Mean Aggregator Is More Robust than Robust Aggregators Under Label Poisoning Attacks

Jie Peng[1] , **Weiyu Li**[2] , **Qing Ling**[1]
[1] Sun Yat-Sen University
[2]Harvard University
pengj95@mail2.sysu.edu.cn, weiyuli@g.harvard.edu, lingqing556@mail.sysu.edu.cn

Abstract

Robustness to malicious attacks is of paramount importance for distributed learning. Existing works often consider the classical Byzantine attacks model, which assumes that some workers can send arbitrarily malicious messages to the server and disturb the aggregation steps of the distributed learning process. To defend against such worst-case Byzantine attacks, various robust aggregators have been proven effective and much superior to the often-used mean aggregator. In this paper, we show that robust aggregators are too conservative for a class of weak but practical malicious attacks, as known as label poisoning attacks, where the sample labels of some workers are poisoned. Surprisingly, we are able to show that the mean aggregator is more robust than the state-of-the-art robust aggregators in theory, given that the distributed data are sufficiently heterogeneous. In fact, the learning error of the mean aggregator is proven to be optimal in order. Experimental results corroborate our theoretical findings, demonstrating the superiority of the mean aggregator under label poisoning attacks.

1 Introduction

With the rising and rapid development of large machine learning models, distributed learning attracts more and more attention by researchers due to its provable effectiveness in solving large-scale problems [Verbraeken *et al.*, 2020]. In distributed learning, there often exist one parameter server (called server thereafter) owing the global model and some computation devices (called workers thereafter) owing the local data. In the training process, the server sends the global model to the workers, and the workers use their local data to compute the local gradients on the global model and send them back to the server. Upon receiving the local gradients from all workers, the server aggregates them and uses the aggregated gradient to update the global model. After the training process, the trained global model is evaluated on the testing data. One important application of distributed learning is federated learning [Ye *et al.*, 2023; Gosselin *et al.*, 2022; Yang *et al.*, 2019; McMahan *et al.*, 2017], which is particularly favorable in terms of privacy preservation.

However, the distributed nature of the server-worker architecture is vulnerable to malicious attacks during the learning process. Due to data corruptions, equipment failures, or cyber attacks, some workers may not follow the algorithmic protocol and send incorrect messages to the server. Previous works often characterize these attacks by the classical Byzantine attacks model, which assumes that some workers can send arbitrarily malicious messages to the server so that the aggregation steps of the learning process is disturbed [Lamport *et al.*, 1982]. For such worst-case Byzantine attacks, various robust aggregators have been proven effective and much superior to the mean aggregator [Chen *et al.*, 2017; Xia *et al.*, 2019; Karimireddy *et al.*, 2021; Wu *et al.*, 2023].

The malicious attacks encountered in reality, on the other hand, are often less destructive than the worst-case Byzantine attacks. For example, a distributed learning system may often suffer from label poisoning attacks, which are weak yet of practical interest. Considering a highly secure email system in a large organization (for example, government or university), if hackers (some users) aim to disturb the online training process of a spam detection model, one of the most effective ways for them is to mislabel received emails from "spam" to "non-spam", resulting in label poisoning attacks. Similar attacks may happen in fraudulent short message service (SMS) detection held by large communication corporations, too.

To this end, in this paper, we consider label poisoning attacks where some workers have local data with poisoned labels and compute the incorrect messages during the learning process. Under label poisoning attacks and with some mild assumptions, surprisingly we are able to show that the mean aggregator is more robust than the state-of-the-art robust aggregators in theory, given that the distributed data are sufficiently heterogeneous. The main contributions of this paper are summarized as follows.

C1) To the best of our knowledge, our work is the first to investigate the robustness of the mean aggregator under attacks in distributed learning. Our work reveals an important fact that the mean aggregator is more robust than the existing robust aggregators under specific types of malicious attacks, which motivates us to rethink the usage of different aggregators within practical scenarios.

C2) Under label poisoning attacks, we theoretically analyze the learning errors of the mean aggregator and the state-of-the-art robust aggregators. The results show that when the heterogeneity of the distributed data is large, the learning error of the mean aggregator is optimal in order regardless of the fraction of poisoned workers.

C3) We empirically evaluate the performance of the mean aggregator and the existing robust aggregators under label poisoning attacks. The experimental results fully support our theoretical findings.

2 Related Works

Poisoning attacks can be categorized into targeted attacks and untargeted attacks; or model poisoning attacks and data poisoning attacks [Kairouz *et al.*, 2021]. In this paper, we focus on the latter categorization. In model poisoning attacks, the malicious workers transmit arbitrarily poisoned models to the server, while data poisoning attacks yield poisoned messages by fabricating poisoned data at the malicious workers' side [Shejwalkar *et al.*, 2022]. Below we briefly review the related works of the two types of poisoning attacks in distributed learning, respectively.

Under model poisoning attacks, most of the existing works design robust aggregators for aggregating local gradients of workers and filter out the potentially poisoned messages. The existing robust aggregators include Krum [Blanchard *et al.*, 2017], geometric median [Chen *et al.*, 2017], coordinate-wise median [Yin *et al.*, 2018], coordinate-wise trimmed-mean [Yin *et al.*, 2018], FABA [Xia *et al.*, 2019], centered clipping [Karimireddy *et al.*, 2021], etc. The key idea behind these robust aggregators is to find a point that has bounded distance to the true gradient such that the learning error is under control. [Farhadkhani *et al.*, 2022a] and [Allouah *et al.*, 2023] propose a unified framework to analyze the performance of these robust aggregators under attacks. Though these methods work well when the data distributions are the same over the workers, their performance degrades when the data distributions become heterogeneous [Karimireddy *et al.*, 2022; Li *et al.*, 2019]. To address this issue, [Li *et al.*, 2019] suggests using model aggregation rather than gradient aggregation to defend against model poisoning attacks in the heterogeneous case. [Karimireddy *et al.*, 2022; Peng *et al.*, 2022; Allouah *et al.*, 2023] propose to use the bucketing/resampling and nearest neighbor mixing techniques to reduce the heterogeneity of the gradients, prior to aggregation. There also exist some works focusing on decentralized optimization without a server, under model poisoning attacks [Wu *et al.*, 2023; He *et al.*, 2022; Peng *et al.*, 2021]. Nevertheless, we focus on distributed learning with a server in this paper.

