Distributed Gradient Descent Algorithm Robust to an Arbitrary Number of Byzantine Attackers

Xinyang Cao and Lifeng Lai, *Senior Member, IEEE*

***Abstract*—Due to the growth of modern dataset size and the desire to harness computing power of multiple machines, there is a recent surge of interest in the design of distributed machine learning algorithms. However, distributed algorithms are sensitive to Byzantine attackers who can send falsified data to prevent the convergence of algorithms or lead the algorithms to converge to value of the attackers' choice. Some recent work proposed interesting algorithms that can deal with the scenario when up to half of the workers are compromised. In this paper, we propose a novel algorithm that can deal with an arbitrary number of Byzantine attackers. The main idea is to ask the parameter server to randomly select a small clean dataset and compute noisy gradient using this small dataset. This noisy gradient will then be used as a ground truth to filter out information sent by compromised workers. We show that the proposed algorithm converges to the neighborhood of the population minimizer regardless the number of Byzantine attackers. We further provide numerical examples to show that the proposed algorithm can benefit from the presence of good workers and achieve better performance than existing algorithms.**

***Index Terms*—Byzantine attacker, convergence, distributed gradient descent.**

I. Introduction

THE design of distributed optimization algorithms has attracted significant recent research interests [2]–[15]. The surge of interest in this area is motivated by many factors. Here we list some of them. First, as the amount of data keeps growing at a fast pace, it is challenging to fit all data in one machine [16]–[18]. Second, distributed optimization algorithms are useful to harness the computing power of multiple machines [16]–[18]. Third, in certain scenarios, data is naturally collected at different locations, and it is too costly to move all data to a centralized location [19].

In a typical distributed optimization setup, there are one parameter server and multiple workers. The whole dataset is divided by the server into small parts and each part is stored in one workers. Most of the existing works in this area assume that these workers behave honestly and follow the protocol. However, in practice, by using distributed optimization algorithms, there is a risk that some of the workers might be compromised. Compromised workers (also called Byzantine attackers in the sequel) can prevent the convergence of the optimization algorithms or lead the algorithms to converge to a value chosen by these attackers by modifying or falsifying intermediate results during the execution of optimization algorithms. For example, as shown in [20], [21], the presence of even a single Byzantine worker can prevent the convergence of distributed gradient descent algorithm.

There have been some interesting recent works to design distributed machine learning algorithms [9]–[11], [20]–[29] that can deal with Byzantine attacks. The main idea of these works is to compare information received from all workers, and compute a quantity that is robust to attackers for algorithm update. For example, the algorithm in [20] uses the geometric median mean of gradient information received from workers for parameter update. The algorithm Krum in [21] chooses the gradient vector that is closest (in certain sense) to its $m - p$ neighbors, where m is the number of workers and p is the number of compromised workers, to be the estimated gradient for parameter updating. Alistarh *et al.* [22] propose a Byzantine-resilient SGD algorithm, in which at each iteration the server combines the current and past gradient information from each worker to compute next update, to solve convex problem with high dimension. Xie *et al.* [23] compare the performance of algorithms that use geometric median, marginal median and mean median to update parameters. Yin *et al.* [24] propose a median-based algorithm that uses only one communication round to perform parameter updates. Chen *et al.* [25] propose DRACO algorithm that uses ideas from coding theory to determine which machines are under attack. Damaskinos *et al.* [26] consider an asynchronous distributed training scenario and propose algorithm Kardam that leverages the Lipschitzness of the cost function to filter out gradient information from attackers. Su *et al.* [27] propose an approximate gradient descent algorithm that employs iterative filtering for robust gradient aggregation in high dimensional estimation. Yin *et al.* [28] consider the problem of defending against saddle point attack and propose a Byzantine PGD that uses random perturbation of the iterates to escape saddle points and compares the performance of three ways for robust aggregation: median, trimmed mean and iterative filtering. These algorithms in [20]–[28] can successfully converge to the neighborhood of the population minimizer even if up to half of all workers are compromised. However, once more than half of the workers are compromised, the algorithms in these interesting work will

Manuscript received April 6, 2019; revised July 31, 2019; accepted October 1, 2019. Date of publication October 7, 2019; date of current version October 28, 2019. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Wenwu Wang. The work of X. Cao and L. Lai was supported by the National Science Foundation under grants ECCS-17-11468, CCF-17-17943, CNS-1824553, and CCF-1908258. This paper was presented in part at the IEEE International Conference on Acoustics, Speech and Signal Processing, AB, Canada, April 2018 [1]. *(Corresponding author: Lifeng Lai.)*

The authors are with the Department of Electrical and Computer Engineering, University of California, Davis, CA 95616 USA (e-mail: xycao@ucdavis.edu; lflai@ucdavis.edu).

