Tolerating Outliers: Gradient-Based Penalties for Byzantine Robustness and Inclusion

Latifa Errami , El Houcine Bergou
College of Computing, Mohammed VI Polytechnic University, Ben Guerir, Morocco
latifa.errami@um6p.ma , elhoucine.bergou@um6p.ma

Abstract

This work investigates the interplay between Robustness and Inclusion in the context of poisoning attacks targeting the convergence of Stochastic Gradient Descent (SGD). While robustness has received significant attention, the standard Byzantine defenses rely on the Independent and Identically Distributed (IID) assumption causing their performance to deteriorate on non-IID data distributions, even without any attack. This is largely due to these defenses being excessively cautious and discarding benign outliers. We introduce a penalty-based aggregation that accounts for the discrepancy between trusted clients and outliers. We propose the use of Linear Scalarization (LS) as an enhancing method to enable current defenses to simultaneously circumvent Byzantine attacks while also granting inclusion of outliers. This empowers existing defenses to not only counteract malicious adversaries effectively but also to incorporate outliers into the learning process. We conduct a theoretical analysis to demonstrate the convergence of our approach. Specifically, we establish the robustness and resilience of our method under standard assumptions. Empirical analysis further validates the viability of the proposed approach. Across mild to strong non-IID data splits, our method consistently either matches or surpasses the performance of current approaches in the literature, under state-of-the-art Byzantine attack scenarios.

1 Introduction

Most real-world applications using learning algorithms are moving towards distributed computation either: (i) Due to some applications being inherently distributed, for instance Federated Learning (FL) [McMahan *et al.*, 2017], (ii) or to speed up computation and benefit from hardware parallelization. We especially resort to distributing Stochastic Gradient Descent (SGD) to alleviate the heavy computation underlying gradient updates during the training phase. This is mainly due to the high dimensionality of large-scale deep learning models and the exponential growth in user-generated data.

However, distributing computation comes at the cost of introducing challenges related to consensus and fault tolerance. In other words, the nodes composing the distributed system need to reach consensus regarding the gradient update. In the honest setting this can be done simply by a Parameter-Server (PS) that takes in charge the aggregation of computation from the workers. However, machines are prone to hardware failure (crash/stop) or arbitrary behavior due to bugs or malicious users. The latter is more concerning as machines may collude and lead to convergence to ineffective models.

Since deep learning pipelines are involved in decision making at a critical level (e.g., computer-aided diagnosis, airport security. . .); it is crucial to ensure their robustness. We study robustness in the sense of granting resilience against malicious adversaries. More precisely, poisoning attacks that target the convergence of SGD. The adversarial model follows the Byzantine abstraction [Lamport *et al.*, 1982]. The basis of distributed Byzantine attacks is the disruption of SGD's convergence by tampering with the direction of the descent or magnitude of the updates. The robustness problem is highly examined and there exists a plethora of aggregations [Blanchard *et al.*, 2017; Alistarh *et al.*, 2018; Yin *et al.*, 2018; Damaskinos *et al.*, 2018; Boussetta *et al.*, 2021; El Mhamdi *et al.*, 2018]. Nonetheless, recent works [Karimireddy *et al.*, 2022; Karimireddy *et al.*, 2021] highlight the inability of these algorithms to learn on non-IID data. Indeed, defending against the Byzantine in a heterogeneous setting is not trivial. Naturally, aggregations rely on the similarity between honest workers to defend against the Byzantine. However, as data becomes unbalanced distinguishing malicious workers from honest ones becomes increasingly challenging: an honest worker may slightly deviate from its peers due to skewed data distribution.

Although, granting robustness is paramount in distributed learning, it need not come at the cost of fairness and model efficiency. In the (PS) architecture most defenses discard a subset of information either by dropping full gradient vectors or by eliminating a set of coordinates along each dimension of the submitted vectors. Nevertheless, dropping users' updates leads to (i) Degradation of the model's final accuracy, especially when no Byzantine attackers are present. (ii) The robust aggregations act too safely at the cost of marginalizing honest clients that exhibit outlier tendencies. One can think of the impact of this strategy of granting robustness in collabo-

rative learning set ups where minorities with vastly diverging views may be actively discarded from the training, as elaborated in [El-Mhamdi *et al.*, 2021a]. Similarly, in the case of a predictive model trained collaboratively among hospitals to facilitate medical diagnosis, robust methods may filter out rare but informative updates computed from records containing rare diseases, atypical symptoms, or unusual responses to treatment by equating outlier with malicious. This, in turn could lead to biased medication recommendations that are less effective for patients with rare conditions or unusual medical profiles.

Motivated by the latter and the scarcity of work on Byzantine non-IID defenses, we study the interplay between robustness and inclusion of outliers and the impact of data heterogeneity on this trade-off. These two problems: (1) learning on non-IID data and (2) defending against Byzantine attacks are similar in the sense that in both cases the central server in charge of aggregating clients' updates is faced with conflicting information. In the Byzantine case the conflict stems from the discrepancy between the updates submitted by the malicious users and honest ones. Whereas, in the non-IID case it is a result of the difference in the data distribution between the users. In a realistic set up such as FL, the central server is faced with both challenges.

From an optimization perspective, the goal is to find the parameters of a model that simultaneously minimizes the loss associated with the local data of each client $\mathcal{L}_i$. More precisely, the objectives in these cases maybe conflicting. This type of optimization problem falls under multi-objective and can be solved as mono-objective through Linear Scalarization eq.1 (LS) . This solution simply consists in introducing a preference vector λ that defines a trade-off between the conflicting objectives. Thereafter, proceed to optimize the weighted sum of the objectives as one

$$\min_{\theta \in \mathbb{R}^d} \left(\mathcal{L}(\theta) \stackrel{\text{def}}{=} \sum_{i=1}^{n} \lambda_i \mathcal{L}_i(\theta) \right) \tag{1}$$

The vector λ is referred to as the preference or trade-off vector and is typically chosen such that $\sum_{i=1}^{n} \lambda_i = 1$ and $\lambda_i > 0$.

