Byzantine-Robust Learning on Heterogeneous Datasets via Resampling

Lie He[*]
MLO, EPFL
lie.he@epfl.ch

Sai Praneeth Karimireddy[*]
MLO, EPFL
sai.karimireddy@epfl.ch

Martin Jaggi
MLO, EPFL
martin.jaggi@epfl.ch

Abstract

In Byzantine robust distributed optimization, a central server wants to train a machine learning model over data distributed across multiple workers. However, a fraction of these workers may deviate from the prescribed algorithm and send arbitrary messages to the server. While this problem has received significant attention recently, most current defenses assume that the workers have identical data. For realistic cases when the data across workers are heterogeneous (non-iid), we design new attacks which circumvent these defenses leading to significant loss of performance. We then propose a simple resampling scheme that adapts existing robust algorithms to heterogeneous datasets at a negligible computational cost. We theoretically and experimentally validate our approach, showing that combining resampling with existing robust algorithms is effective against challenging attacks.

1 Introduction

Distributed or federated machine learning, where the data is distributed across multiple workers, has become an increasingly important learning paradigm both due to growing sizes of datasets, as well as privacy and security concerns. In such a setting, the workers collaborate to train a single model without transmitting their data directly over the networks (McMahan et al., 2016; Bonawitz et al., 2019; Kairouz et al., 2019). Due to the presence of either actively malicious agents in the network, or simply due to system and network failures, some workers may disobey the protocols and send arbitrary messages; such workers are also known as *Byzantine* workers (Lamport et al., 2019). Byzantine robust optimization algorithms combine the gradients received by all workers using robust aggregation rules, to ensure that the training is not impacted by the malicious workers.

While this problem has received significant recent attention, (Alistarh et al., 2018; Blanchard et al., 2017; Yin et al., 2018a), most of the current approaches assume that the data present on each different worker has identical distribution. In this work, we show that existing Byzantine-robust methods catastrophically fail in the realistic setting when the data is distributed heterogeneously across the workers. We then propose a simple resampling scheme which can be readily combined with existing aggregation rules to allow robust training on heterogeneous data.

Contribution. Concretely, our contributions in this work are

- We show that when the data across workers is heterogeneous, existing robust rules might not converge, even without any Byzantine adversaries.
- We propose two new attacks, normalized gradient and mimic, which take advantage of data heterogeneity and circumvent median and sign-based defenses (Blanchard et al., 2017; Pillutla et al., 2019; Li et al., 2019).

[*]These two authors contributed equally

Presented at SpicyFL Workshop at 34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

- We propose a simple new resampling step which can be used before any existing robust aggregation rule. We instantiate our scheme with KRUM and theoretically prove that the resampling generalizes it to the setting of heterogeneous data.
- Our experiments evaluate the proposed resampling scheme against known and new attacks and show that it drastically improves the performance of 3 existing schemes on realistic heterogeneously distributed datasets.

Setup and notations. We study the general distributed optimization problem

$$\mathcal{L}^\star = \min_{\boldsymbol{x}\in\mathbb{R}^d}\{\mathcal{L}(\boldsymbol{x}) := \tfrac{1}{n}\sum_{i=1}^n \mathcal{L}_i(\boldsymbol{x})\} \tag{1}$$

where $\mathcal{L}_i : \mathbb{R}^d \to \mathbb{R}$ are the individual loss functions distributed among n workers, each having its own (heterogeneous) data distribution $\{\mathcal{D}_i\}_{i=1}^n$. The case of empirical risk minimization with m_i datapoints $\boldsymbol{\xi}_i \sim \mathcal{D}_i$ on worker i is obtained when using $\mathcal{L}_i(\boldsymbol{x}) := \frac{1}{m_i}\sum_{j=1}^{m_i}\mathcal{L}_i(\boldsymbol{x},\ \boldsymbol{\xi}_i^j)$. The (stochastic) gradient computed by a **good** node i with sample j is given as $\boldsymbol{g}_i(\boldsymbol{x}) := \nabla\mathcal{L}_i(\boldsymbol{x},\ \boldsymbol{\xi}_i^j)$ with mean $\boldsymbol{\mu}_i$ and variance σ_i^2. We also assume that the heterogeneity (variance across good workers) is bounded i.e.

