w}_i, \bar{w}) = F(\bar{w}) - F(\tilde{w}_i) - \rho\|\bar{w} - \tilde{w}_i\|^2, \tag{12}$$

$$Score(w_i, \bar{w}) = F(\bar{w}) - F(w_i) - \rho\|\bar{w} - w_i\|^2. \tag{13}$$

We then prove that under certain conditions, the score of Byzantine updates is greater than benign updates, thus Zeno would choose the wrong ones for global model updating. Subtract Equation (12) from Equation (13), we have

$$\begin{aligned} &Score(w_i, \bar{w}) - Score(\tilde{w}_i, \bar{w}) \\ &= F(\tilde{w}_i) - F(w_i) + \rho\|\bar{w} - \tilde{w}_i\|^2 - \rho\|\bar{w} - w_i\|^2. \end{aligned} \tag{14}$$

With Equation (9), We have $\tilde{w}_i = -\frac{\delta}{f}\bar{w}, w_i = \bar{w} - V$. Using Assumption 1, then

$$\begin{aligned} Score(w_i, \bar{w}) - Score(\tilde{w}_i, \bar{w}) \leq \langle \nabla F(w_i), -\frac{\delta}{f}\bar{w} - \bar{w} + V \rangle \\ + \frac{L}{2}\| - \frac{\delta}{f}\bar{w} - \bar{w} + V\|^2 + \rho\|\bar{w} + \frac{\delta}{f}\bar{w}\|^2 - \rho\|V\|^2. \end{aligned}$$

When $\frac{Vf(L-2\rho)}{(f+\delta)(L+2\rho)} \leq \bar{w}_j \leq \frac{Vf}{f+\delta}, j \in [d]$ and with assumptions above, we can obtain that

$$\begin{aligned} &Score(w_i, \bar{w}) - Score(\tilde{w}_i, \bar{w}) \\ &\leq \frac{L}{2}\| - \frac{\delta}{f}\bar{w} - \bar{w} + V\|^2 + \rho\|\bar{w} + \frac{\delta}{f}\bar{w}\|^2 - \rho\|V\|^2. \end{aligned}$$

With $\frac{Vf(L-2\rho)}{(f+\delta)(L+2\rho)} \leq \bar{w}_j \leq \frac{Vf}{f+\delta}, j \in [d]$, we obtain that

$$(\frac{L}{2}+\rho)(f+\delta)^2\bar{w}^2 - L(\delta+f)fV\bar{w} + (\frac{L}{2}-\rho)f^2V^2 \leq 0.$$

So Equation (14) satisfies that

$$Score(w_i, \bar{w}) - Score(\tilde{w}_i, \bar{w}) \leq 0.$$

Since $Score(w_i, \bar{w}) \leq Score(\tilde{w}_i, \bar{w})$, Zeno would choose the poisoned updates, which means $d_{\text{Zeno}}(b_{\text{GGO}}(\mathcal{G})) = -\frac{\delta}{f}\bar{w}$ and $\langle \bar{w},\ d_{\text{Zeno}}(b_{\text{GGO}}(\mathcal{G}))\rangle \leq 0$. Then we consider the situation where GGO attackers choose the second f of them to poison, and we denote $\mathcal{B} = \{\tilde{w}_1, \cdots, \tilde{w}_f\}$, $\mathcal{H} = \{w_{f+1}, \cdots, w_n\}$, where $\mathcal{B} \cup \mathcal{H} = \text{GGO}(\mathcal{G})$. As mentioned before, for Byzantine and benign updates respectively, Zeno calculates their score via Equation (12) and (13). Subtract Equation (12) from Equation (13), we obtain Equation (14). With Equation (9), We have $\tilde{w}_i = -\frac{\delta}{f}\bar{w}, w_i = \bar{w} + V$. Using Assumption 1, then

$$\begin{aligned}
&Score(w_i, \bar{w}) - Score(\tilde{w}_i, \bar{w}) \leq \langle \nabla F(w_i), \tilde{w}_i - w_i \rangle \\
&+ \frac{L}{2}\|\tilde{w}_i - w_i\|^2 + \rho\|\bar{w} - \tilde{w}_i\|^2 - \rho\|\bar{w} - w_i\|^2 \\
&\leq \langle \nabla F(w_i), -\frac{\delta}{f}\bar{w} - \bar{w} - V \rangle + \frac{L}{2}\| - \frac{\delta}{f}\bar{w} - \bar{w} - V\|^2 \\
&+ \rho\|\bar{w} + \frac{\delta}{f}\bar{w}\|^2 - \rho\|V\|^2.
\end{aligned}$$