There are a large amount of papers focusing on data poisoning attacks [Bagdasaryan *et al.*, 2020; Wang *et al.*, 2020; Sun *et al.*, 2019; Rosenfeld *et al.*, 2020]. To defend against data poisoning attacks, the existing works use data sanitization to remove poisoned data [Steinhardt *et al.*, 2017], and prune activation units that are inactive on clean data [Liu *et al.*, 2018]. For more defenses against data poisoning attacks, we refer the reader to the survey paper [Kairouz *et al.*, 2021].

In practice, however, attacks may not necessarily behave as arbitrarily malicious as the above well-established works consider. Some weaker attacks models are structured; for example, [Tavallali *et al.*, 2022] considers the label poisoning attacks in which some workers mislabel their local data and compute the incorrect messages using those poisoned data. Specifically, [Tolpegin *et al.*, 2020; Lin *et al.*, 2021; Jebreel *et al.*, 2024; Jebreel and Domingo-Ferrer, 2023] consider the case where some workers flip the labels of their local data from source classes to target classes. Notably, label poisoning is a kind of data poisoning but not necessarily the worst-case attack, since label poisoning attacks fabricates the local data, yet only on the label level.

It has been shown that label poisoning attacks are equivalent to model poisoning attacks in essence [Farhadkhani *et al.*, 2022b]. Therefore, defenses designed for model poisoning attacks are still effective, including Krum, geometric median, coordinate-wise median, coordinate-wise trimmed-mean, FABA, and centered clipping, as validated by [Karimireddy *et al.*, 2022; Gorbunov *et al.*, 2022; Fang *et al.*, 2020]. There also exist some works designing robust aggregators based on specific properties of label poisoning. For example, the work of [Tavallali *et al.*, 2022] proposes regularization-based defense to detect and exclude the samples with flipped labels in the training process. However, [Tavallali *et al.*, 2022] requires to access a clean validation set, which has privacy concerns in distributed learning. Another work named as LFighter [Jebreel *et al.*, 2024], is the state-of-the-art defense for label poisoning attacks in federated learning. [Jebreel *et al.*, 2024] proposes to cluster the local gradients of all workers, identify the smaller and denser clusters as the potentially poisoned gradients, and discard them. The key idea of LFighter is that the difference between the gradients connected to the source and target output neurons of poisoned workers and regular workers becomes larger when the training process evolves. Therefore it is able to identify the potentially poisoned gradients. However, LFighter only works well when data distributions at different workers are similar. If the heterogeneity of the distributed data is large, the performance of LFighter degrades, as we will show in Section 5.

A recent work of [Shejwalkar *et al.*, 2022], similar to our findings, reveals the robustness of the mean aggregator under poisoning attacks in production federated learning systems. Nevertheless, their study is restrictive in terms of the poisoning ratio (for example, less than 0.1% workers are poisoned while we can afford 10% in the numerical experiments) and lacks theoretical analysis. In contrast, we provide both theoretical analysis and experimental validations.

In conclusion, our work is the first one to investigate the robustness of the mean aggregator under attacks in distributed learning. It reveals an important fact that robust aggregators cannot always outperform the mean aggregator under specific attacks, promoting us to rethink the application scenarios for the use of robust aggregators.

3 Problem Formulation

Consider a distributed learning system with one server and W workers. Denote the set of workers as $\mathcal{W}$ with $|\mathcal{W}| = W$, and the set of regular workers as $\mathcal{R}$ with $|\mathcal{R}| = R$. Note that the number and identities of the regular workers are unknown. Our goal is to solve the following distributed learning problem defined over the regular workers in $\mathcal{R}$, at the presence of the set of poisoned workers $\mathcal{W} \setminus \mathcal{R}$:

$$\min_{x \in \mathbb{R}^D} f(x) \triangleq \frac{1}{R} \sum_{w \in \mathcal{R}} f_w(x), \tag{1}$$

$$\text{with} \;\; f_w(x) \triangleq \frac{1}{J} \sum_{j=1}^{J} f_{w,j}(x), \;\; \forall w \in \mathcal{R}.$$

Here, $x \in \mathbb{R}^D$ is the global model, and $f_w(x)$ is the local cost of worker $w \in \mathcal{R}$ that averages the costs $f_{w,j}(x)$ of J samples. Without loss of generality, we assume that all workers have the same number of samples J.

We begin with characterizing the behaviors of the poisoned workers in $\mathcal{W} \setminus \mathcal{R}$. Different to the classical Byzantine attacks model that assumes some workers to disobey the algorithmic protocol and send arbitrarily malicious messages to the server [Lamport *et al.*, 1982], here we assume the poisoned workers to: (i) have samples with poisoned labels; (ii) exactly follow the algorithmic protocol during the distributed learning process. The formal definition is given as follows.

Definition 1 (Label poisoning attacks). *In solving* (1)*, there exist a number of poisoned workers, whose local costs are in the same form as the regular workers but a fraction of sample labels are poisoned. Nevertheless, these poisoned workers exactly follow the algorithmic protocol during the distributed learning process.*

To solve (1) with the distributed gradient descent algorithm, the server needs to average the local gradients of the regular workers at each iteration. However, as we have emphasized, the number and identities of the regular workers are unknown, such that the server cannot distinguish the true local gradients of the regular workers and the poisoned local gradients from the poisoned workers. We call the true and poisoned local gradients as messages, which the server must judiciously aggregate.

Let the global model be x^t at iteration t. Denote the true local gradient of regular worker $w \in \mathcal{R}$ as $\nabla f_w(x^t)$ and the poisoned local gradient of poisoned worker $w \in \mathcal{W} \setminus \mathcal{R}$ as $\nabla \tilde{f}_w(x^t)$. For notational convenience, we denote the message sent by worker w, no matter true or poisoned, as:

$$\nabla \hat{f}_w(x^t) = \begin{cases} \nabla f_w(x^t), & w \in \mathcal{R}, \\ \nabla \tilde{f}_w(x^t), & w \in \mathcal{W} \setminus \mathcal{R}. \end{cases} \tag{2}$$

Upon receiving all the messages $\{\nabla \hat{f}_w(x^t) : w \in \mathcal{W}\}$, the server can aggregate them with a robust aggregator $\text{RAgg}(\cdot)$ and then move a step along the negative direction:

$$x^{t+1} = x^t - \gamma \cdot \text{RAgg}(\{\nabla \hat{f}_w(x^t) : w \in \mathcal{W}\}), \tag{3}$$

where $\gamma > 0$ is the step size. State-of-the-art robust aggregators include trimmed mean (TriMean) [Chen *et al.*, 2017], centered clipping (CC) [Karimireddy *et al.*, 2021], FABA [Xia *et al.*, 2019], to name a few.