$$\mathbb{E}_i\|\nabla\mathcal{L}_i(\boldsymbol{x}) - \nabla\mathcal{L}(\boldsymbol{x})\|^2 \le \bar{\sigma}^2\,, \forall \boldsymbol{x}\,.$$

We write $\boldsymbol{g}_i$ instead of $\boldsymbol{g}_i(\boldsymbol{x}^t)$ when there is no ambiguity. A distributed training step using an aggregation rule is given as

$$\boldsymbol{x}^{t+1} := \boldsymbol{x}^t - \gamma^t \text{Aggr}(\{\boldsymbol{g}_i(\boldsymbol{x}^t) : i \in [n]\}) \tag{2}$$

If the aggregation rule is the arithmetic mean, then (2) recovers standard minibatch SGD.

Byzantine attack model. In each iteration, there is a set **Byz** of at most f Byzantine workers. The remaining workers are **good**, thus follow the described protocol. A Byzantine worker $j \in$**Byz** can deviate from protocol and send an arbitrary vecter to the server. Besides, we also allow that Byzantine workers can collude with each other and know every state of the system. Unlike martingale-based approaches like (Alistarh et al., 2018), we allow the set **Byz** to change over time (Blanchard et al., 2017; Chen et al., 2017; Mhamdi et al., 2018).

2 Attacks against existing aggregation schemes

In this section we show that when the data across the workers is heterogeneous (non-iid), then we can design new attacks which take advantage of the heterogeneity, leading to the failure of existing aggregation schemes. We study three classes of robust aggregation schemes: i) schemes which select a representative worker in each round (e.g. KRUM (Blanchard et al., 2017)), ii) schemes which use normalized means (e.g. RSA (Li et al., 2019)), and iii) those which use the median (e.g. RFA (Pillutla et al., 2019)). We show realistic settings under which each of these classes would fail when faced with heterogeneous data.

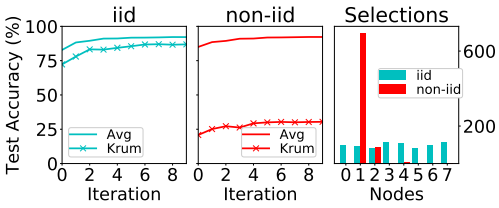
2.1 Failure of representative worker schemes on non-iid data

Algorithms like KRUM select workers who are representative of a majority of the workers, by relying on statistics such as pairwise differences between the various worker updates. Let $(\boldsymbol{g}_1, \ldots, \boldsymbol{g}_n)$ be the gradients by the workers, f of which are Byzantine (e.g. $n \ge 2f+3$ for KRUM). For $i \ne j$, let $i \to j$ denote that $\boldsymbol{g}_j$ belongs to the $n-f-2$ closest vectors to $\boldsymbol{g}_i$. Then KRUM is defined as follows

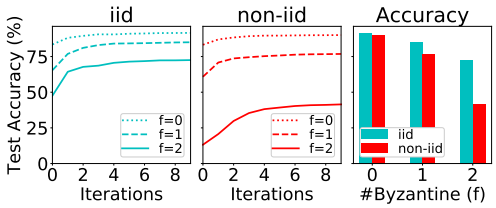
$$\text{KRUM}(\boldsymbol{g}_1, \ldots, \boldsymbol{g}_n) := \arg\min_i \sum_{i\to j} \|\boldsymbol{g}_i - \boldsymbol{g}_j\|^2 \tag{3}$$

However, when the data across the workers is heterogeneous, there is no 'representative' worker. This is because each worker computes their local gradient over vastly different local data. Hence, for convergence it is important to not only select a good (non-Byzantine) worker, but also ensure that each of the good workers is selected with roughly equal frequency. Hence KRUM suffers a significant loss in performance with heterogeneous data, even when there are *no Byzantine workers*.