When $-\frac{Vf}{f+\delta} \leq \bar{w}_j \leq \frac{Vf(2\rho-L)}{(f+\delta)(L+2\rho)}$, we have $\langle \nabla F(w_i), -\frac{(\delta+f)\bar{w}+Vf}{f} \rangle \leq 0$, then

$$\begin{aligned}
&Score(w_i, \bar{w}) - Score(\tilde{w}_i, \bar{w}) \leq \frac{L}{2}\| - \frac{\delta}{f}\bar{w} - \bar{w} - V\|^2 \\
&+ \rho\|\bar{w} + \frac{\delta}{f}\bar{w}\|^2 - \rho\|V\|^2.
\end{aligned}$$

And we have

$$(\frac{L}{2}+\rho)(f+\delta)^2\bar{w}^2 - L(\delta+f)fV\bar{w} + (\frac{L}{2}-\rho)f^2V^2 \leq 0.$$

So Equation (14) satisfies that

$$Score(w_i, \bar{w}) - Score(\tilde{w}_i, \bar{w}) \leq 0.$$

Since $Score(w_i, \bar{w}) \leq Score(\tilde{w}_i, \bar{w})$, Zeno would choose the poisoned updates, which means $d_{\text{Zeno}}(b_{\text{GGO}}(\mathcal{G})) = -\frac{\delta}{f}\bar{w}$ and $\langle \bar{w},\ d_{\text{Zeno}}(b_{\text{GGO}}(\mathcal{G}))\rangle \leq 0$. Combining the results above, we have proved Theorem 1.

Remark 1. Theorem 1 brings an intuition for Byzantine attackers that via elementwisely controlling the values of update results, they could manipulate the aggregated updating direction on the server. This process would be more likely to occur when the variance is large. In addition, we can observe an obvious drawback of Zeno. To properly decide the hyper-parameter b (Definition 3 in [32]), Zeno requires knowing the number of Byzantine attackers in advance. It is unrealistic in real scenarios, where the server has no access to local clients.

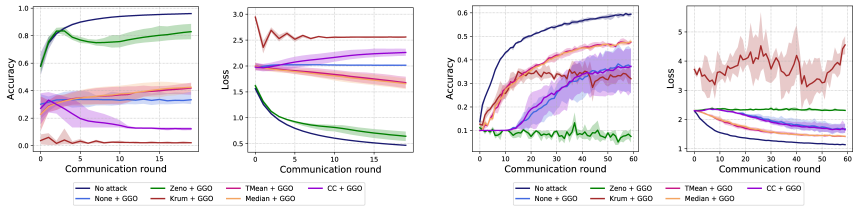
5 Experimental Evaluations

In this section, we evaluate the performance of GGO on five state-of-the-art defence settings, then compare it with four prevailing Byzantine attacks. We also have done some extensive experiments , including the impact of the number of clients on GGO and the impact of data complexity on Zeno.

5.1 Experimental Settings

Datasets We consider the classification task on two datasets, a generated dataset and the CIFAR-10 dataset [12]. The generated dataset consists of 8 classes of 3D vectors. There are 40K training examples and 10K testing examples in it. The CIFAR-10 dataset consists of 10 classes of 32×32 images with 50K training examples and 10K testing examples.

Defences We evaluate the performance of proposed GGO attack on six different defence settings. The involved defences are Zeno, Krum, Trimmed Mean (referred as TMean), Median and Centered Clipping (referred as CC). Additionally, we consider the None defence setting where there is no defence on the server. We use no defence without attacks (referred as No attack) as baselines. Note that for TMean, we follow the common setting that the number of trimmed dimensions $m = f$ in all experiments. For Zeno, after pretuning, we set the hyper-parameter $\rho = 0.01$. For Centered Clipping (referred as CC), we follow the experimental setting in Section 7.3 of [10], pretune the hyper-parameters τ and L via a grid search scheme, where $\tau \in \{0.1, 1, 10, 100\}, L \in \{1, 3, 5\}$. We choose $\tau = 1, L = 5$ for the generated dataset, $\tau = 10, L = 1$ for CIFAR-10.