In this paper, we argue that the mean aggregator $\text{Mean}(\cdot)$, which is often viewed as vulnerable, is more robust than state-of-the-art robust aggregators under label poisoning attacks. With the mean aggregator, the update is

$$x^{t+1} = x^t - \gamma \cdot \text{Mean}(\{\nabla \hat{f}_w(x^t) : w \in \mathcal{W}\}), \tag{4}$$

Algorithm 1

Input: Initialization $x^0 \in \mathbb{R}^D$, step size $\gamma > 0$, and number of overall iterations T.

for $t = 0, 1, \cdots, T-1$ **do**
    Server broadcasts x^t to all workers.
    Regular worker $w \in \mathcal{R}$ computes $\nabla f_w(x^t)$ and
      sends $\nabla \hat{f}_w(x^t) = \nabla f_w(x^t)$ to server.
    Poisoned worker $w \in \mathcal{W} \setminus \mathcal{R}$ computes $\nabla \tilde{f}_w(x^t)$
      and sends $\nabla \hat{f}_w(x^t) = \nabla \tilde{f}_w(x^t)$ to server.
    Server receives $\{\nabla \hat{f}_w(x^t)\}_{w \in \mathcal{W}}$ from all workers
      and updates x^{t+1} according to (3) or (4).
end for

Output: $\hat{x} = x^\tau$ where

$$\tau \in \mathop{\arg\min}_{0 \le t \le T-1} \|\text{RAgg}(\{\nabla \hat{f}_w(x^t) : w \in \mathcal{W}\})\|,$$

or

$$\tau \in \mathop{\arg\min}_{0 \le t \le T-1} \|\text{Mean}(\{\nabla \hat{f}_w(x^t) : w \in \mathcal{W}\})\|.$$

where

$$\text{Mean}(\{\nabla \hat{f}_w(x^t) : w \in \mathcal{W}\}) \triangleq \frac{1}{W} \sum_{w \in \mathcal{W}} \nabla \hat{f}_w(x^t). \tag{5}$$

We summarize the distributed learning algorithm with different aggregators in Algorithm 1.

4 Convergence Analysis

In this section, we analyze the learning errors of Algorithm 1 with different aggregators under label poisoning attacks. We make the following assumptions.

Assumption 1 (Lower boundedness). *The global cost $f(\cdot)$ is lower bounded by f^*, namely, $f(x) \ge f^*$.*

Assumption 2 (Lipschitz continuous gradients). *The global cost $f(\cdot)$ has L-Lipschitz continuous gradients. Namely, for any $x, y \in \mathbb{R}^D$, it holds that*

$$\|\nabla f(x) - \nabla f(y)\| \le L \|x - y\|. \tag{6}$$

Assumption 3 (Bounded heterogeneity). *For any $x \in \mathbb{R}^D$, the maximum distance between the local gradients of regular workers $w \in \mathcal{R}$ and the global gradient is upper-bounded by ξ, namely,*

$$\max_{w \in \mathcal{R}} \|\nabla f_w(x) - \nabla f(x)\| \le \xi. \tag{7}$$

Assumptions 1, 2 and 3 are all common in the analysis of distributed first-order algorithms. In particular, Assumption 3 characterizes the heterogeneity of the distributed data across the regular workers; larger ξ means higher heterogeneity.

Assumption 4 (Bounded disturbances of poisoned local gradients). *For any $x \in \mathbb{R}^D$, the maximum distance between the poisoned local gradients of poisoned workers $w \in \mathcal{W} \setminus \mathcal{R}$ and the global gradient is upper-bounded by A, namely,*

$$\max_{w \in \mathcal{W} \setminus \mathcal{R}} \|\nabla \tilde{f}_w(x) - \nabla f(x)\| \le A. \tag{8}$$

Assumption 4 bounds the disturbances caused by the poisoned workers. This assumption does not hold for the worst-case Byzantine attacks model, where the disturbances caused by the Byzantine workers can be arbitrary. However, under label poisoning attacks, we prove that this assumption holds for distributed softmax regression as follows. We will also demonstrate with numerical experiments that this assumption holds naturally in training neural networks.

4.1 Justification of Assumption 4

Example: Distributed softmax regression under label poisoning attacks. Distributed softmax regression is common for classification tasks, where the local cost of worker $w \in \mathcal{W}$ is in the form of

$$f_w(x) \tag{9}$$
$$= -\frac{1}{J}\sum_{j=1}^{J}\sum_{k=1}^{K}\mathbf{1}\{b^{(w,j)} = k\}\log\frac{\exp(x_k^T a^{(w,j)})}{\sum_{l=1}^{K}\exp(x_l^T a^{(w,j)})}.$$

In (9), K stands for the number of classes; $(a^{(w,j)}, b^{(w,j)})$ represents the j-th sample of worker $w \in \mathcal{W}$ with $a^{(w,j)} \in \mathbb{R}^d$ and $b^{(w,j)} \in \mathbb{R}$ being the feature and the label, respectively; $\mathbf{1}\{b^{(w,j)} = k\}$ is the indicator function that outputs 1 if $b^{(w,j)} = k$ and 0 otherwise; $x_k \triangleq [x]_{kd:(k+1)d} \in \mathbb{R}^d$ is the k-th block of x. Note that for poisoned worker $w \in \mathcal{W} \setminus \mathcal{R}$, the labels $b^{(w,j)}$ are possibly poisoned for all $j \in [J]$.

It is easy to verify that the global cost $f(x)$ with the local costs $f_w(x)$ in (9) satisfies Assumptions 1 and 2. Since the gradients of the local costs $f_w(x)$ in (9) are bounded (see the supplementary material), the global cost $f(x)$ satisfies Assumption 3 and ξ refers to the heterogeneity of the local costs $f_w(x)$. Next, we show that Assumption 4 also holds.

Lemma 1. *Consider the distributed softmax regression problem where the local costs of the workers are in the form of* (9)*. The poisoned workers are under label poisoning attacks, with arbitrary fractions of sample labels being poisoned. If $a^{(w,j)}$ is entry-wise non-negative for all $w \in \mathcal{W}$ and all $j \in [J]$, then Assumption 4 is satisfied with*

$$A = 2\sqrt{K}\max_{w\in\mathcal{W}}\|\frac{1}{J}\sum_{j=1}^{J}a^{(w,j)}\|. \tag{10}$$

Lemma 1 explicitly gives the constant A for Assumption 4. Observe that the non-negativity assumption of $a^{(w,j)}$ naturally holds; for example, in image classification tasks, each entry of the feature stands for a pixel value. For other tasks, we can shift the features to meet this requirement.