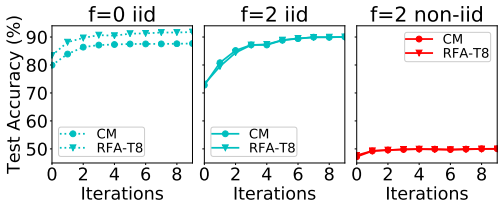
For example, when KRUM is used for iid datasets without adversary ($f = 0$, see left of Figure 1a), the test accuracy is close to simple average and the gap can be filled by MULTI-KRUM (Blanchard et al., 2017). The right plot of Figure 1a also shows that KRUM's selection of gradients is biased towards certain nodes. When KRUM is applied to non-iid datasets (the middle of Figure 1a), KRUM performs poorly even without any attack. This is because KRUM mostly selects gradients from a few nodes whose distribution is closer to others (the right of Figure 1a). This is an example of how robust aggregation rules may fail on realistic non-iid datasets.



(a) Left & middle: Comparing arithmetic mean with KRUM on iid and non-iid datasets, **without** any Byzantine workers. Right: Histogram of selected gradients.



(b) Comparing normalized mean (RFA with T=1) under the **normalized mean attack** with $f = 0, 1, 2$ attackers.



(c) Comparing coordinate-wise median (CM) and geometric median (RFA with T=8) under the **mimic2 attack** on iid and non-iid datasets.

Figure 1: Failures of existing aggregation rules on the non-iid MNIST dataset. In all experiments, there are 8 good and f Byzantine workers.

2.2 Attacks on normalized aggregation schemes

Instead of simply averaging the gradients, some methods first normalize them and then average. This limits the influence of the Byzantine workers since they cannot output extremely large gradients, and hence is more robust. For example RFA (Pillutla et al., 2019) with T=1 uses following aggregation rule:

$$\mathrm{NM}(\boldsymbol{g}_1, \ldots, \boldsymbol{g}_n) = \sum_{i=1}^{n} \frac{\boldsymbol{g}_i}{\|\boldsymbol{g}_i\|_2} \tag{4}$$

Other methods such as RSA (Li et al., 2019) or signum (Bernstein et al., 2018) normalize entries coordinate-wise before taking a majority vote i.e. update the server model $\boldsymbol{x}_0$ on server using local model $\boldsymbol{x}_i$ from node i (not gradient) using

$$\mathrm{RSA}(\boldsymbol{x}_0; \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) := \nabla f_0(\boldsymbol{x}_0) + \lambda \sum_{i=1}^{n} \mathrm{sign}(\boldsymbol{x}_0 - \boldsymbol{x}_i) \tag{5}$$

where f_0 is a strongly convex penalty term and $\lambda > 0$ is a relaxation parameter.

However, a Byzantine worker can still craft an "omniscient" attack to foil robust aggregations, using an approach similar to the negative sum for the arithmetic mean (Blanchard et al., 2017; Li et al., 2019):

$$\boldsymbol{v} := -\sum_{i \in \mathbf{good}} \frac{\boldsymbol{g}_i}{\|\boldsymbol{g}_i\|_2} \tag{6}$$

On the right side of Figure 1b, we can see that this attack lowers the accuracy of RFA-T1 significantly, as the number of Byzantine workers increases. Comparing to its iid counterpart, the normalized mean attack is even more impactful in the non-iid setting.

2.3 Attacks on median-based schemes

Geometric median and its variants are popular in robust learning research (Blanchard et al., 2017; Chen et al., 2017; Pillutla et al., 2019; Yin et al., 2018a; Mhamdi et al., 2018). Given gradients $\{\boldsymbol{g}_1, \ldots, \boldsymbol{g}_n\}$, we use the estimator

$$\mathrm{GM}(\boldsymbol{g}_1, \ldots, \boldsymbol{g}_n) := \mathrm{argmin}_{\boldsymbol{v}} \sum_{i=1}^{n} \|\boldsymbol{v} - \boldsymbol{g}_i\| \,. \tag{7}$$

If the vectors $\{\boldsymbol{g}_1, \ldots, \boldsymbol{g}_n\}$ are drawn independently from the same distribution, intuitively most of them would concentrate around their mean. Then, even if there are some Byzantine outputs, the median would ignore those as outliers and output a 'central' point close to the mean.

However, when $\{\boldsymbol{g}_1, \ldots, \boldsymbol{g}_n\}$ are gradients over heterogeneous data, they may be vastly different from each other and do not concentrate around the mean. In such a scenario, the median such as (7) can be even less robust than simply taking the mean. Suppose that worker 0 is Byzantine

Algorithm 1 Robust Learning with Resampling

Setup: n workers, f of which are Byzantine; resampling T times, each time samples s gradients. A robust learning algorithm AGGR on iid datasets; γ is the learning rate.