(a) performance of GGO in the generated dataset

(b) performance of GGO in the CIFAR-10 dataset

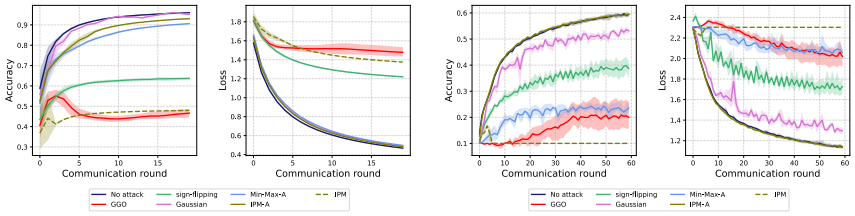
Fig. 2. Evaluate the performance of GGO on five defence settings. The test accuracy (left) and loss (right) of five defence settings on the (a) generated and (b) CIFAR-10 datasets.

Models For the generated dataset, we use a perceptron model with a fully connected layer. For the CIFAR-10 dataset, we use a CNN model with 2 convolutional layers and 3 fully connected layers. We have our experiments in label distribution skew non-IID settings, where the label distributions vary across parties

[15]. We have evaluated the performance of GGO attack with $\delta \in \{0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75, 2\}$, $\gamma \in \{0.25, 0.5, 0.75\}$ and we've found that $\delta = 1, \gamma = 0.5$ is a universally good choice. So in all experiments, we use the threshold-based approach for GGO attack and set γ and δ to be 0.5 and 1 respectively. The learning rate η and the batch size are 0.01 and 64 respectively, and the max number of communication rounds T for the generated and the CIFAR-10 dataset are 20 and 60 respectively. We set the number of local updates $\xi_i = 5, i \in [n]$. In every communication round, the selected set is fixed as $\mathcal{S} = [n]$. All experiments are repeated for 5 times and the average results are shown.

5.2 Evaluating Effectiveness of GGO on Five Defences

As shown in Fig. 2, GGO is Byzantine-disturbing towards 5 involved defences. In this experiment, the number of total clients and Byzantine clients are 27 and 12 respectively. For the generated dataset, GGO attack shows good performance on Krum, TMean, Median and CC, with decrease of accuracy from 96.08% to less than 43%, while it shows less remarkable performance on Zeno. For the CIFAR-10 dataset, the results just shows the opposite. GGO attack perfectly defeats Zeno, while it shows less remarkable impact on Krum, TMean, Median and CC, with decreases of accuracy from 59.42% to around 37% $\sim$ 48%. The performance of GGO attack on None defence setting is similar to TMean and Median for the generated dataset, CC for the CIFAR-10 dataset, which indicates that these defences have no effect against our GGO attack. It can be noticed in Fig. 2(b) that Krum shows the most significant variance among all the defence methods. This is owing to the number of chosen local updates. As mentioned in Section 2.2, Krum chooses only one result with the lowest distance score for global updating. In addition, note that the performance of GGO towards Zeno on two datasets are quite different. It results from different data complexity of these two datasets, since CIFAR-10 data is much more complex than the generated data. We include more advanced experiments about this phenomenon in Section 5.5.



(a) performance of five attacks on the generated dataset

(b) performance of five attacks on the CIFAR-10 dataset

Fig. 3. Compare GGO with four Byzantine attacks. The test accuracy (left) and loss (right) of five attacks on the (a) generated and (b) CIFAR-10 datasets.

5.3 Comparing GGO with Four Attacks

As shown in Fig. 3, we compare our GGO attack with four existing Byzantine attacks, including three black-box attacks, Min-Max-A, sign-flipping and Gaussian and one prevailing attack IPM with its agnostic version IPM-A. Note that the other three attacks mentioned in Section 2.3 are not involved, because they would fail without white-box knowledge.