Relation between Assumptions 3 and 4. Interestingly, the constants ξ and A in Assumptions 3 and 4 are related. Similar to Lemma 1 that gives A, for the distributed softmax regression problem, we can give ξ as follows.

Lemma 2. *Consider the distributed softmax regression problem where the local costs of the regular workers are in the form of* (9)*. If $a^{(w,j)}$ is entry-wise non-negative for all $w \in \mathcal{R}$ and all $j \in [J]$, then Assumption 3 is satisfied with*

$$\xi \le 2\sqrt{K}\max_{w\in\mathcal{R}}\|\frac{1}{J}\sum_{j=1}^{J}a^{(w,j)}\|. \tag{11}$$

In particular, when the distributed data across the regular workers are sufficiently heterogeneous, the constant ξ is close to $2\sqrt{K}\max_{w\in\mathcal{R}}\|\frac{1}{J}\sum_{j=1}^{J}a^{(w,j)}\|$ (see the supplementary material). Further, if the feature norms of the regular and poisoned workers have similar magnitudes, which generally holds in practice, then $2\sqrt{K}\max_{w\in\mathcal{R}}\|\frac{1}{J}\sum_{j=1}^{J}a^{(w,j)}\|$ is in the same order as $2\sqrt{K}\max_{w\in\mathcal{W}}\|\frac{1}{J}\sum_{j=1}^{J}a^{(w,j)}\|$. Hence, we can conclude that $A = O(\xi)$ when the distributed data are sufficiently heterogeneous. This conclusion will be useful in our ensuing analysis.

4.2 Main Results

To analyze the learning errors of Algorithm 1 with robust aggregators, we need to characterize the approximation abilities of the robust aggregators, namely, how close their outputs are to the average of the messages from the regular workers. This gives rise to the definition of ρ-robust aggregator [Wu *et al.*, 2023; Dong *et al.*, 2024].

Definition 2 (**ρ-robust aggregator**)**.** *Consider any W vectors $y_1, y_2, \cdots, y_W \in \mathbb{R}^D$, among which R vectors are from regular workers $w \in \mathcal{R}$. An aggregator RAgg$(\cdot)$ is said to be a ρ-robust aggregator if there exists a contraction constant $\rho \ge 0$ such that*

$$\|RAgg(\{y_1, \cdots, y_W\}) - \bar{y}\| \le \rho \cdot \max_{w\in\mathcal{R}}\|y_w - \bar{y}\|, \tag{12}$$

where $\bar{y} = \frac{1}{R}\sum_{w\in\mathcal{R}} y_w$ is the average vector of the regular workers.

From Definition 2, a small contraction constant ρ means that the output of the robust aggregator is close to the average of the messages from the regular workers. The error is proportional to the heterogeneity of the messages from the regular workers, characterized by $\max_{w\in\mathcal{R}}\|y_w - \bar{y}\|$.

However, since a robust aggregator cannot distinguish the regular and poisoned workers, ρ is unable to be arbitrarily close to 0. Additionally, when the messages from the poisoned workers are majority, there is no guarantee to satisfy Definition 2. Therefore, we have the following lemma.

Lemma 3. *Denote $\delta \triangleq 1 - \frac{R}{W}$ as the fraction of the poisoned workers. Then a ρ-robust aggregator exists only if $\delta < \frac{1}{2}$ and $\rho \ge \min\{\frac{\delta}{1-2\delta}, 1\}$.*

We prove that several state-of-the-art robust aggregators, such as TriMean [Chen *et al.*, 2017], CC [Karimireddy *et al.*, 2021] and FABA [Xia *et al.*, 2019], all satisfy Definition 2 when the fraction of poisoned workers is below their respective thresholds. Their corresponding contraction constants ρ are given in the supplementary material [Peng *et al.*, 2024].

Remark 1. *Our definition is similar to (f, κ)-robustness in [Allouah* et al.*, 2023], while our heterogeneity measure is $\max_{w\in\mathcal{R}}\|y_w - \bar{y}\|$ instead of $\frac{1}{R}\sum_{w\in\mathcal{R}}\|y_w - \bar{y}\|^2$. Due to the fact $\max_{w\in\mathcal{R}}\|y_w - \bar{y}\|^2 \le \sum_{w\in\mathcal{R}}\|y_w - \bar{y}\|^2$, our definition implies (f, κ)-robustness in [Allouah* et al.*, 2023]. Further according to Propositions 8 and 9 in [Allouah* et al.*, 2023], our definition also implies (f, λ)-resilient averaging and $(\delta_{\max}, c)$-ARAgg in [Farhadkhani* et al.*, 2022a] and [Karimireddy* et al.*, 2022], respectively.*

Thanks to the contraction property in Definition 2, we can prove that the learning error of Algorithm 1 with a ρ-robust aggregator is bounded under label poisoning attacks.

Theorem 1. *Consider Algorithm 1 with a ρ-robust aggregator RAgg$(\cdot)$ to solve* (1)*. Under label poisoning attacks where the fraction of poisoned workers is $\delta \in [0, \frac{1}{2})$, if the step size is $\gamma \in (0, \frac{1}{L}]$ and Assumptions 1, 2 and 3 are satisfied, then we have*

$$\|\nabla f(\hat{x})\|^2 \leq \frac{8(f(x^0) - f^*)}{\gamma T} + 10\rho^2\xi^2. \tag{13}$$

Interestingly, we are also able to prove that Algorithm 1 with the mean aggregator, under label poisoning attacks, has a bounded learning error.

Theorem 2. *Consider Algorithm 1 with the mean aggregator Mean$(\cdot)$ to solve* (1)*. Under label poisoning attacks where the fraction of poisoned workers is $\delta \in [0, 1)$, if the step size is $\gamma \in (0, \frac{1}{L}]$ and Assumptions 1, 2 and 4 are satisfied, then we have*

$$\|\nabla f(\hat{x})\|^2 \leq \frac{8(f(x^0) - f^*)}{\gamma T} + 10\delta^2 A^2. \tag{14}$$

Theorems 1 and 2 demonstrate that Algorithm 1 with both ρ-robust aggregators and the mean aggregator can sublinearly converge to neighborhoods of a first-order stationary point of (1), and non-vanishing learning errors are $\rho^2\xi^2$ for ρ-robust aggregators and $\delta^2 A^2$ for the mean aggregator. Note that the $O(\frac{1}{T})$ convergence rates are optimal for first-order nonconvex optimization algorithms [Carmon *et al.*, 2020].