Digital Object Identifier 10.1109/TSP.2019.2946020

not converge. As machine learning algorithms are increasingly deployed in security and safety critical applications, it is important to consider the robustness of these algorithms in adversarial environments where we need to make less or no assumption about the attackers (including the assumption that less than half of the workers are attackers in the distributed learning) [30]. These attacks may be achieved by a variety of ways including but not limited to: vulnerable communication channels, poisoned datasets, or virus. In 2006, 65% of companies surveyed in the CSI/FBI Computer Crime and Security Survey [31] reported that they had been attacked by virus. Once captured by virus, these devices can be used to attack the network from inside. For example, Xie *et al.* [29] consider the case that the number of Byzantine worker is arbitrarily large and propose an algorithm named Zeno that the server first sorts the gradient by a stochastic descendant score then averages the $m-b$ gradients with highest score, in which m is the total number of workers and b is an important parameter in the algorithm. The algorithm must have at least one good worker, it cannot solve the problem when server is isolated. Furthermore, in order to properly set the parameter b, Zeno must know an upper bound on the number of Byzantine workers. In addition, if b is selected to be larger than the true number of attackers, the algorithm may not benefit from all good workers.

In this paper, we propose a new robust distributed gradient descent algorithm that can converge to the neighborhood of the population minimizer regardless of the number of compromised workers (i.e. even when more than half of workers are compromised). The main idea is to ask the server to randomly select a *small* subset of clean data and compute a noisy gradient based on this small dataset. Even though the computed gradient is very noisy, it can be used as a proxy of the ground truth to filter out information from attackers. In particular, once the server receives gradient information from workers, it compares the gradient information from each worker with the noisy gradient it has computed. If the distance between the gradient from worker and the noisy gradient computed by itself is small, the server accepts the gradient information from that worker as authentic. After the comparison step, the server then computes the average of all accepted gradient and its own noisy gradient as the final estimated gradient for updating. We prove that the algorithm can converge to the neighborhood of the population minimizer regardless of the number of compromised workers. We show this result by proving that the distance between the estimated gradient and the true gradient can be universally bounded. In the analysis, we consider two different scenarios. In the first scenario, we do not assume any knowledge about the number of attackers. We provide a convergence proof in this case with minimal assumption about attackers. In the second scenario, we assume that the number of attackers is bounded from above by a constant p. We note that, here p is an upper bound of the number of attackers, it is not the exact number of attackers. Hence, this additional knowledge is not too restrictive. With this additional knowledge, we provide a modified algorithm that has a tighter convergence bound.

The paper is organized as follows. In Section II, we describe the model. In Section III, we describe the proposed robust gradient descent algorithm. In Section IV, we analyze the convergence property of the proposed algorithm. In Section V, we provide numerical examples to validate the theoretic analysis and show that we can benefit from the good workers to obtain a better convergence accuracy. Finally, we offer several concluding remarks in Section VI. The proofs are collected in Appendix.

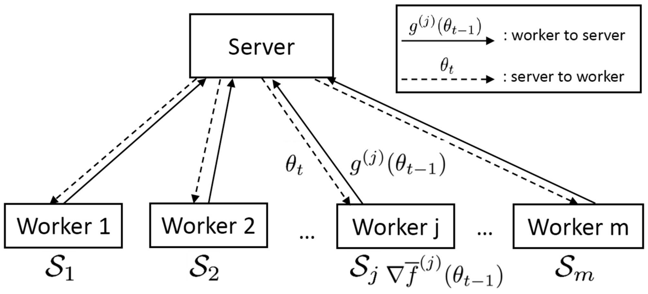

Server
$g^{(j)}(\theta_{t-1})$: worker to server
θ_t : server to worker
θ_t
$g^{(j)}(\theta_{t-1})$
Worker 1
Worker 2
...
Worker j
...
Worker m
$\mathcal{S}_1$
$\mathcal{S}_2$
$\mathcal{S}_j$ $\nabla\overline{f}^{(j)}(\theta_{t-1})$
$\mathcal{S}_m$

Fig. 1. Distributed optimization model.