Our insight is to treat the Byzantine-robust learning problem as multi-objective, then to deploy LS to solve it. That is, the trade-off vector can be leveraged to balance users' updates by penalizing the suspected Byzantine ones without dropping any user's contribution. Additionally, this scheme addresses the trade-off between granting Byzantine resilience and inclusion as no client update is fully discarded.We note that our objective is not to explicitly find all the Pareto front (PF). Instead, we are interested in identifying a point within the PF where the trade-off vector associated with Byzantine clients approaches zero. This task is particularly challenging, especially in non-IID setups where distinguishing Byzantine from honest workers is difficult. To address this challenge, we propose penalizing users' gradients based on their deviation from a reference gradient g_{ref}. If g_{ref} is accurate enough, gradients deviating significantly from g_{ref} contribute less to the model update (theirs corresponding λ is small). In scenarios where the server has access to a clean validation dataset, computing an approximation of a trade-off vector λ is more straightforward because g_{ref} computed using these data is accurate. This simplifies the process of identifying and appropriately penalizing outliers. However, obtaining validation data can be challenging for certain applications. To address this limitation, we introduce $\mathcal{F} \circ LS$, a method that combines standard aggregations with LS. This hybrid approach serves two main purposes: (i) enhancing the performance of standard aggregations and mitigating the robustness and inclusion trade-off, and (ii) providing a practical alternative in situations where acquiring a validation dataset is challenging.

Our contribution can be summarized as follows:

- We present the first algorithm that grants inclusion of outliers while preserving robustness against a fraction of Byzantine attackers: We propose $\mathcal{F} \circ LS$, a linear scalarization based aggregation that leverages existing defenses to define a trade-off vector over the n-conflicting objectives in order to grant Byzantine resilience and Inclusion of outliers. Moreover, we show that the cost of inclusion can be controlled when associating gradient-based penalties with the re-incorporated clients.
- We conduct theoretical analysis to demonstrate robustness and resilience of our approach under mild assumptions. We mainly show that (i) the deviation of $\mathcal{F} \circ LS$ gradient from the average of the honest users gradients stays controlled by the gradients of the honest users (Robustness), (ii) and that the average of the honest users gradients goes to zero over the iterations (resilience).
- We evaluate our scheme against state of the art Byzantine attacks. We compare our approach to the standard aggregations it enhances to assess the impact granting inclusion has on preserving robustness. We also compare our approach to that of bucketing [Karimireddy *et al.*, 2022] and NNM [Allouah *et al.*, 2023]. Additionally, we investigate the performance of LS when validation data is available at the server.

1.1 Notations and Set Up

Notations. $\mathcal{F}(.)$: Denotes a Robust Aggregation Rule. $\mathcal{L}_i$: The objective function associated with the i-th worker. $\mathcal{G} = \{g_1, g_2, ..., g_n\}$: corresponds to users gradient vectors. $g_{\mathcal{F}}$ the gradient computed by $\mathcal{F}$. $\mathcal{G}_t$, $\mathcal{G}_b$ Respectively denote the sets of gradient vectors associated with the set of trusted workers and suspected workers based on $\mathcal{F}$. λ: the preference or trade-off vector. n the total number of users (called also clients, workers or nodes), among which a subset f are Byzantine. f: The number of Byzantine clients, $f < \frac{n}{2}$. $\mathcal{D}_i = \{(x_j, y_j)\}_{j=1}^{|\mathcal{D}_i|}$ the subset of data of worker i. C_{R_i} denotes a robust metric computed based on the gradient update of user i. S denotes the set of honest users and $|S|$ its cardinality.

Set Up. We consider the standard Parameter Server (**PS**) architecture, comprised of a central server in charge of the exchange between the n nodes. The (PS) architecture assumes that the server is trusted and a proportion δ of the nodes behave arbitrarily or maliciously. The arbitrary behavior is modelled following the Byzantine abstraction introduced in

[Lamport *et al.*, 1982]. The Byzantine workers are omniscient (i.e., they have access to the whole dataset and are able to access other clients' updates). We assume data distribution among clients to be non-IID. We are interested in studying the Byzantine Resilience of Distributed Synchronous SGD.

1.2 Outline

We structure the paper as follows: In section 2 we present an overview of key relevant literature. We then introduce a formulation of our approach in section 3. In section 4 we present theoretical analysis of the LS approach. The final two sections respectively present the experimental evaluation and concluding remarks.

2 Related Work

In this section relevant works are highlighted. We first present some of the standard aggregation rules used in the IID setting then in the non-IID setting. Followed by Byzantine attacks on SGD. Finally, we present some of the works that deploy linear scalarization to balance trade-offs between the objectives.

IID Defenses: **Krum** ([Blanchard *et al.*, 2017]) is an aggregation that filters the gradient submissions based on a trust score s_i computed based on the euclidean distance between each client's update and set of its $n - f$ closest neighbors. Trimmed Mean β**-TrMean** [Yin *et al.*, 2018] averages the vectors submissions after discarding a proportion β of the highest and smallest values along each coordinate. **Bulyan** [El Mhamdi *et al.*, 2018] is a meta gradient aggregation rule that enhances existing aggregations against adversaries exploiting the high dimensionality of the model. [Alistarh *et al.*, 2018] uses the Mean around the Median alongside an iterative scheme that allows the detection of Byzantine workers. The works of [El-Mhamdi *et al.*, 2021b; Karimireddy *et al.*, 2021; Farhadkhani *et al.*, 2022] illustrate the role of momentum in strengthening the aggregations capacity of withstanding Byzantine attacks. Although theses aggregations have proven to perform well in the IID case. Their performance significantly degrades in the non-IID set up. Which motivated the following.