For Min-Max-A attack, we choose the inverse standard deviation [23] with generally good experimental performances as the perturbation vector $\mathbf{w}^p$. Besides, we set two hyper-parameters $\gamma_{\text{init}} = 1, \tau = 0.01$. For sign-flipping attacker i, it sends $-10\mathbf{w}_i$ rather than $\mathbf{w}_i$ to the server. For the Gaussian attacker j, it adds a random noise from a standard normal distribution $N(0, 1)$ to $\mathbf{w}_j$ before sending it to the server. For IPM, we follow the definition in [30]. Besides, we randomize the result of the negative average to involve an agnostic version of IPM, which is referred as IPM-A. In this experiment, the number of total clients and Byzantine clients are 27 and 12 respectively. We evaluate the performances of attacks on two defences, Zeno and CC, and the average results are shown. To the best of our knowledge, we are the first to consider Byzantine attacks on these two defences. Fig. 3 shows that GGO attack outperforms four attacks for both datasets. Specifically, sign-flipping shows a moderate performance among these attacks. Min-Max-A attack could compromise the model on CIFAR-10, but it fails on the generated dataset. The Gaussian attack could not influence the model on both datasets. As for IPM, note that although the original method shows good performances, its agnostic version performs dramatically worse than GGO.

5.4 Impact of The Number of Clients

Table 1. Accuracy of GGO attack. The number of total clients $n \in \{15, 27, 39\}$.

Defence	Generated Dataset			CIFAR-10		
	$n = 15$	$n = 27$	$n = 39$	$n = 15$	$n = 27$	$n = 39$
No attack	95.8	96.1	97.9	62.0	60.0	57.3
None	43.4(↓ 54.7%)	34.7(↓ 63.9%)	25.1(↓ 74.4%)	38.8(↓ 37.4%)	40.9(↓ 31.9%)	32.8(↓ 42.8%)
Zeno	82.5(↓ 13.9%)	82.6(↓ 14.6%)	91.9(↓ 6.1%)	12.0(↓ 80.6%)	10.7(↓ 82.2%)	4.6(↓ 91.9%)
Krum	33.4(↓ 65.1%)	2.9(↓ 97.0%)	8.7(↓ 91.1%)	36.3(↓ 41.5%)	29.7(↓ 50.6%)	18.5(↓ 67.7%)
TMean	57.3(↓ 40.2%)	39.1(↓ 59.3%)	50.6(↓ 48.3%)	51.9(↓ 16.3%)	46.9(↓ 21.9%)	47.7(↓ 16.7%)
Median	56.1(↓ 41.4%)	39.4(↓ 59.0%)	49.0(↓ 49.9%)	50.6(↓ 18.4%)	47.8(↓ 20.3%)	47.9(↓ 16.4%)
CC	58.5(↓ 38.9%)	12.3(↓ 87.2%)	31.8(↓ 67.5%)	34.2(↓ 64.3%)	35.3(↓ 63.3%)	32.5(↓ 66.8%)

Table 1 shows the significant performances of GGO on two datasets with the number of total clients $n \in \{15, 27, 39\}$. To satisfy the constrains of all defence methods, we set that $n = 2f+3$. In these two tables, (↓) indicates the percentage decrease in accuracy. For the generated dataset,GGO attack successfully defeats Krum, TMean, Median and CC, with reduction in accuracy ranging from 40% to 97%. For the CIFAR-10 dataset, GGO attack can defeat Zeno completely, with up to 91.9% drop in accuracy when $n = 39$. It shows stable impacts for other

four defence methods. Note that the different performances of GGO against Zeno on two datasets result from data complexity as mentioned in Section 5.5. For both two datasets, when the number of clients increases, the performances of GGO only show acceptable slight fluctuation. Thus, it can be concluded that the number of clients doesn't have a significant impact on the performance of GGO.

5.5 Impact of Data Complexity on Zeno

Fig. 4 shows that Byzantine tolerance of Zeno is heavily influenced by data complexity. In this experiment, we set the generated data to be 3D, 4D and 5D vectors and the number of labels in $\{8, 16, 32\}$ respectively. The number of total clients and Byzantine clients are 27 and 12. As shown in the figure, when the complexity of data, i.e., the number of labels and features increases, there is a sharp fall of the test accuracy, and a marked rise in the final loss. Therefore, Zeno is deeply influenced by the data complexity. This conclusion marches results of aforementioned experiments. Beyond that, the results also join the consensus with the previous work [22], which pointed out that the success rate of an attack could be strongly related to the dataset size.