Workers:

1. Each good worker i randomly samples a datapoint j and computes a stochastic gradient $\boldsymbol{g}_i := \nabla F_i(\boldsymbol{x}, \boldsymbol{\xi}_i^j)$ where $\boldsymbol{\xi}_i^j \sim \mathcal{D}_i$; each Byzantine worker i sends arbitrary vector $\boldsymbol{g}_i$.
2. Send $\boldsymbol{g}_i$ to server.

Servers:

1. Receive $\{\boldsymbol{g}_i\}_{i=1}^n$ from all workers.
2. $\mathcal{S}, \mathcal{I}_S$ = Resampling($\{\boldsymbol{g}_i : i \in [n]\}$, f, T, s); See Algorithm 2 .
3. Compute $\boldsymbol{x}' := \boldsymbol{x} - \gamma \text{AGGR}(\mathcal{S})$;
4. Broadcast $\boldsymbol{x}'$ to all workers.

Algorithm 2 Resampling with s-replacement

Input: $\{\boldsymbol{g}_i : i \in [n]\}$, $T := n$, s, $\{c[i] := 0 : i \in [n]\}$
for $t := 1, \ldots, T$ **do**
  for $i := 1, \ldots, s$ **do**
    while Select $j_i \sim \mathbf{Uniform}([n])$ **do**
      if $c[j_i] < s$ **then**
        $c[j_i] \mathrel{+}= 1$
        If $c[j_i] == s$ Break;
  Compute average $\bar{\boldsymbol{g}}_t := \frac{1}{s}\sum_{i=1}^s \boldsymbol{g}_{j_i}$
Return $\{\bar{\boldsymbol{g}}_t : t \in [T]\}$, $\{j_i^t : t \in [T], i \in [s]\}$

and the remaining workers $\{1, \ldots, 2n\}$ are good, with a total of $2n + 1$ workers. Now suppose that $g_i = (-1)^i$ for all the workers, with half the good workers having -1 and the other half $+1$. This means that the true mean is 0, however, the median estimator (7) will output 1.

Mimic attack. This motivates our *mimic* attack in which all Byzantine workers collude and agree to always send gradients from the same worker. We define a specialized attack, called *mimic2*, where half of the good workers have same datasets and send $\boldsymbol{g}_1$ while the rest good workers send $\boldsymbol{g}_2$; then all Byzantine workers send $\boldsymbol{v} = \boldsymbol{g}_1$ such that the geometric median of the gradients received by the server is always $\boldsymbol{g}_1$. Therefore, this attack breaks geometric-median-based robust aggregation rules, by leading them to wrong solutions. The left plot of Figure 1c shows the impact of the mimic2 attack. Test accuracies of CM and RFA both drop drastically to around 50%.

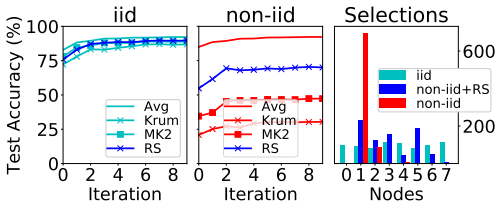
3 Robust aggregation on non-iid data

In Section 2 we have demonstrated how existing robust aggregation rules can fail in realistic non-iid scenarios, with and without attackers (Sections 2.2 and 2.3 and Section 2.1 respectively). To overcome this problem, we propose a simple new resampling-based aggregation rule for training, shown in Algorithm 1. More specifically, we choose *s-resampling without replacement* in Algorithm 2 where each gradient can be sampled at most s times. The key property of our rule is that after resampling, the resulting set of averaged gradients $\{\bar{\boldsymbol{g}}_t : t \in [T]\}$ are much more homogeneous (lower variance). Then these averaged gradients are fed to existing Byzantine robust aggregation schemes, such as KRUM, see Appendix B. Given an existing aggregation rule AGGR, we denote by AGGR $\circ$ Resampling the resulting new robust aggregation rule for non-iid input gradients.