II. Model

In this section, we introduce our model. Suppose that the data $X \in \mathcal{X} \subset \mathbb{R}^n$ is generated randomly from a unknown distribution $\mathcal{D}$ parameterized by unknown vector θ taken value from a set $\Theta \subset \mathbb{R}^d$. Our goal is to infer the unknown parameter θ from data samples. In particular, consider a loss function $f : \mathcal{X} \times \Theta \rightarrow \mathbb{R}$, with $f(x, \theta)$ being the risk induced by data point x under the model parameter θ. We aim to find the model parameter θ^* that minimizes the population risk $F(\theta)$:

$$\theta^* \in \arg\min_{\theta\in\Theta} F(\theta) \triangleq \mathbb{E}[f(X,\theta)]. \qquad (1)$$

In this paper, we assume that $F(\theta)$ satisfies the following typical assumption.

Assumption 1: The population risk function $F : \Theta \rightarrow \mathbb{R}$ is L-strongly convex, and differentiable over Θ with M-Lipschitz gradient. That is for all $\theta, \theta' \in \Theta$,

$$F(\theta') \geq F(\theta) + \langle \nabla F(\theta), \theta' - \theta \rangle + L \parallel \theta' - \theta \parallel^2 /2, \qquad (2)$$

and

$$\parallel \nabla F(\theta') - \nabla F(\theta) \parallel \leq M \parallel \theta' - \theta \parallel,$$

in which $\parallel \cdot \parallel$ is the ℓ_2 norm and $0 < L \leq M$.

When we know the distribution of X, the population risk can be evaluated exactly and θ^* can be computed by solving the above problem (1). However, in a typical machine learning problem, the distribution is unknown. To handle this, one normally approximates the population risk $F(\theta)$ from the observed data samples. In particular, we assume that there exist N independently and identically distributed (i.i.d.) data samples X_i, with $i = 1, 2, \ldots, N$, from the distribution $\mathcal{D}$. Instead of minimizing the population risk (1) directly, we minimize the empirical risk

$$\min_{\theta\in\Theta} \frac{1}{N} \sum_{i=1}^{N} f(X_i, \theta). \qquad (3)$$

Consider a typical distributed optimization model in Fig. 1, there are one server and m workers in the system. These N data samples are distributed into these m workers, and the server can communicate with all workers synchronously. Let $\mathcal{S}_j$ be the set of data samples that the j-th worker receives from the server. In a system with data shuffling, $\mathcal{S}_j$ changes over iterations, while in a system without shuffling, $\mathcal{S}_j$ is fixed. Our algorithm and proof hold regardless whether there is data shuffling or not.

In the classic batch gradient descent, each worker solves (3) using distributed gradient descent. In particular, at iteration t, each worker $j \in [1, m]$ calculates $\nabla \overline{f}^{(j)}(\theta_{t-1})$ based on local data

$$\nabla \overline{f}^{(j)}(\theta_{t-1}) = \frac{1}{|\mathcal{S}_j|} \sum_{i \in \mathcal{S}_j} \nabla f(X_i, \theta_{t-1}), \tag{4}$$

and sends it back to the server, where $|\mathcal{S}_j|$ is the size of data in the j-th worker. After receiving information from all workers, the server updates the parameter using

$$\theta_t = \theta_{t-1} - \eta \sum_{i=1}^{m} w_i \nabla \overline{f}^{(j)}(\theta_{t-1}) \tag{5}$$

where $w_i = |\mathcal{S}_i|/N$ and sends the updated parameter θ_t to workers. Here η is the step size. This process continues until a certain stop criteria is satisfied.

In this paper, we consider a system with Byzantine workers, in which an unknown subset of workers might be comprised. Furthermore, the set of compromised workers might change over time. If a worker is compromised, instead of the gradient calculated from local data, it can send arbitrary information to the server. In particular, let $\mathcal{B}_t$ denote the set of compromised workers at iteration t, the server receives data $g^{(j)}(\theta_{t-1})$ from the j-th worker with

$$g^{(j)}(\theta_{t-1}) = \begin{cases} \nabla \overline{f}^{(j)}(\theta_{t-1}) & j \notin \mathcal{B}_t \\ \star & j \in \mathcal{B}_t \end{cases}, \tag{6}$$

in which $\star$ denotes an arbitrary vector chosen by the attacker.