Non-IID Defenses: [Li *et al.*, 2019] RSA uses l_1 regularization term to penalize the deviation between local iterates and server iterate. However, RSA is unable to defend against ([Baruch *et al.*, 2019; Xie *et al.*, 2020]). When the Parameter Server has access to a clean validation dataset, the task of detecting Byzantine clients becomes more manageable [Cao *et al.*, 2021]. The work in [Karimireddy *et al.*, 2021] highlights the weakness of standard Byzantine defenses in the non-IID setting and proposes the use of Centered Clipping and momentum. [Gorbunov *et al.*, 2023] uses variance reduction to counter the impact of heterogeneity in defending against Byzantine attacks. [Karimireddy *et al.*, 2022] proposes a bucketing technique that reduces inter-client variance by averaging clients updates within buckets. However when using Bucketing the number of Byzantine workers that can be tolerated decreases to 1/s. As such when using $s = 2$ (as recommended in the paper), the number of Byzantine workers that can be tolerated drops by a half. Similarly, [Allouah *et al.*, 2023] mixes users gradient updates before applying an aggregation rule. It does so by averaging each gradient update with its closest $n - f$ gradient vectors. However, finding the closest neighbors of each gradient vector results in a complexity of $\mathcal{O}(n^2.d)$.

Byzantine Attacks: The Byzantine problem([Lamport *et al.*, 1982]) describes any faulty behaviour that occurs in a distributed system, ranging from machines outputting random values due to hardware or software failures to carefully crafted updates targeting specific algorithms. We focus on poisoning attacks targeting distributed GD. A little is enough **ALIE** ([Baruch *et al.*, 2019]) is a non-omniscient attack wherein the adversary leverages the IID assumption to estimate the mean μ and standard deviation σ of honest gradients in order to compute a malicious gradient vector. **IPM** ([Xie *et al.*, 2020]) operates by injecting gradient updates in the opposite direction of the descent of the gradient in order to disrupt the convergence. **Bit Flip (Sign Flip)** simulates hardware failures in clients. A Byzantine client experiences an internal failure, causing it to flip the sign of the gradient. In **Min-Max** and **Min-Sum** ([Shejwalkar and Houmansadr, 2021]) the adversary computes the malicious update by maximally perturbing the benign reference aggregate in the malicious direction, while also evading detection by robust aggregation algorithms. **Mimic** ([Karimireddy *et al.*, 2022]) is an attack that is designed to take advantage of the non-IID setting by over-representing a worker i^* with least significant update.

Linear Scalarization: is a technique used in Multi-Objective Optimization (MOO) to balance conflicting objectives. The formulation of leaning algorithms as multi-objective problems is indeed a more natural way of addressing them. In ([Sener and Koltun, 2018]) Multi-Task Learning (MTL) is modelled as a multi-objective problem, [Lin *et al.*, 2019] propose Pareto MTL, to solve Multi-Task Learning as MOO. [Mohri *et al.*, 2019] propose a reformulation of Federated Learning that instates fairness among users by optimizing users' objectives for the worst trade-off vector. However, we are not aware of any work addressing the Byzantine Learning problem as MOO. Although there is a line of work that deploys weights or trust scores to penalize the Byzantine ([Jayanth Regatti, 2022; Fu *et al.*, 2019; Li *et al.*, 2019]). However, in the aforementioned works the trade-off vector is used for regularization and not as a trade-off vector.

3 Algorithm

The proposed framework serves as an enhancements to existing robust defenses. We cover four aggregations: Krum, RFA, CM and trMean (A detailed definition of each aggregation can be found in the Appendix). However our approach can easily be applied to any aggregation rule. In the following, we explicitly define the LS procedure.

3.1 Byzantine LS Approach

Given a Robust Aggregation rule $\mathcal{F}$ and $g_1, ..., g_n \in \mathcal{R}^d$ gradient vectors, among which $f < \frac{n}{2}$ are Byzantine. $\mathcal{F}$ generally operates as follows: i) Use some robust approach (statistics, trust scores ...) to split users into two subsets: honest

$\mathcal{G}_t$ and suspected malicious $\mathcal{G}_b$. ii) Return the aggregation of $\mathcal{G}_t$ and fully drop $\mathcal{G}_b$. Following that, the main idea behind $\mathcal{F} \circ LS$ is, after applying a robust aggregation $\mathcal{F}$ rather than dropping the gradients suspected by the aggregation, we re-introduce them with a penalty that is proportional to the degree of derail from the output of the aggregation.

Let $C_{R_i} = \|g_i - g_{ref}\|$ denote a criterion that measures the degree of honesty of a client i by computing the distance between g_i and a reference to the true gradient g_{ref}.

However, as it is easier to access to robust approximation of the gradient computed by $\mathcal{F}$. As such we define $C_{R_i} = \|g_i - g_{\mathcal{F}}\|$ (i.e., how far is g_i from the output of the robust aggregation used $\mathcal{F}$).

The LS method leverages the standard aggregations to define a trade-off vector λ between clients' update vectors such that the more suspected a user is to be Byzantine, the smaller its weight is. That is, our approach keeps all users' gradient vectors and the λ_i serves as a penalty: As such, for i in the subset of discarded gradients $\mathcal{G}_b$, the penalty associated with the re-introduction of g_i is computed as follows $\lambda_i = \alpha_b \min(1, \frac{1}{C_{R_i}})$ where $\alpha_b \in [0, 1]$ (the trade-off vector λ is normalized such that $\sum_i \lambda_i = 1$). To preserve robustness we associate with the set of trusted gradients $\mathcal{G}_t$ (i.e., the output of the aggregation) the weight $\lambda_t = \alpha_t > 0$.