Before comparing the learning errors in Theorems 1 and 2, we give the lower bound of the learning error for Algorithm 1 with either a ρ-robust aggregator or the mean aggregator.

Theorem 3. *Under label poisoning attacks with $\delta = 1 - \frac{R}{W}$ fraction of poisoned workers, consider Algorithm 1 with any ρ-robust aggregator or the mean aggregator, where $\rho \geq \min\{\frac{\delta}{1-2\delta}, 1\}$. There exist R regular local functions $\{f_w(x) : w \in \mathcal{R}\}$ and $W - R$ poisoned local functions $\{\tilde{f}_w(x) : w \in \mathcal{W} \setminus \mathcal{R}\}$ satisfying Assumptions 1, 2 , 3 and 4 such that the output $\hat{x}$ of Algorithm 1 has*

$$\|\nabla f(\hat{x})\|^2 \geq \Omega(\delta^2\xi^2). \tag{15}$$

Since the identities of poisoned workers are unknown, it is impossible to fully eliminate the effect of malicious attacks, which results in the non-vanishing learning error as demonstrated in Theorem 3. In Table 1, we compare the learning errors for different aggregators, given large heterogeneity such that A is in the same order as ξ (which holds when the distributed data are sufficiently heterogeneous, as we have discussed in Section 4.1).

According to Table 1, we know that the learning errors of TriMean, FABA and the mean aggregator all match the lower bound in order, when δ is small. However, the learning errors of TriMean and FABA explode when δ approaches $\frac{1}{2}$ and $\frac{1}{3}$, respectively, while the mean aggregator is insensitive. Therefore, the learning error of the mean aggregator is optimal in order regardless of the fraction of poisoned workers. In addition, the learning error of the mean aggregator is smaller than that of CC by a magnitude of δ.

Aggregator	Learning error
TriMean	$O(\frac{\delta^2\xi^2}{(1-2\delta)^2})$
CC	$O(\delta\xi^2)$
FABA	$O(\frac{\delta^2\xi^2}{(1-3\delta)^2})$
Mean	$O(\delta^2\xi^2)$
Lower bound	$\Omega(\delta^2\xi^2)$

Table 1: Learning errors of Algorithm 1 with TriMean, CC, FABA and the mean aggregator when the heterogeneity of distributed data is sufficiently large such that A is in the same order as ξ. The lower bound of using any ρ-robust aggregator and the mean aggregator is also given.

Remark 2. *We analyze Algorithm 1 in the deterministic setting, but will evaluate its performance in the stochastic setting (namely, sampling a mini-batch of samples at each iteration), due to the high computation overhead of computing full gradients. Extending the analysis from the deterministic setting to stochastic is non-trivial, due to the difficulty of handling the variance of stochastic gradients in establishing the upper bounds and the tight lower bound. We will investigate this issue in our future work.*

5 Numerical Experiments

In this section, we conduct numerical experiments to validate our theoretical findings and demonstrate the performance of Algorithm 1 with the mean and robust aggregators under label poisoning attacks.

5.1 Experimental Settings

Datasets and partitions. For the convex problem, consider softmax regression on the MNIST dataset. As for the nonconvex problem, we train multi-layer perceptrons on the MNIST dataset and convolutional neural networks on the CIFAR10 dataset. We setup $W = 10$ workers where $R = 9$ workers are regular and the remaining one is poisoned. We consider three data distributions: i.i.d., mild non-i.i.d. and non-i.i.d. cases. In the i.i.d. case, we uniformly randomly divide the training data among all workers. In the mild non-i.i.d. case, we divide the training data using the Dirichlet distribution with hyper-parameter $\alpha = 1$ by default [Ronning, 1989]. In the non-i.i.d. case, we assign each class of the training data to one worker.

Label poisoning attacks. We investigate two types of label poisoning attacks: static label flipping where the poisoned worker flips label b to $9 - b$ with b ranging from 0 to 9, and dynamic label flipping where the poisoned worker flips label b to the least probable label with respect to the global model x^t [Shejwalkar *et al.*, 2022].

Aggregators to compare. We are going to compare the mean aggregator with several representative ρ-robust aggregators, including TriMean, FABA, CC, and LFighter. The baseline is the mean aggregator without attacks[1].

[1]The code is available at https://github.com/pengj97/LPA

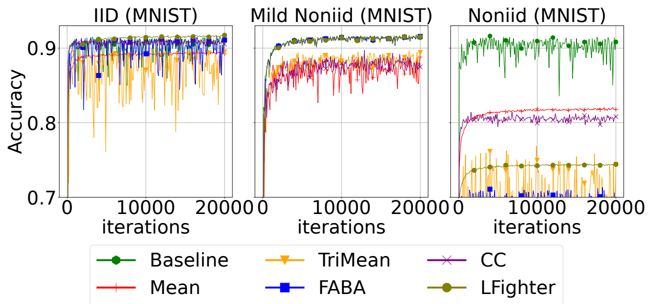

Figure 1: Classification accuracies of softmax regression on the MNIST dataset under static label flipping attacks.

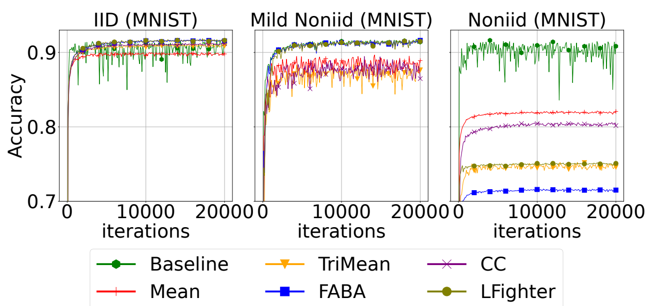

Figure 2: Classification accuracies of softmax regression on the MNIST dataset under dynamic label flipping attacks.

5.2 Convex Case

Classification accuracy. We consider softmax regression on the MNIST dataset. The classification accuracies under static label flipping and dynamic label flipping attacks are shown in Figure 1 and Figure 2, respectively. In the i.i.d. case, all methods perform well and close to the baseline. In the mild non-i.i.d. case, FABA and LFighter are the best among all aggregators and the other aggregators have similar performance. In the non-i.i.d. case, since the heterogeneity is large, all aggregators are tremendously affected by the label poisoning attacks, and have gaps to the baseline in terms of classification accuracy. Notably, the mean aggregator performs the best among all aggregators in this case, which validates our theoretical results.