We assume there are up to p Byzantine attackers in the system. Note that, in this paper, p is not the exact number of attackers, it is merely an upper bound on the number of attackers. In this case with Byzantine attackers, if one continues to use the classic batch gradient as in (5), the algorithm will fail to converge even if there is only one attacker [20], [21]. As discussed above, [20], [21] designed algorithms that converge to the neighborhood of the population minimizer if the number of compromised machines p is less than $m/2$ (i.e., more than half of the machines are not compromised).

The goal of our paper is to design a robust batch gradient descent algorithm that can tolerate *any number* of Byzantine attackers.

III. Algorithm

In this section, we describe our algorithm that can deal with an arbitrary number of Byzantine attackers under two scenarios: p being unknown and knowing the value of p.

TABLE I
Proposed Algorithm With Unknown p

```
Algorithm
Parameter server:
Initialize: Randomly selects $\theta_0 \in \Theta$;
For $t \geq 1$ :
    randomly selects $\mathcal{S}_0$;
    1: Broadcasts the current model parameter
    estimator $\theta_{t-1}$ to all workers;
    2: Waits to receive gradients from the $m$ workers;
    $g^{(j)}(\theta_{t-1})$ denote the value received from worker $j$;
    3: Computes $\nabla \overline{f}^{(0)}(\theta_{t-1})$ using $\mathcal{S}_0$;
    4: Compares $g^{(j)}(\theta_{t-1})$ with $\nabla \overline{f}^{(0)}(\theta_{t-1})$; If
    $\| g^{(j)}(\theta_{t-1}) - \nabla \overline{f}^{(0)}(\theta_{t-1}) \| \leq \xi \| \nabla \overline{f}^{(0)}(\theta_{t-1}) \|$,
    the server accepts it and sets it to be $q_t^{(l)}(\theta_{t-1})$;
    5: Assume the acceptable value are in $\mathcal{V}_t$, then
    $G(\theta_{t-1}) \leftarrow \sum_{l \in \mathcal{V}_t} w_l q_t^{(l)}(\theta_{t-1}) + w_0 \nabla \overline{f}^{(0)}(\theta_{t-1})$;
    6:Updates $\theta_t \leftarrow \theta_{t-1} - \eta G(\theta_{t-1})$;
Worker $j$:
For $t \geq 1$ :
    1: Computes the gradient $\nabla \overline{f}^{(j)}(\theta_{t-1})$;
    2: If worker $j$ is honest,
    it sends $\nabla \overline{f}^{(j)}(\theta_{t-1})$ back to the server;
    If worker $j$ is compromised,
    it sends the value determined by the attacker;
```

A. Unknown p

In the first scenario, we do not have any knowledge about p. Main steps of the algorithm under the first scenario is listed in Table I. The main idea of our algorithm is to ask the server to randomly select a small set of data points $\mathcal{S}_0$ at very beginning, where $|\mathcal{S}_0| \leq \min_{j \in [1,m]} |\mathcal{S}_j|$. Once $\mathcal{S}_0$ is selected, it is fixed throughout the algorithm. Then at each iteration t, the server calculates a noisy gradient using data points in $\mathcal{S}_0$:

$$\nabla \overline{f}^{(0)}(\theta_{t-1}) = \frac{1}{|\mathcal{S}_0|} \sum_{i \in \mathcal{S}_0} \nabla f(X_i, \theta_{t-1}).$$

Different choices of the size of $\mathcal{S}_0$ will strike a tradeoff between convergence speed and computational complexity.

The server then compares $g^{(j)}(\theta_{t-1})$ received from worker j with $\nabla \overline{f}^{(0)}(\theta_{t-1})$. The server will accept $g^{(j)}(\theta_{t-1})$ as authentic value and use it for further processing, if

$$\| g^{(j)}(\theta_{t-1}) - \nabla \overline{f}^{(0)}(\theta_{t-1}) \| \leq \xi \| \nabla \overline{f}^{(0)}(\theta_{t-1}) \|, \tag{7}$$

where ξ is a constant. The choice of ξ will impact the proposed scheme. Roughly speaking, choosing a smaller ξ can limit the effect of an attack, but it may also reject more correct information from honest workers. On the other hand, a larger ξ can increase the probability of data from honest workers being accepted, but it will also increase the probability of accepting information from attackers. We will discuss how to choose this parameter in the analysis.