α_b and α_t are hyper-parameters that defines the trade-off between inclusion and robustness and quantifies the confidence in the C_{R_i}. We add these parameters to offer more flexibility in adjusting the trade-off.

We associate with each dropped user i a scalar λ_i namely a trade-off that is defined such that $\lambda_i \propto \frac{1}{C_{R_i}}$. In the following we summarize the main LS steps:

1. Use $\mathcal{F}(.)$ to split users into two subsets potential honest (constitute the gradients selected as output by the aggregation) $\mathcal{G}_t$ and suspected malicious $\mathcal{G}_b$ dropped by the aggregation. For $\mathcal{F}$ that does not split users into two subsets (i.e., does not pick full gradient vectors), we define $\mathcal{G}_t$ contains a single vector (the blended output of the aggregation).
2. Return the weighted sum of all users gradients $\mathcal{F} \circ LS = \sum_i \lambda_i g_i$.

Thereafter, $\mathcal{F} \circ LS$ can be considered as a generalization to the standard $\mathcal{F}(.)$, that pick $\lambda_i = 0$ for the suspected Byzantine and associates uniform weights with the honest clients. Additionally, this formulation alleviates the Robustness/Inclusion trade-off. Algorithm 1 gives the pseudo code of our approach.

Algorithm 1 $\mathcal{F} \circ LS$

Input: $g_1, ..., g_n \in \mathcal{R}^d$, two hyper parameters $\alpha_t > 0, 0 \leq \alpha_b \leq 1$ and $\mathcal{F}$,
Use $\mathcal{F}$ **to determine** $\mathcal{G}_t$ **and** $\mathcal{G}_b$
for $g_i \in \mathcal{G}_t$ **do**
  $\lambda_i = \alpha_t$
end for
for $g_i \in \mathcal{G}_b$ **do**
  $\lambda_i = \alpha_b \min\left(1, \frac{1}{\|g_i - g_{\mathcal{F}}\|}\right)$
end for
Normalise λ
Return $\mathcal{F} \circ LS(g_1, .., g_n) = \sum_{i=1}^{|\lambda|} \lambda_i g_i$

4 Analysis of the LS Approach

In this section we present the theoretical analysis of the $\mathcal{F} \circ LS$. We first start by presenting standard assumptions. Followed by the the definition of robustness and resilience introduced in [Allouah *et al.*, 2023].

- **A1. L-smoothness** We assume the loss function associated with each client $\mathcal{L}_i$ to be L-smooth; for all $\theta, \theta' \in \mathbb{R}^d$:

$$\|\nabla \mathcal{L}_i(\theta) - \nabla \mathcal{L}_i(\theta')\| \leq L\|\theta - \theta'\|$$

- **A2. Bounded Heterogeneity** There exists a real value G such that for all $\theta \in \mathbb{R}^d$,

$$\frac{1}{|S|} \sum_{i \in S} \|\nabla \mathcal{L}_i(\theta) - \nabla \mathcal{L}_S(\theta)\|^2 \leq G^2$$

Definition 1 ((f, κ) Robustness)**.** *Given $g_1, ..., g_n \in \mathbb{R}^d$, $f < \frac{n}{2}$ and any subset $S \subset [n]$ where $|S| = n - f$, an aggregation rule $\mathcal{F}$ is said to be (f, κ) robust if the following holds:*

$$\|\mathcal{F}(g_1, ..., g_n) - \bar{g}_S\|^2 \leq \frac{\kappa}{S} \sum_{i \in S} \|g_i - \bar{g}_S\|^2,$$

where κ is the coefficient of robustness and $\bar{g}_S = \frac{1}{|S|} \sum_{i \in S} g_i$.

Definition 2 ((f, ϵ) Resilience)**.** *Distributed GD is said to be (f, ϵ) resilient if for n clients among which $f < \frac{n}{2}$ are Byzantine, the learning algorithm finds an ϵ-stationary solution $\hat{\theta}$ for the average loss of honest clients:*

$$\|\nabla \mathcal{L}_S(\hat{\theta})\|^2 \leq \epsilon, \text{ where } \nabla \mathcal{L}_S(\theta) = \frac{1}{|S|} \sum_{i \in S} \nabla \mathcal{L}_i(\theta).$$

The previous two definitions are the basis of our theoretical analysis. In the following we study the Robustness of the LS approach then we show that distributed GD with $\mathcal{F} \circ LS$ is (f, ϵ) resilient.

A key concept in our algorithm is the definition of the trade-off vector. This is crucial as the trade-off (penalty) associated with users' gradient updates reflects the impact of the user's gradient in the update of the model. We introduce an upper bound on the inclusion penalty associated with the re-introduction of a user in the training process.

Lemma 1. ***Upper Bound on Inclusion Penalties***

*Let Assumption **A2** hold and $\mathcal{F}$ an aggregation that satisfies the (f, κ) Robustness definition. Then, following the $\mathcal{F} \circ LS$ approach to define inclusion penalties λ.*

For each $i \in \mathcal{G}_b$, we have:

$$\lambda_i \leq \alpha_b \min\left(1, \frac{1}{\max\left(0, \|g_i - \bar{g}_{\mathcal{S}}\| - \sqrt{\kappa G^2}\right)}\right)$$

Note that when the deviation from the average of honest clients gradients ($\|g_i - \bar{g}_{\mathcal{S}}\|$) introduced by user i is very large the bound reads

$$\lambda_i \leq \frac{\alpha_b}{\|g_i - \bar{g}_{\mathcal{S}}\| - \sqrt{\kappa G^2}},$$

which shows that the larger the deviation from the average of honest clients gradients ($\|g_i - \bar{g}_{\mathcal{S}}\|$) introduced by user i, the smaller is the contribution of user i to the gradient update. Consequently, the importance of outliers contribution degrades as their perturbation grows.