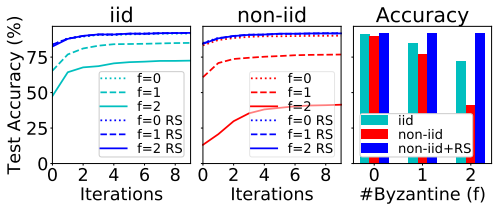
In the following proposition, we list the desired properties of Algorithm 2

Proposition I. *Given a population $\{\boldsymbol{g}_i : i \in [n]\} \subset \mathbb{R}^d$ of mean $\boldsymbol{\mu} := \frac{1}{n}\sum_{i=1}^n \boldsymbol{g}_i$ and variance $\sigma^2 := \frac{1}{n}\sum_{i=1}^n \|\boldsymbol{g}_i - \boldsymbol{\mu}\|^2$, let $\{\bar{\boldsymbol{g}}_t : t \in [T]\}$ be the output of Algorithm 2 on $\{\boldsymbol{g}_i : i \in [n]\}$. Then*

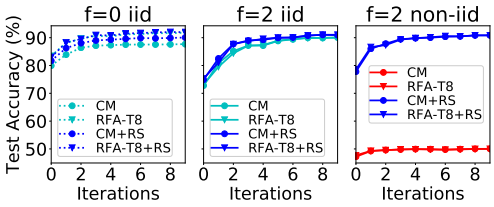
- *If there are no Byzantine workers, then $\{\bar{\boldsymbol{g}}_t : t \in [T]\}$ are identically distributed*
$$\mathbb{E}[\bar{\boldsymbol{g}}_t] = \boldsymbol{\mu}, \;\; var(\bar{\boldsymbol{g}}_t) = \tfrac{n-1}{sn-1}\sigma^2 \qquad \forall\, t \in [T] \tag{8}$$
- *If f of the n inputs are Byzantine, then at least $T - sf$ gradients in $\{\bar{\boldsymbol{g}}_t : t \in [T]\}$ are good; that is, a good $\bar{\boldsymbol{g}}_t$ is the average of gradients $\{\boldsymbol{g}_{j_i^t} : i \in [s]\} \subset$**good**$\subset [n]$. Then such good $\{\bar{\boldsymbol{g}}_t\}$ are identically distributed with*



(a) Left & middle: Comparing arithmetic mean with KRUM on iid and non-iid datasets, **without** any Byzantine workers. Right: Histogram of selected gradients.



(b) Comparing normalized mean (RFA with T=1) under the **normalized mean attack** with $f = 0, 1, 2$ attackers.



(c) Comparing coordinate-wise median (CM) and geometric median (RFA with T=8) under **mimic2 attack** on iid and non-iid datasets.

Figure 2: Combining resampling with existing aggregation rules on non-iid MNIST dataset. In all experiments, there are 8 good and f Byzantine workers. For each aggregation we resample and average s gradients for $T = n$ times.

$$\mathbb{E}[\bar{\boldsymbol{g}}_t] = \tilde{\boldsymbol{\mu}}, \quad var(\bar{\boldsymbol{g}}_t) = \tfrac{n-1}{sn-1}\tilde{\sigma}^2 \tag{9}$$

$$\textit{where } \tilde{\boldsymbol{\mu}} := \tfrac{1}{|\boldsymbol{good}|}\textstyle\sum_{i\in\boldsymbol{good}} \boldsymbol{g}_i, \textit{ and } \tilde{\sigma}^2 := \tfrac{1}{|\boldsymbol{good}|}\textstyle\sum_{i\in\boldsymbol{good}} \|\boldsymbol{g}_i - \mathbb{E}[\bar{\boldsymbol{g}}_t]\|^2.$$

Note that the $\{\bar{\boldsymbol{g}}_t : t \in [T]\}$ are identically distributed but not independent. This does not directly fit into the original assumptions of Byzantine robust algorithms like KRUM and hence the robustness has to be reproved for our more general setting.

4 Experiments

In this section, we demonstrate the effect of resampling on datasets distributed in a non-iid fashion. Throughout the section, we illustrate the challenge, attacks, and defense by an example of training an MLP on the MNIST dataset (LeCun et al., 1998). In Appendix F, we present the results of similar experiments on Fashion-MNIST (Xiao et al., 2017) and CIFAR-10 (Krizhevsky et al., 2009). The dataset is sorted by labels and sequentially divided into equal parts among good workers; Byzantine workers have access to the dataset on all good workers.