Heterogeneity of regular local gradients and disturbance of poisoned local gradients. To further validate the reasonableness of Assumptions 3 and 4, as well as the correctness of our theoretical results in Section 4.1, we compute the smallest ξ and A that satisfy Assumptions 3 and 4 for the softmax regression problem. As shown in Figure 3, the disturbances of the poisoned local gradients, namely A, are bounded under both static label flipping and dynamic label flipping attacks, which corroborates the theoretical results in Lemma 1. From i.i.d., mild non-i.i.d. to the non-i.i.d. case, the heterogeneity of regular local gradients characterized by ξ increases. Particularly, in the non-i.i.d. case, ξ is close to A under both static label flipping and dynamic label flipping attacks, which aligns our discussions below Lemma 2. Recall Table 1 that when the heterogeneity is sufficiently large, the learning error of the mean aggregator is optimal in order. This explains the results in Figures 1 and 2.

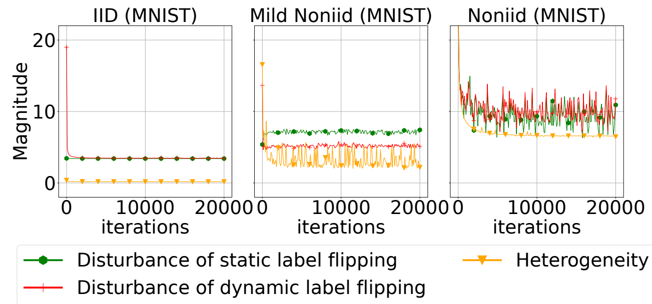

Figure 3: Heterogeneity of regular local gradients (the smallest ξ satisfying Assumption 3) and disturbance of poisoned local gradients (the smallest A satisfying Assumption 4) in softmax regression on the MNIST dataset, under static label flipping and dynamic label flipping attacks.

5.3 Nonconvex Case

Classification accuracy. Next, we train multi-layer perceptrons on the MNIST dataset and neural networks on the CIFAR10 dataset under static label flipping and dynamic label flipping attacks, as depicted in Figures 4 and 5. In the i.i.d. case, all methods have good performance and are close to the baseline, except for TriMean that performs worse than the other aggregators on the CIFAR10 dataset under dynamic label flipping attacks. In the mild non-i.i.d. case and on the MNIST dataset, Mean, FABA and LFighter are the best and close to the baseline, CC and TriMean perform worse, while TriMean is oscillating. On the CIFAR10 dataset, Mean, FABA, CC, and LFighter perform the best, while TriMean performs worse than the other methods with an obvious gap. In the non-i.i.d. case, all methods are affected by the attacks and cannot reach the same classification accuracy of the baseline, but mean aggregator still performs the best. CC, FABA and LFighter perform worse and TriMean fails.

Heterogeneity of regular local gradients and disturbance of poisoned local gradients. We also calculate the smallest values of ξ and A satisfying Assumptions 3 and 4, respectively. As shown in Figure 6, the disturbance of poisoned local gradients measured by A are bounded on the MNIST and CIFAR10 datasets under both static label flipping and dynamic label flipping attacks. From i.i.d., mild non-i.i.d. to the non-i.i.d. case, the heterogeneity of regular local gradients ξ is increasing. In the non-i.i.d. case, ξ is close to A.

5.4 Impacts of Heterogeneity and Attack Strengths

To further show the impacts of heterogeneity of data distributions and strengths of label poisoning attacks, we compute classification accuracies of the trained multi-layer perceptrons neural network on the MNIST dataset, varying the data distributions and the levels of label poisoning attacks. We employ the Dirichlet distribution by varying the hyper-parameter $\alpha = \{100, 1, 0.1, 0.001\}$ to simulate various heterogeneity of data distributions, in which a smaller α corresponds to larger heterogeneity. In addition, we let the poisoned worker apply static label flipping attacks by flipping labels with probability $p = \{0.0, 0.4, 0.7, 1.0\}$ to simulate different attack strengths. A larger flipping probability indicates stronger attacks.

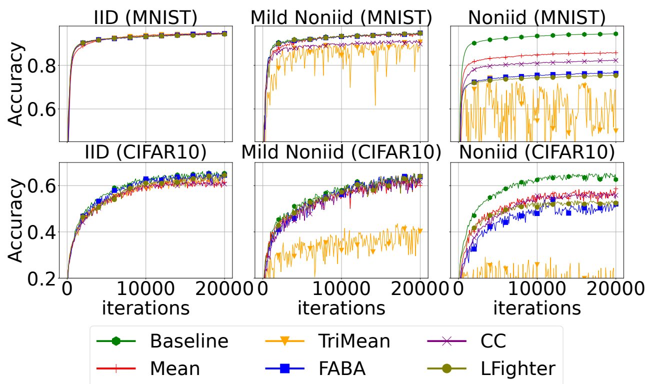

Figure 4: Classification accuracies of multi-layer perceptrons on the MNIST dataset and convolutional neural networks on the CIFAR10 dataset under static label flipping attacks.

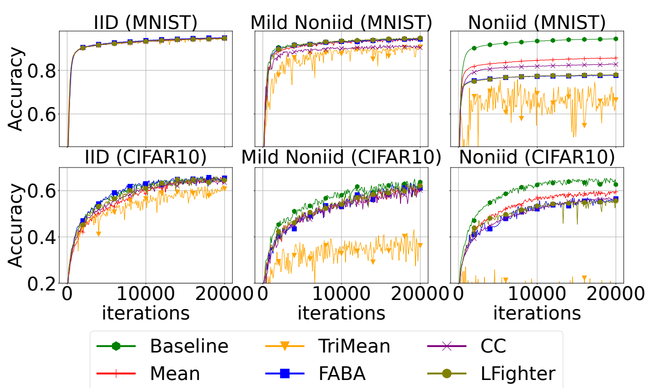

Figure 5: Classification accuracies of multi-layer perceptrons on the MNIST dataset and convolutional neural networks on the CIFAR10 dataset under dynamic label flipping attacks.