Lemma 2 (Robustness)**.** *Let Assumption **A2** hold and $\mathcal{F}$ an aggregation that satisfies the (f, κ) Robustness definition. Then, following the $\mathcal{F} \circ LS$ approach the output of the aggregation satisfies the following:*

$$\|g_{\mathcal{F}\circ LS} - \bar{g}_S\|^2 \leq \kappa' G^2 + \sigma_I, \tag{2}$$

where $\kappa' = 2\kappa$ and $\sigma_I = \frac{2\alpha_b^2|\mathcal{G}_b|^2}{\alpha_t^2|\mathcal{G}_t|^2}\left(1 + \sqrt{\kappa G^2}\right)^2$.

We note that for the particular case where $\alpha_b = 0$, σ_I will be equal to zero, i.e. the case where we we nullify the weights of the suspicious users by setting them to zero, we recover similar bound of robustness known in the literature for the standard robust aggregation. The extra term σ_I in the equation (2) emerges due to the integration of users under suspicion. However, this inclusion is meticulously managed to ensure it has no impact on the robustness.

Theorem 1. *Let assumptions A1 and A2 hold, consider distributed SGD with $T \geq 1$ and a learning rate $\alpha = \frac{1}{L}$, if $\mathcal{F}$ is (f, κ) robust then*

$$\|\nabla \mathcal{L}_S(\hat{\theta})\|^2 \leq \kappa' G^2 + \sigma_I + \frac{2L(\mathcal{L}_S(\theta_0) - \mathcal{L}^*)}{T}$$

where $\hat{\theta} = \arg\min_{1\leq t\leq T} \|\nabla \mathcal{L}_S(\theta_t)\|$ and $\mathcal{L}^* := \inf_{\theta\in\mathbb{R}^d} \mathcal{L}_S(\theta)$

The previous theorem establishes the convergence bound for our method which matches the same bound of the standard robust aggregation convergence in the literature, up to the additive constant σ_I. This additional constant, proportional to α_b^2/α_t^2, arises due to the inclusion of potential Byzantine clients in the learning process. It can be managed by adjusting the extra hyper parameters α_b and α_t. Specifically, when α_b is small enough compared to α_t, the influence of σ_I becomes negligible. Empirical observations, as illustrated in Figure 1, demonstrate good performance even when α_b is equal to α_t.

5 Empirical Evaluation

In this section we evaluate the introduced LS variants both on IID and non-IID data when a proportion of the clients is Byzantine. We give a detailed description of the experimental set up: the used data sets, the techniques deployed for non-IID data partitioning, the models' architectures alongside all the relevant hyper-parameters. Finally, we present Top-1 Test Accuracy as an evaluation metric.

5.1 Experimental Set Up

Data Splits We study the following datasets: MNIST [LeCun *et al.*, 1998], FMNIST [Xiao *et al.*, 2017], SVHN [Netzer *et al.*, 2011] and CIFAR10 [Krizhevsky and Hinton, 2009]. We experiment with IID and non-IID data splits. The later is generated by creating label unbalances between clients. We use Latent Dirichlet Sampling which uses the Dirichlet distribution $Dir(\beta)$, such that each party gets assigned a proportion of samples of a given label, where β a hyper-parameter that allows to introduce varying levels of skewness (the smaller the β the more skewed is the resulting split). Following existing works [Li *et al.*, 2022; Tang *et al.*, 2022; Allouah *et al.*, 2023], each dataset is partitioned with varying non-IID degrees using $\beta = 0.1$ for high non-IID and $\beta = 1.0$ for low non-IID splits. Moreover, to verify the impact of LS, we conduct experiments using another type of non-IID partition: C_k-classes partition referred to as pathological non-IID [McMahan *et al.*, 2017], we however set $K = 3$ i.e. each client only has access to labels from 3 classes (Results on MNIST for $K = 3$ can be found in the Appendix).

Learning Model For MNIST [LeCun *et al.*, 1998] and FMNIST [Xiao *et al.*, 2017] we use LeNet5 [LeCun *et al.*, 1998]. For SVHN [Netzer *et al.*, 2011] and CIFAR10 [Krizhevsky and Hinton, 2009] we use AlexNet [Krizhevsky *et al.*, 2017]. Finally we use VGG11 for [Simonyan and Zisserman, 2015] CIFAR100 [Krizhevsky and Hinton, 2009]. (Due to space constraints experiments on SVHN, FMNIST and CIFAR100 can be found in the Appendix).

5.2 Experimental Results

In this section we present the experimental results. Our work investigates the impact of granting inclusion to outliers on the performance of deep learning models. Especially when the outliers maybe malicious. To the best of our knowledge no work so far investigated the interplay between inclusion and robustness. As such we are primarily focused on how the performance of the standard robust aggregations changes as one grants Inclusion of outliers. In the non-IID case we also compare $\mathcal{F} \circ LS$ to variants of $\mathcal{F}$ that are enhanced to perform well on non-IID data mainly the Bucketing technique [Karimireddy *et al.*, 2022] and NNM [Allouah *et al.*, 2023].

In the following, we test the LS variants of the following aggregations: Krum [Blanchard *et al.*, 2017], RFA [Pillutla *et al.*, 2022], CM and trMean [Yin *et al.*, 2018]. We report $Top-1$ Test Accuracy averaged over 3 runs for each defense.

In fig. 1 We investigate the impact of the choice of the hyper-parameters α_b in the performance of our approach. Clearly, $\alpha_t = \alpha_b = 1$ is the best choice overall as this strategy keeps the output of the standard aggregation and penalizes the dropped contributions based on their deviation from the output of $\mathcal{F}$ with no further penalty. Further penalize the inclusion term $\alpha_b < \alpha_t$ the performance slightly deteriorates but the model still converges .