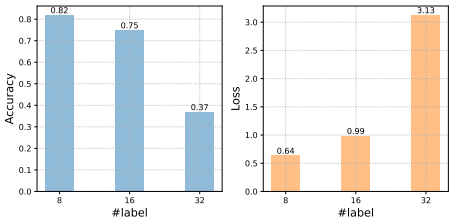


Fig. 4. Evaluate the impact of dataset scale on Zeno. The test accuracy (left), loss (right) on the generated dataset. The number of labels in $\{8, 16, 32\}$ corresponds to the number of features in $\{3, 4, 5\}$.

Acknowledgments. This work was supported by National Natural Science Foundation of China (Grant Numbers 61906040, 61972085, 62276063, 6509009710), the National Key Research and Development Program of China (Grant Number 2022YFF0712400), the Natural Science Foundation of Jiangsu Province (Grant Number BK20221457, BK20230083), and the Fundamental Research Funds for the Central Universities (Grant Number 2242021R41177).

References

1. Baruch, G., Baruch, M., Goldberg, Y.: A little is enough: Circumventing defenses for distributed learning. Advances in Neural Information Processing Systems **32**, 8635–8645 (2019)

2. Blanchard, P., El Mhamdi, E.M., Guerraoui, R., Stainer, J.: Machine learning with adversaries: Byzantine tolerant gradient descent. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. pp. 118–128 (2017)
3. Chen, L., Wang, H., Charles, Z., Papailiopoulos, D.: Draco: Byzantine-resilient distributed training via redundant gradients. In: International Conference on Machine Learning. pp. 903–912 (2018)
4. Chen, M., Poor, H.V., Saad, W., Cui, S.: Wireless communications for collaborative federated learning. IEEE Communications Magazine **58**(12), 48–54 (2020)
5. Chen, Y., Su, L., Xu, J.: Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. Proceedings of the ACM on Measurement and Analysis of Computing Systems **1**(2), 1–25 (2017)
6. Damaskinos, G., Guerraoui, R., Patra, R., Taziki, M., et al.: Asynchronous byzantine machine learning (the case of sgd). In: International Conference on Machine Learning. pp. 1145–1154 (2018)
7. Fang, M., Cao, X., Jia, J., Gong, N.: Local model poisoning attacks to byzantine-robust federated learning. In: 29th {USENIX} Security Symposium ({USENIX} Security 20). pp. 1605–1622 (2020)
8. Guerraoui, R., Rouault, S., et al.: The hidden vulnerability of distributed learning in byzantium. In: International Conference on Machine Learning. pp. 3521–3530 (2018)
9. Kairouz, P., McMahan, H.B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A.N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al.: Advances and open problems in federated learning. Foundations and Trends® in Machine Learning **14**(1–2), 1–210 (2021)
10. Karimireddy, S.P., He, L., Jaggi, M.: Learning from history for byzantine robust optimization. In: Proceedings of the 38th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 139, pp. 5311–5319 (2021), `https://proceedings.mlr.press/v139/karimireddy21a.html`
11. Karimireddy, S.P., Kale, S., Mohri, M., Reddi, S.J., Stich, S.U., Suresh, A.T.: Scaffold: Stochastic controlled averaging for on-device federated learning. CoRR **abs/1910.06378** (2019), `http://arxiv.org/abs/1910.06378`
12. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images. Master's thesis, University of Tront (2009)
13. LAMPORT, L., SHOSTAK, R., PEASE, M.: The byzantine generals problem. ACM Transactions on Programming Languages and Systems **4**(3), 382–401 (1982)
14. Li, H., Sun, X., Zheng, Z.: Learning to attack federated learning: A model-based reinforcement learning attack framework. In: Advances in Neural Information Processing Systems (2022)
15. Li, Q., Diao, Y., Chen, Q., He, B.: Federated learning on non-iid data silos: An experimental study. arXiv preprint arXiv:2102.02079 (2021)
16. Li, S., Cheng, Y., Liu, Y., Wang, W., Chen, T.: Abnormal client behavior detection in federated learning. arXiv preprint arXiv:1910.09933 (2019)
17. Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V.: Federated optimization in heterogeneous networks. Proceedings of Machine Learning and Systems **2**, 429–450 (2020)
18. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial intelligence and statistics. pp. 1273–1282 (2017)
19. Posner, J., Tseng, L., Aloqaily, M., Jararweh, Y.: Federated learning in vehicular networks: opportunities and solutions. IEEE Network **35**(2), 152–159 (2021)