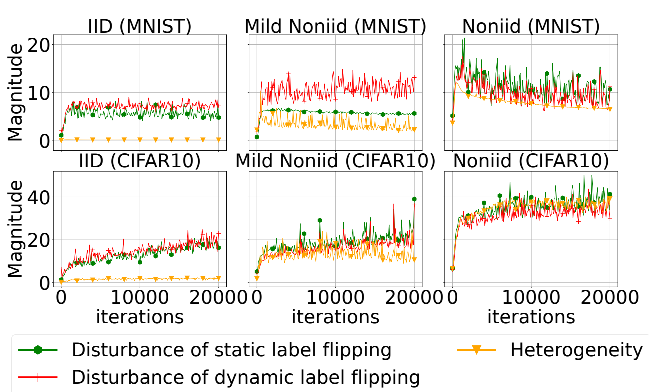

Figure 6: Heterogeneity of regular local gradients (the smallest ξ satisfying Assumption 3) and disturbance of poisoned local gradients (the smallest A satisfying Assumption 4) in training multi-layer perceptrons on the MNIST dataset and training convolutional neural networks on the CIFAR10 dataset, under static label flipping and dynamic label flipping attacks.

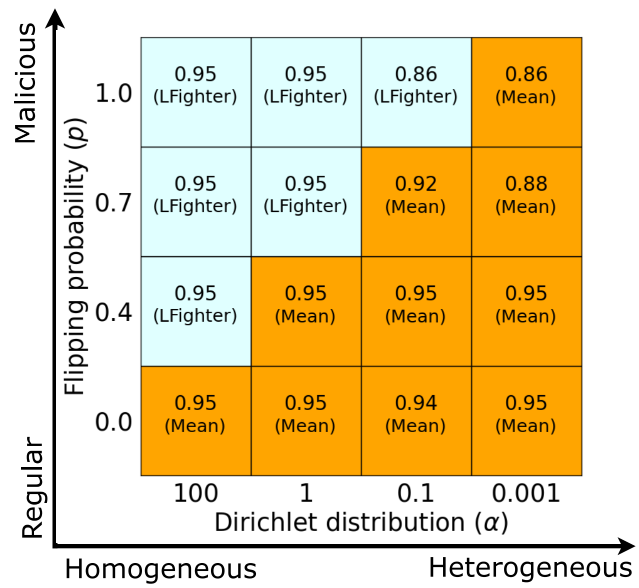

Figure 7: Best classification accuracies of trained multi-layer perceptrons by all aggregators on the MNIST dataset under static label flipping. Each block is associated with a hyper-parameter α that characterizes the heterogeneity and the flipping probability p that characterizes the attack strength. For each block, the best classification accuracy and the corresponding aggregator is marked.

We present the best performance among all aggregators, and mark the corresponding best aggregator in Figure 7. The mean aggregator outperforms the robust aggregators when the heterogeneity is large. For example, the mean aggregator exhibits superior performance when $\alpha = 0.001$ and the flipping probability $p = \{0, 0.4, 0.7, 1.0\}$, as well as when $\alpha = 0.1$ and $p = \{0, 0.4, 0.7\}$. Furthermore, fixing the flipping probability p, when the hyper-parameter α becomes smaller which means that the heterogeneity becomes larger, the mean aggregator gradually surpasses the robust aggregators. Fixing the hyper-parameter α, when the flipping probability p becomes smaller which means that the attack strength becomes smaller, the mean aggregator gradually surpasses the robust aggregators. According to the above observations, we recommend to apply the mean aggregator when the distributed data are sufficiently heterogeneous, or the disturbance caused by label poisoning attacks is comparable to the heterogeneity of regular local gradients.

6 Conclusions

We studied the distributed learning problem subject to label poisoning attacks. We theoretically proved that when the distributed data are sufficiently heterogeneous, the learning error of the mean aggregator is optimal in order. Further corroborated by numerical experiments, our work revealed an important fact that state-of-the-art robust aggregators cannot always outperform the mean aggregator, if the attacks are confined to label poisoning. We expect that this fact can motivate us to revisit which application scenarios are proper for using robust aggregators. Our future work will extend the analysis to the more challenging decentralized learning problem.

Acknowledgments

Qing Ling (corresponding author) is supported by NSF China grants 61973324, 12126610 and 62373388, Guangdong Basic and Applied Basic Research Foundation grant 2021B-1515020094, R&D project of Pazhou Lab (Huangpu) grant 2023K0606, and Guagndong Key Lab of Mathematical Foundations for Artificial Intelligence, The Chinese University of Hong Kong, Shenzhen.

References

[Allouah *et al.*, 2023] Youssef Allouah, Sadegh Farhadkhani, Rachid Guerraoui, Nirupam Gupta, Rafaël Pinot, and John Stephan. Fixing by mixing: a recipe for optimal Byzantine ML under heterogeneity. In *International Conference on Artificial Intelligence and Statistics*, pages 1232–1300, 2023.

[Bagdasaryan *et al.*, 2020] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *International Conference on Artificial Intelligence and Statistics*, pages 2938–2948, 2020.

[Blanchard *et al.*, 2017] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Advances in Neural Information Processing Systems*, pages 118–128, 2017.

[Carmon *et al.*, 2020] Yair Carmon, John C. Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points i. *Mathematical Programming*, 184(1–2):71–120, 2020.

[Chen *et al.*, 2017] Yudong Chen, Lili Su, and Jiaming Xu. Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. *ACM on Measurement and Analysis of Computing Systems*, 1(2):1–25, 2017.

[Dong *et al.*, 2024] Xingrong Dong, Zhaoxian Wu, Qing Ling, and Zhi Tian. Byzantine-robust distributed online learning: taming adversarial participants in an adversarial environment. *IEEE Transactions on Signal Processing*, 72:235–248, 2024.

[Fang *et al.*, 2020] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Gong. Local model poisoning attacks to Byzantine-robust federated learning. In *USENIX Security Symposium*, pages 1605–1622, 2020.

[Farhadkhani *et al.*, 2022a] Sadegh Farhadkhani, Rachid Guerraoui, Nirupam Gupta, Rafael Pinot, and John Stephan. Byzantine machine learning made easy by resilient averaging of momentums. In *International Conference on Machine Learning*, pages 6246–6283, 2022.

[Farhadkhani *et al.*, 2022b] Sadegh Farhadkhani, Rachid Guerraoui, Lê-Nguyên Hoang, and Oscar Villemaud. An equivalence between data poisoning and Byzantine gradient attacks. In *International Conference on Machine Learning*, pages 6284–6323, 2022.

[Gorbunov *et al.*, 2022] Eduard Gorbunov, Samuel Horváth, Peter Richtárik, and Gauthier Gidel. Variance reduction is an antidote to Byzantines: better rates, weaker assumptions and communication compression as a cherry on the top. In *International Conference on Learning Representations*, 2022.

[Gosselin *et al.*, 2022] Rémi Gosselin, Loïc Vieu, Faiza Loukil, and Alexandre Benoit. Privacy and security in federated learning: a survey. *Applied Sciences*, 12(19):9901, 2022.