In the IID case fig. 2 shows that the LS variants are as efficient as the standard robust aggregations they deploy. This shows that granting inclusion in the IID case does not hinder the performance of the model.

	IPM	ALIE	BF	Mimic	MinMax	MinSum	Worst Case
CMLS	85.0	**94.4**	**93.1**	**94.5**	**85.2**	89.0	85.0
CM	54.2	27.3	35.3	32.7	14.9	12.8	12.8
CM NNM	**89.8**	70.9	78.3	90.5	83.6	**92.1**	70.9
CM buck	77.9	79.8	76.2	86.6	27.2	14.0	14.0
mKLS	93.7	90.8	95.3	**95.1**	**86.8**	**94.1**	86.8
mKrum	88.3	93.1	88.1	90.3	85.9	93.2	85.9
mKrum NNM	88.4	51.5	76.7	90.4	84.6	92.9	51.5
mKrum buck	**93.8**	**94.3**	**95.6**	93.0	86.1	93.5	86.1
trLS	**85.3**	**94.5**	**93.6**	**94.5**	85.0	88.4	85.0
trMean	48.0	32.2	45.1	64.1	22.3	15.2	15.2
trMean NNM	85.2	52.2	82.8	89.7	**91.1**	**90.5**	52.2
trMean buck	83.3	91.7	86.6	86.0	20.0	31.3	20.0
RFA LS	74.6	**94.5**	93.7	93.0	78.8	78.1	78.1
RFA	67.0	62.0	94.0	92.0	79.6	74.4	62.0
RFA NNM	87.0	64.6	74.5	91.1	82.3	90.8	64.6
RFA buck	**95.2**	92.6	**94.9**	**96.7**	**87.0**	**94.1**	87.0

Table 1: MNIST on high non-IID $\beta = 0.1$ under $\delta = 20\%$. We report Maximum Top-1 test Accuracy (%) across T = 600 iterations for six Byzantine attacks. In each of the four horizontal blocks and under each attack, we highlight in bold the best accuracy under each attack. For every method, we also show the worst-case accuracy across attacks, and highlight in green the best one in each block.

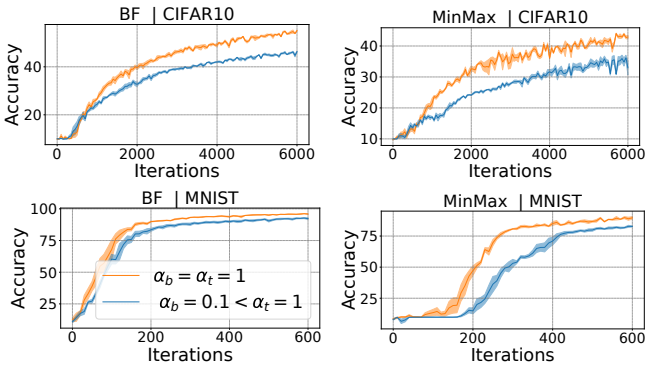

Figure 1: **CMLS** Top-1 Test Accuracy of on **MNIST** and **CIFAR10** non-IID $\beta = 1.0$ **BF** and **MinMax** attacks. under a Byzantine ratio of Byzantine workers $\delta = 30\%$.

In the high non-IID data regime table 1 shows that the LS variants offer a significant boost to the standard aggregations under Byzantine attacks. The improvement is comparable or outperforming techniques used to relief the non-IID impact such as NNM and Bucketing. This behaviour can be explained by the fact that robust aggregations may discard benign outliers and using $\mathcal{F} \circ LS$ for Inclusion re-introduces the benign clients updates offering a boost in the model's performance in the non-IID case. The same result holds for CIFAR10 fig. 3. The figures show that if an aggregation is performing poorly under an attack, the LS variant enhances its performance (e.g., CM and trMean). Whereas, for aggregations that are already achieving good accuracy (e.g., RFA) the LS variant is comparable. It follows that, granting inclusion does not have a negative impact on the final acccuracy achieved even in the presence of Byzantine workers.

In a settings wherein the (PS) has access to a validation dataset, it is simpler to define an accurate trade-off vector. Using a clean validation dataset guarantees access to a non perturbed approximation of the gradient of the objective function. Thereafter simplifying the task of finding and appropriately penalizing outliers.

We investigate the efficiency of this variant against state of the art Byzantine attacks under varying ratios of Byzantine clients fig. 4: our result on CIFAR10 shows that using a reference gradient computed at the server to establish the trade-off vector performs well in practice. Our method even surpasses state of the art defense **FL trust** ([Cao *et al.*, 2021]) that also operates under the same assumption. Although, this method of defining the penalties is more reliable, it however assumes access to a validation dataset which can be challenging to acquire for certain applications thus limiting its practicality. Nonetheless, these results further validate the idea that accurate penalizing of outliers is a suitable approach to guarantee both some degree of inclusion while also maintaining good accuracy.

6 Conclusion

The main goal of this work is to provide an answer the following question: Is it possible to grant Byzantine-robustness and inclusion of outliers without compromising the model's accuracy? In other words, can we get away with tolerating the outliers including those that may be Byzantine ?