4.1 Resampling against the attacks on non-iid data

In Section 2 we have presented how heterogeneous data can lead to failure of existing robust aggregation rules. Here we apply our proposed resampling with T=n, s=2 to the same aggregation rules, showing that resampling overcomes the described failures. Results are presented in Figure 2. In Figure 2a, we show that using resampling helps KRUM to achieve better test accuracy on non-iid data. Since resampling KRUM with s=2 actually averages 2 gradients, we compare it with MULTIKRUM with m=2. The middle of Figure 2a shows that MULTIKRUM with m=2 performs better than KRUM, but KRUM with resampling is even better which suggests the resampling step improves the performance on non-iid data. The selection histogram on the rightmost part of Figure 2a shows that after resampling, KRUM's selection is much more evenly distributed between the good workers. In Figure 2b, we show that resampling fixes RFA with T=1 and allows it to defend against the normalized mean attack. The resampling-based aggregation can almost reach same accuracy for both iid and non-iid setup. In Figure 2c, while mimic attack does not work for median-based rules in the iid setting, resampling still slightly improves the performance due to variance reduction. In the non-iid setting, resampling drastically improves the accuracy to the same level as the iid setting.

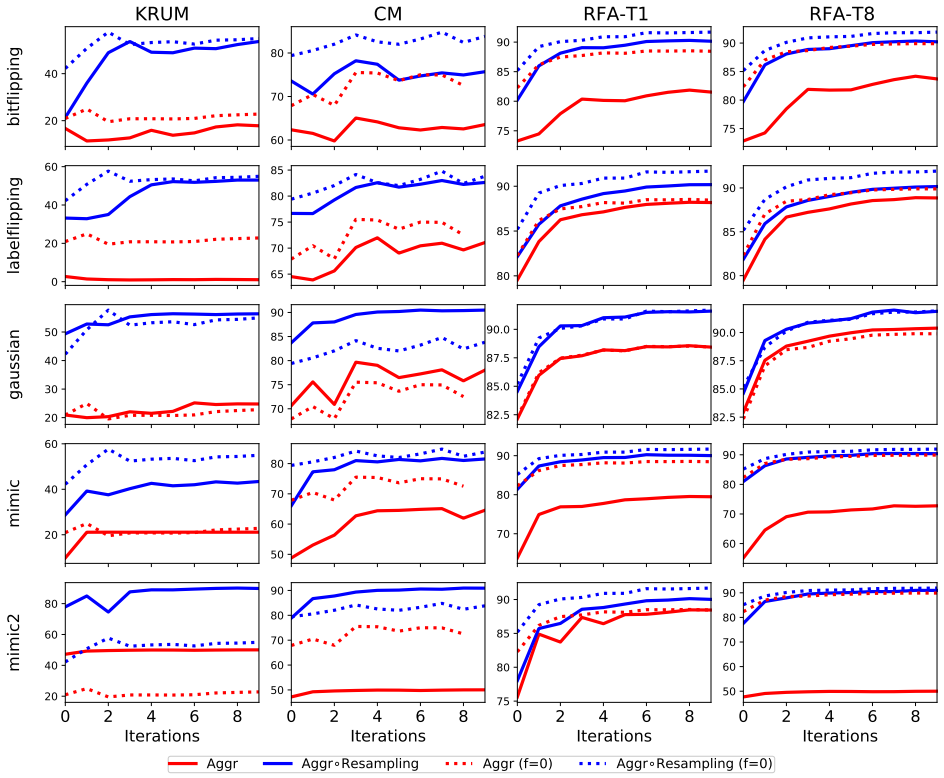


Figure 3: Test accuracies of KRUM, CM, RFA under 5 kinds of attacks (and without attack) on non-iid datasets. There are 10 workers and 2 of them are Byzantine according to each attack row. Columns show each aggregation rule applied without (red) and with resampling (blue). Dotted lines for comparison are showing the same method without any Byzantine workers ($f = 0$). For RFA, T1, T8 refers to the number of inner iterations of Weiszfeld's algorithm.