20. Qu, L., Zhou, Y., Liang, P.P., Xia, Y., Wang, F., Adeli, E., Fei-Fei, L., Rubin, D.: Rethinking architecture design for tackling data heterogeneity in federated learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10061–10071 (2022)
21. Saputra, Y.M., Hoang, D.T., Nguyen, D.N., Dutkiewicz, E., Mueck, M.D., Srikanteswara, S.: Energy demand prediction with federated learning for electric vehicle networks. In: 2019 IEEE Global Communications Conference (GLOBECOM). pp. 1–6 (2019)
22. Schwarzschild, A., Goldblum, M., Gupta, A., Dickerson, J.P., Goldstein, T.: Just how toxic is data poisoning? a unified benchmark for backdoor and data poisoning attacks. In: International Conference on Machine Learning. pp. 9389–9398. PMLR (2021)
23. Shejwalkar, V., Houmansadr, A.: Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning. In: NDSS (2021)
24. So, J., Güler, B., Avestimehr, A.S.: Byzantine-resilient secure federated learning. IEEE Journal on Selected Areas in Communications (2020)
25. Sun, F., Zhang, Z., Zeadally, S., Han, G., Tong, S.: Edge computing-enabled internet of vehicles: Towards federated learning empowered scheduling. IEEE Transactions on Vehicular Technology **71**(9), 10088–10103 (2022)
26. Tran, N.H., Bao, W., Zomaya, A., Nguyen, M.N., Hong, C.S.: Federated learning over wireless networks: Optimization model design and analysis. In: IEEE INFOCOM 2019-IEEE Conference on Computer Communications. pp. 1387–1395 (2019)
27. Wang, J., Liu, Q., Liang, H., Joshi, G., Poor, H.V.: Tackling the objective inconsistency problem in heterogeneous federated optimization. Advances in neural information processing systems **33**, 7611–7623 (2020)
28. Wu, Z., Ling, Q., Chen, T., Giannakis, G.B.: Federated variance-reduced stochastic gradient descent with robustness to byzantine attacks. IEEE Transactions on Signal Processing **68**, 4583–4596 (2020)
29. Xie, C., Koyejo, O., Gupta, I.: Generalized byzantine-tolerant sgd. arXiv preprint arXiv:1802.10116 (2018)
30. Xie, C., Koyejo, O., Gupta, I.: Fall of empires: Breaking byzantine-tolerant sgd by inner product manipulation. In: Uncertainty in Artificial Intelligence. pp. 261–270 (2020)
31. Xie, C., Koyejo, S., Gupta, I.: Practical distributed learning: Secure machine learning with communication-efficient local updates. In: European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD) (2019)
32. Xie, C., Koyejo, S., Gupta, I.: Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance. In: International Conference on Machine Learning. pp. 6893–6901 (2019)
33. Xie, C., Koyejo, S., Gupta, I.: Zeno++: Robust fully asynchronous sgd. In: International Conference on Machine Learning. pp. 10495–10503 (2020)
34. Xu, J., Glicksberg, B.S., Su, C., Walker, P., Bian, J., Wang, F.: Federated learning for healthcare informatics. Journal of Healthcare Informatics Research **5**(1), 1–19 (2021)
35. Yang, Z., Chen, M., Wong, K.K., Poor, H.V., Cui, S.: Federated learning for 6g: Applications, challenges, and opportunities. Engineering **8**, 33–41 (2022)
36. Yin, D., Chen, Y., Kannan, R., Bartlett, P.: Byzantine-robust distributed learning: Towards optimal statistical rates. In: International Conference on Machine Learning. pp. 5650–5659 (2018)