We propose a Linear Scalarization approach that takes advantage of the existing Byzantine defenses to define a trade-off between the clients' objectives. This formulation allows to balance users' contribution by giving greater weight to honest workers' gradients while minimizing the impact of suspected

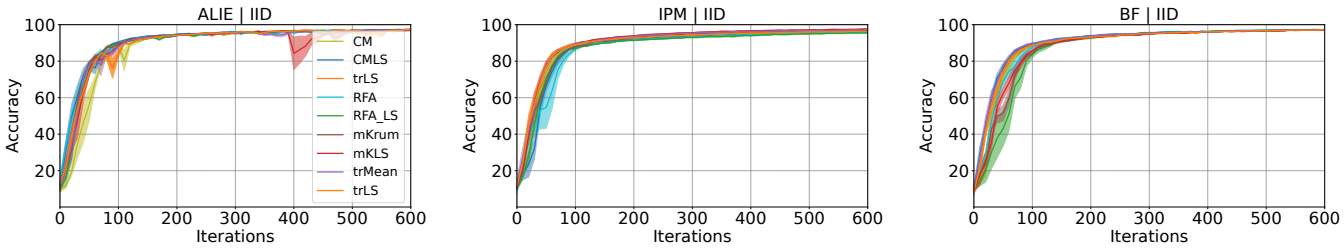

Figure 2: MNIST $Top-1$ Test Accuracy for all defenses on IID data split under the following attacks **ALIE**,**IPM** and **BF** Bit Flip for a ratio $\delta = 20\%$. This scenario is implemented to test the impact inclusion on model accuracy. The LS variants achieve final accuracy identical to the robust aggregation their based-on.

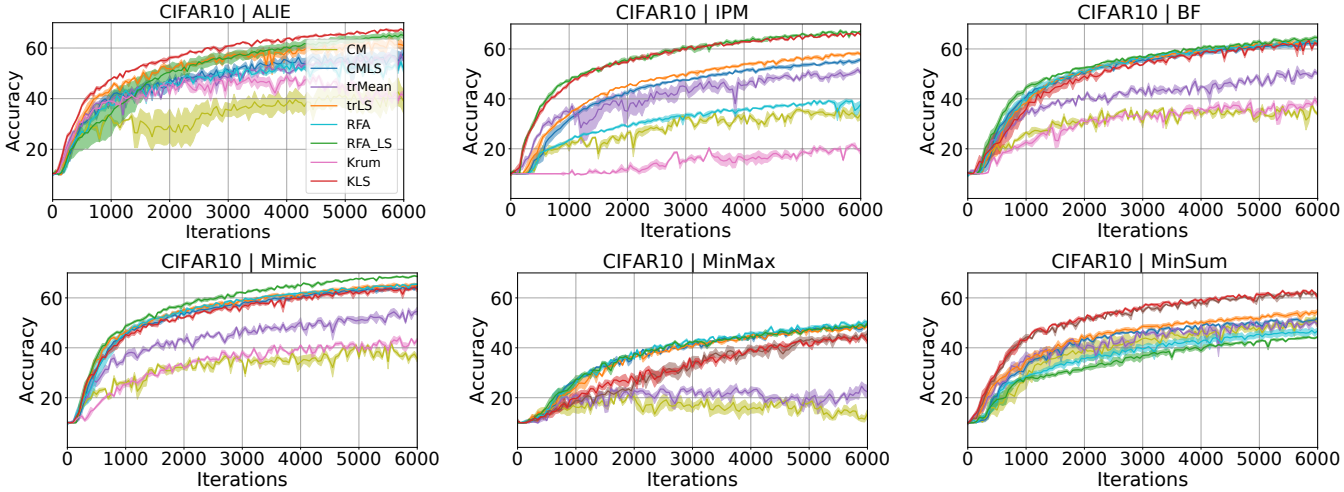

Figure 3: CIFAR10 $Top-1$ Test Accuracy for all studied aggregations on non-IID data split $\beta \sim 1.0$ under $\delta = 20\%$ Byzantine performing **ALIE**, **IPM**, **BF**,**Mimic**, **MinMax** and **MinSum**.

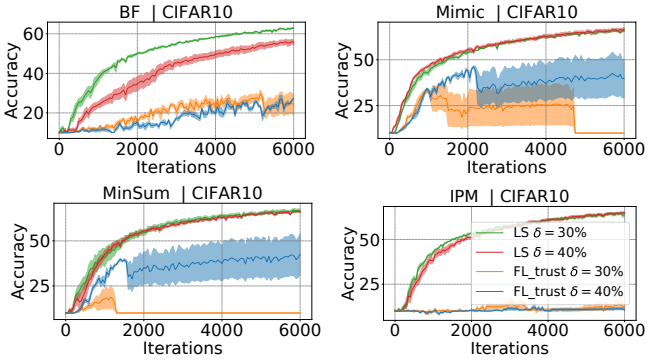

Figure 4: CIFAR10 non-IID $\beta = 1.0$ under **BF**, **Mimic**, **MinSum** and **IPM**. Each figure shows Top-1 Accuracy of LS and FL trust under two varying degrees of Byzantine clients $\delta = 20\%$ and $\delta = 30\%$ for each of the mentioned attacks.

workers. Although, LS is primarily a strategy for granting inclusion and does not address the essence of heterogeneity, our experimental results show that LS can however withstand data heterogeneity and alleviate it to some degree. For instance, in the event that a benign client is treated as an outlier due to the varying degrees of heterogeneity, the client may be excluded from most rounds of training. LS acts as a corrective measure in the sense that for the underlying aggregation: if $\mathcal{F}$ on its vanilla format fails in a given set up, this would entail that the aggregated updates contain those of Byzantine clients while a subset of honest are discarded. In that case, re-introducing the discarded updates would help improve the overall accuracy. This is exemplified by the performance of the CM and trMean. In the non-IID set-up these aggregations fail to defend against standard Byzantine attacks independently. However, by re-introducing the weighted sum of discarded updates the accuracy is restored to its baseline. Consequently, the incorporation of the penalized clients' updates does not negatively impact the performance of the aggregations. Our approach grants Byzantine robustness while allowing outliers' contributions. Thereafter alleviating the Inclusion/Robustness trade-off posed by standard aggregations.