4.2 Resampling against general Byzantine attacks

In Figure 3, we present thorough experiments on non-iid data over 10 workers with 2 Byzantine workers. In each subfigure, we compare an aggregation rule with its variant with resampling. Three aggregation rules are compared: KRUM, CM, RFA. In particular, we compare to RFA with both T=1 (normalized mean) and T=8 (geometric median).

Attacks. 5 different kinds of attacks are applied (one per row in the figure): 1) *bitflipping*: attacker flips the sign bits and sends $-\nabla f(\boldsymbol{x})$ instead of $\nabla f(\boldsymbol{x})$; 2) *Labelflipping*: attacker transform labels through $\mathcal{T}(y) := 9 - y$; 3) *gaussian*: attacker samples a random gradient of 0 mean and isotropic covariance matrix with standard deviation 200 (Xie et al., 2018); 4) & 5) *mimic & mimic2*: explained in Section 2.3.

From Figure 3 we can see that resampling improves the accuracy on most of the tasks. The final accuracies achieved vary with the aggregation rules we use. Notice that RFA-T1 is more robust to the mimic attack than RFA-T8 in Figure 3 because more inner iterations lead to better approximate geometric median and less robust to normalized mean attacks.

5 Conclusion

In this paper, we initiated a study of robust distributed learning problem under realistic heterogeneous data. We showed that many existing Byzantine-robust aggregation rules fail under simple new attacks, or sometimes even without any Byzantine workers. As a solution, we propose a resampling scheme which effectively adapts existing robust algorithms to heterogeneous datasets at a negligible computational cost. We believe robustness under heterogeneous conditions has been an overlooked direction of research thus far and hope to inspire more work on this topic. Extending to the decentralized setting, stronger Byzantine adversaries, as well as obtaining optimal algorithms are other challenging directions for future work.

References

Alistarh, D., Allen-Zhu, Z., and Li, J. (2018). Byzantine stochastic gradient descent. In *NeurIPS - Advances in Neural Information Processing Systems*, pages 4613–4623.

Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., and Shmatikov, V. (2018). How to backdoor federated learning.

Baruch, M., Baruch, G., and Goldberg, Y. (2019). A little is enough: Circumventing defenses for distributed learning. *arXiv preprint arXiv:1902.06156*.

Bernstein, J., Zhao, J., Azizzadenesheli, K., and Anandkumar, A. (2018). signSGD with majority vote is communication efficient and fault tolerant. *arXiv preprint arXiv:1810.05291*.

Bhagoji, A. N., Chakraborty, S., Mittal, P., and Calo, S. (2018). Analyzing federated learning through an adversarial lens.

Blanchard, P., El Mhamdi, E. M., Guerraoui, R., and Stainer, J. (2017). Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent. In *NeurIPS - Advances in Neural Information Processing Systems 30*, pages 119–129.

Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., Kiddon, C., Konecny, J., Mazzocchi, S., McMahan, H. B., et al. (2019). Towards federated learning at scale: System design. In *SysML - Proceedings of the 2nd SysML Conference, Palo Alto, CA, USA*.

Chen, L., Wang, H., Charles, Z., and Papailiopoulos, D. (2018). Draco: Byzantine-resilient distributed training via redundant gradients. *arXiv preprint arXiv:1803.09877*.

Chen, Y., Su, L., and Xu, J. (2017). Distributed statistical machine learning in adversarial settings. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 1(2):1–25.

Damaskinos, G., El Mhamdi, E. M., Guerraoui, R., Guirguis, A. H. A., and Rouault, S. L. A. (2019). Aggregathor: Byzantine machine learning via robust gradient aggregation. *Conference on Systems and Machine Learning (SysML) 2019, Stanford, CA, USA*, page 19.

Data, D. and Diggavi, S. (2020). Byzantine-resilient sgd in high dimensions on heterogeneous data. *arXiv preprint arXiv:2005.07866*.

Diakonikolas, I., Kamath, G., Kane, D., Li, J., Moitra, A., and Stewart, A. (2019). Robust estimators in high-dimensions without the computational intractability. *SIAM Journal on Computing*, 48(2):742–864.