[He *et al.*, 2022] Lie He, Sai Praneeth Karimireddy, and Martin Jaggi. Byzantine-robust decentralized learning via clippedgossip. *arXiv preprint arXiv:2202.01545*, 2022.

[Jebreel and Domingo-Ferrer, 2023] Najeeb Moharram Jebreel and Josep Domingo-Ferrer. Fl-defender: combating targeted attacks in federated learning. *Knowledge-Based Systems*, 260:110178, 2023.

[Jebreel *et al.*, 2024] Najeeb Moharram Jebreel, Josep Domingo-Ferrer, David Sánchez, and Alberto Blanco-Justicia. Lfighter: defending against the label-flipping attack in federated learning. *Neural Networks*, 170:111–126, 2024.

[Kairouz *et al.*, 2021] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G.L. D'Oliveira, Hubert Eichner, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konečný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Mariana Raykova, Hang Qi, Daniel Ramage, Ramesh Raskar, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 14(1–2):1–210, 2021.

[Karimireddy *et al.*, 2021] Sai Praneeth Karimireddy, Lie He, and Martin Jaggi. Learning from history for Byzantine robust optimization. In *International Conference on Machine Learning*, pages 5311–5319, 2021.

[Karimireddy *et al.*, 2022] Sai Praneeth Karimireddy, Lie He, and Martin Jaggi. Byzantine-robust learning on heterogeneous datasets via bucketing. In *International Conference on Learning Representations*, 2022.

[Lamport *et al.*, 1982] Leslie Lamport, Robert Shostak, and Marshall Pease. The Byzantine generals problem. *ACM Transactions on Programming Languages and Systems*, 4(3):382–401, 1982.

[Li *et al.*, 2019] Liping Li, Wei Xu, Tianyi Chen, Georgios B. Giannakis, and Qing Ling. RSA: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets. In *AAAI Conference on Artificial Intelligence*, pages 1544–1551, 2019.

[Lin *et al.*, 2021] Jing Lin, Long Dang, Mohamed Rahouti, and Kaiqi Xiong. ML attack models: adversarial attacks and data poisoning attacks. *arXiv preprint arXiv:2112.02797*, 2021.

[Liu *et al.*, 2018] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: defending against backdooring attacks on deep neural networks. In *International Symposium on Research in Attacks, Intrusions, and Defenses*, pages 273–294, 2018.

[McMahan *et al.*, 2017] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282, 2017.

[Peng *et al.*, 2021] Jie Peng, Weiyu Li, and Qing Ling. Byzantine-robust decentralized stochastic optimization over static and time-varying networks. *Signal Processing*, 183:108020, 2021.

[Peng *et al.*, 2022] Jie Peng, Zhaoxian Wu, Qing Ling, and Tianyi Chen. Byzantine-robust variance-reduced federated learning over distributed non-iid data. *Information Sciences*, 616:367–391, 2022.

[Peng *et al.*, 2024] Jie Peng, Weiyu Li, and Qing Ling. Mean aggregator is more robust than robust aggregators under label poisoning attacks. *arXiv preprint arXiv:2404.13647*, 2024.

[Ronning, 1989] Gerd Ronning. Maximum likelihood estimation of dirichlet distributions. *Journal of Statistical Computation and Simulation*, 32(4):215–221, 1989.

[Rosenfeld *et al.*, 2020] Elan Rosenfeld, Ezra Winston, Pradeep Ravikumar, and Zico Kolter. Certified robustness to label-flipping attacks via randomized smoothing. In *International Conference on Machine Learning*, pages 8230–8241, 2020.

[Shejwalkar *et al.*, 2022] Virat Shejwalkar, Amir Houmansadr, Peter Kairouz, and Daniel Ramage. Back to the drawing board: a critical evaluation of poisoning attacks on production federated learning. In *IEEE Symposium on Security and Privacy*, pages 1354–1371, 2022.

[Steinhardt *et al.*, 2017] Jacob Steinhardt, Pang Wei Koh, and Percy Liang. Certified defenses for data poisoning attacks. In *Advances in Neural Information Processing Systems*, pages 3520–3532, 2017.

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

[Tavallali *et al.*, 2022] Pooya Tavallali, Vahid Behzadan, Azar Alizadeh, Aditya Ranganath, and Mukesh Singhal. Adversarial label-poisoning attacks and defense for general multi-class models based on synthetic reduced nearest neighbor. In *International Conference on Image Processing*, pages 3717–3722, 2022.

[Tolpegin *et al.*, 2020] Vale Tolpegin, Stacey Truex, Mehmet Emre Gursoy, and Ling Liu. Data poisoning attacks against federated learning systems. In *European Symposium on Research in Computer Security*, pages 480–501, 2020.

[Verbraeken *et al.*, 2020] Joost Verbraeken, Matthijs Wolting, Jonathan Katzy, Jeroen Kloppenburg, Tim Verbelen, and Jan S. Rellermeyer. A survey on distributed machine learning. *Acm Computing Surveys*, 53(2):1–33, 2020.

[Wang *et al.*, 2020] Hongyi Wang, Kartik Sreenivasan, Shashank Rajput, Harit Vishwakarma, Saurabh Agarwal, Jy-yong Sohn, Kangwook Lee, and Dimitris Papailiopoulos. Attack of the tails: yes, you really can backdoor federated learning. In *Advances in Neural Information Processing Systems*, pages 16070–16084, 2020.

[Wu *et al.*, 2023] Zhaoxian Wu, Tianyi Chen, and Qing Ling. Byzantine-resilient decentralized stochastic optimization with robust aggregation rules. *IEEE Transactions on Signal Processing*, 71:3179–3195, 2023.

[Xia *et al.*, 2019] Qi Xia, Zeyi Tao, Zijiang Hao, and Qun Li. FABA: an algorithm for fast aggregation against Byzantine attacks in distributed neural networks. In *International Joint Conferences on Artificial Intelligence*, 2019.

[Yang *et al.*, 2019] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: concept and applications. *ACM Transactions on Intelligent Systems and Technology*, 10(2):1–19, 2019.

[Ye *et al.*, 2023] Mang Ye, Xiuwen Fang, Bo Du, Pong C. Yuen, and Dacheng Tao. Heterogeneous federated learning: state-of-the-art and research challenges. *ACM Computing Surveys*, 56(3):1–44, 2023.

[Yin *et al.*, 2018] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International Conference on Machine Learning*, pages 5650–5659, 2018.