Our approach can be seen as an extension of the standard defenses , which sets $\lambda_i = 0$ for all the suspected clients. Our theoretical results expand upon the existing guarantees established for the standard $\mathcal{F}$, we essentially establish analogous complexity bounds, with an added constant that emerges naturally due to the incorporation of potentially malicious clients into the training process.

References

[Alistarh *et al.*, 2018] Dan Alistarh, Zeyuan Allen-Zhu, and Jerry Li. Byzantine stochastic gradient descent. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[Allouah *et al.*, 2023] Youssef Allouah, Sadegh Farhadkhani, Rachid Guerraoui, Nirupam Gupta, Rafael Pinot, and John Stephan. Fixing by mixing: A recipe for optimal byzantine ml under heterogeneity. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent, editors, *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 1232–1300. PMLR, 25–27 Apr 2023.

[Baruch *et al.*, 2019] Gilad Baruch, Moran Baruch, and Yoav Goldberg. A little is enough: Circumventing defenses for distributed learning. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

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

[Boussetta *et al.*, 2021] Amine Boussetta, El Mahdi El-Mhamdi, Rachid Guerraoui, Alexandre Maurer, and Sébastien Rouault. AKSEL: Fast Byzantine SGD. *Leibniz International Proceedings in Informatics, LIPIcs*, 184(8), 2021.

[Cao *et al.*, 2021] Xiaoyu Cao, Minghong Fang, Jia Liu, and Neil Zhenqiang Gong. Fltrust: Byzantine-robust federated learning via trust bootstrapping, 2021.

[Damaskinos *et al.*, 2018] Georgios Damaskinos, El Mahdi El Mhamdi, Rachid Guerraoui, Rhicheek Patra, and Mahsa Taziki. Asynchronous byzantine machine learning (the case of sgd). *35th International Conference on Machine Learning, ICML 2018*, 3, 2018.

[El Mhamdi *et al.*, 2018] El Mahdi El Mhamdi, Rachid Guerraoui, and Sebastien Rouault. The hidden vulnerability of distributed learning in byzantium. In *35th International Conference on Machine Learning, ICML 2018*, volume 8, 2018.

[El-Mhamdi *et al.*, 2021a] El Mahdi El-Mhamdi, Sadegh Farhadkhani, Rachid Guerraoui, Arsany Guirguis, Lê-Nguyên Hoang, and Sébastien Rouault. Collaborative learning in the jungle (decentralized, byzantine, heterogeneous, asynchronous and nonconvex learning). In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34. Curran Associates, Inc., 2021.

[El-Mhamdi *et al.*, 2021b] El-Mahdi El-Mhamdi, Rachid Guerraoui, and Sébastien Rouault. Distributed Momentum for Byzantine- resilient Stochastic Gradient Descent. *International Conference on Learning Representations*, 2021.

[Farhadkhani *et al.*, 2022] Sadegh Farhadkhani, Rachid Guerraoui, Nirupam Gupta, Rafael Pinot, and John Stephan. Byzantine machine learning made easy by resilient averaging of momentums. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*. PMLR, 17–23 Jul 2022.

[Fu *et al.*, 2019] Shuhao Fu, Chulin Xie, Bo Li, and Qifeng Chen. Attack-resistant federated learning with residual-based reweighting. *CoRR*, abs/1912.11464, 2019.

[Gorbunov *et al.*, 2023] 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*, 2023.

[Jayanth Regatti, 2022] Abhishek Gupta Jayanth Regatti, Hao Chen. Byzantine resilience with reputation scores. In *Annual Allerton Conference on Communication, Control, and Computing*, 2022.

[Karimireddy *et al.*, 2021] Sai Praneeth Karimireddy, Lie He, and Martin Jaggi. Learning from history for byzantine robust optimization. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*. PMLR, 18–24 Jul 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.

[Krizhevsky and Hinton, 2009] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, Toronto, Ontario, 2009.

[Krizhevsky *et al.*, 2017] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90, may 2017.

[Lamport *et al.*, 1982] Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 4(3):382–401, jul 1982.

[LeCun *et al.*, 1998] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[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 *Association for the Advancement of Artificial Intelligence*, 2019.

[Li *et al.*, 2022] Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. Federated learning on non-iid data silos: An experimental study. In *IEEE International Conference on Data Engineering*, 2022.

[Lin *et al.*, 2019] Xi Lin, Hui-Ling Zhen, Zhenhua Li, Qing-Fu Zhang, and Sam Kwong. Pareto multi-task learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[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 Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*. PMLR, 20–22 Apr 2017.

[Mohri *et al.*, 2019] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic federated learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*. PMLR, 09–15 Jun 2019.

[Netzer *et al.*, 2011] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NeurIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011.

[Pillutla *et al.*, 2022] Krishna Pillutla, Sham M. Kakade, and Zaid Harchaoui. Robust aggregation for federated learning. In *IEEE Transactions on Signal Processing*, 2022.

[Sener and Koltun, 2018] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[Shejwalkar and Houmansadr, 2021] Virat Shejwalkar and Amir Houmansadr. Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning. In *Network and Distributed Systems Security (NDSS) Symposium*, 2021.

[Simonyan and Zisserman, 2015] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[Tang *et al.*, 2022] Zhenheng Tang, Yonggang Zhang, Shaohuai Shi, Xin He, Bo Han, and Xiaowen Chu. Virtual homogeneity learning: Defending against data heterogeneity in federated learning. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 21111–21132. PMLR, 17–23 Jul 2022.

[Xiao *et al.*, 2017] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017.

[Xie *et al.*, 2020] Cong Xie, Oluwasanmi Koyejo, and Indranil Gupta. Fall of empires: Breaking byzantine-tolerant sgd by inner product manipulation. In *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, volume 115 of *Proceedings of Machine Learning Research*. PMLR, 22–25 Jul 2020.

[Yin *et al.*, 2018] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*. PMLR, 10–15 Jul 2018.