Ghosh, A., Hong, J., Yin, D., and Ramchandran, K. (2019). Robust federated learning in a heterogeneous environment. *arXiv preprint arXiv:1906.06629*.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., D'Oliveira, R. G. L., Rouayheb, S. E., Evans, D., Gardner, J., Garrett, Z., Gascón, A., Ghazi, B., Gibbons, P. B., Gruteser, M., Harchaoui, Z., He, C., He, L., Huo, Z., Hutchinson, B., Hsu, J., Jaggi, M., Javidi, T., Joshi, G., Khodak, M., Konecný, J., Korolova, A., Koushanfar, F., Koyejo, S., Lepoint, T., Liu, Y., Mittal, P., Mohri, M., Nock, R., Özgür, A., Pagh, R., Raykova, M., Qi, H., Ramage, D., Raskar, R., Song, D., Song, W., Stich, S. U., Sun, Z., Suresh, A. T., Tramèr, F., Vepakomma, P., Wang, J., Xiong, L., Xu, Z., Yang, Q., Yu, F. X., Yu, H., and Zhao, S. (2019). Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*.

Karimireddy, S. P., Rebjock, Q., Stich, S. U., and Jaggi, M. (2019). Error Feedback Fixes SignSGD and other Gradient Compression Schemes. In *ICML 2019 - Proceedings of the 36th International Conference on Machine Learning*.

Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images.

Lai, K. A., Rao, A. B., and Vempala, S. (2016). Agnostic estimation of mean and covariance. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 665–674. IEEE.

Lamport, L., Shostak, R., and Pease, M. (2019). The byzantine generals problem. In *Concurrency: the Works of Leslie Lamport*, pages 203–226.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Li, L., Xu, W., Chen, T., Giannakis, G. B., and Ling, Q. (2019). RSA: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1544–1551.

McMahan, H. B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. (2016). Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*.

Meeds, E., Hendriks, R., Al Faraby, S., Bruntink, M., and Welling, M. (2015). Mlitb: machine learning in the browser. *PeerJ Computer Science*, 1:e11.

Mhamdi, E. M. E., Guerraoui, R., and Rouault, S. (2018). The hidden vulnerability of distributed learning in byzantium. *arXiv preprint arXiv:1802.07927*.

Miura, K. and Harada, T. (2015). Implementation of a practical distributed calculation system with browsers and javascript, and application to distributed deep learning. *arXiv preprint arXiv:1503.05743*.

Peng, J. and Ling, Q. (2020). Byzantine-robust decentralized stochastic optimization. In *ICASSP 2020 - IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5935–5939. IEEE.

Pillutla, K., Kakade, S. M., and Harchaoui, Z. (2019). Robust Aggregation for Federated Learning. *arXiv preprint arXiv:1912.13445*.

Rajput, S., Wang, H., Charles, Z., and Papailiopoulos, D. (2019). Detox: A redundancy-based framework for faster and more robust gradient aggregation. *arXiv preprint arXiv:1907.12205*.

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

Su, L. and Xu, J. (2018). Securing distributed gradient descent in high dimensional statistical learning. *arXiv preprint arXiv:1804.10140*.

Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms.

Xie, C., Koyejo, O., and Gupta, I. (2018). Generalized Byzantine-tolerant SGD. *arXiv preprint arXiv:1802.10116*.

Xie, C., Koyejo, O., and Gupta, I. (2019a). Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance. In *ICML 2019 - 35th International Conference on Machine Learning*.

Xie, C., Koyejo, S., and Gupta, I. (2019b). Fall of Empires: Breaking Byzantine-tolerant SGD by Inner Product Manipulation. *arXiv preprint arXiv:1903.03936*.

Yang, Z. and Bajwa, W. U. (2019a). Bridge: Byzantine-resilient decentralized gradient descent. *arXiv preprint arXiv:1908.08098*.

Yang, Z. and Bajwa, W. U. (2019b). ByRDiE: Byzantine-resilient distributed coordinate descent for decentralized learning. *IEEE Transactions on Signal and Information Processing over Networks*.

Yin, D., Chen, Y., Ramchandran, K., and Bartlett, P. (2018a). Byzantine-robust distributed learning: Towards optimal statistical rates. *arXiv preprint arXiv:1803.01498*.

Yin, D., Chen, Y., Ramchandran, K., and Bartlett, P. (2018b). Defending against saddle point attack in byzantine-robust distributed learning. *arXiv preprint arXiv:1806.05358*.