Assuming there are $|\mathcal{V}_t|$ values (which is a random variable) being accepted after the comparison step at iteration t, we denote

TABLE II
PROPOSED ALGORITHM WITH KNOWN p

```
Algorithm
Parameter server:
Initialize: Randomly selects $\theta_0 \in \Theta$;
For $t \geq 1$ :
    randomly selects $\mathcal{S}_0$;
    1: Broadcasts the current model parameter
    estimator $\theta_{t-1}$ to all workers;
    2: Waits to receive gradients from the $m$ workers;
    $g^{(j)}(\theta_{t-1})$ denote the value received from worker $j$;
    3: Computes $\nabla\overline{f}^{(0)}(\theta_{t-1})$ using $\mathcal{S}_0$;
    4: Compares $g^{(j)}(\theta_{t-1})$ with $\nabla\overline{f}^{(0)}(\theta_{t-1})$; If
    $\| g^{(j)}(\theta_{t-1}) - \nabla\overline{f}^{(0)}(\theta_{t-1}) \| \leq \xi \| \nabla\overline{f}^{(0)}(\theta_{t-1}) \|$,
    the server accepts it and sets it to be $q_t^{(l)}(\theta_{t-1})$;
    5: After accepting, the server collects $m - p$ gradient
    information which are closest to its own.
    5: Then
    $G(\theta_{t-1}) \leftarrow \sum_{j\in\mathcal{U}_t} w_j q_t^{(j)}(\theta_{t-1}) + w_0\nabla\overline{f}^{(0)}(\theta_{t-1})$;
    6:Updates $\theta_t \leftarrow \theta_{t-1} - \eta G(\theta_{t-1})$;
```

these values by $q_t^{(1)}(\theta_{t-1}), \ldots, q_t^{(|\mathcal{V}_t|)}(\theta_{t-1})$. Then the server updates the parameters as $\theta_t = \theta_{t-1} - \eta G(\theta_{t-1})$, where

$$G(\theta_{t-1}) = \sum_{l\in\mathcal{V}_t} w_l q_t^{(l)}(\theta_{t-1}) + w_0\nabla\overline{f}^{(0)}(\theta_{t-1}), \tag{8}$$

where $w_l = \frac{|\mathcal{S}_l|}{\sum_{i\in\mathcal{V}_t}|\mathcal{S}_i| + |\mathcal{S}_0|}$.

B. Known p

In the second scenario, we assume that we know p. We again note that here p is an upper bound of the number of attackers, it is not the exact number of attackers. Hence, this additional knowledge is not too restrictive. With this additional knowledge, we modify the algorithm at the server side above slightly. The worker side remains the same. This modification will allow us to prove a tighter bound in the convergence analysis section. Main steps of the modified algorithm at the server side is listed in Table II. The main difference with the algorithm in Table I is that we now sort the gradient information accepted by the server in an increasing order by $\|q^{(i)}(\theta_t) - \nabla\overline{f}^{(0)}(\theta_t)\|$, then keep the first $m - p$ gradient value in a set (as we know the number p). We call this set $\mathcal{U}_t$ at iteration t. Using this notation, the gradient used for updating at the server can be written as

$$G(\theta_{t-1}) = \sum_{j\in\mathcal{U}_t} w_j q_t^{(j)}(\theta_{t-1}) + w_0\nabla\overline{f}^{(0)}(\theta_{t-1}),$$

where $w_j = \frac{|\mathcal{S}_j|}{\sum_{i\in\mathcal{U}_t}|\mathcal{S}_i| + |\mathcal{S}_0|}$.

IV. CONVERGENCE ANALYSIS

In this section, we analyze the convergence property of the proposed algorithm. We consider two different scenarios: 1) In scenario 1, we do not have any knowledge about p; 2) In scenario 2, we assume that we know the value of p.

In this section, we will prove results that hold simultaneously for all $\theta \in \Theta$ with a high probability. Hence, in the following, we will drop subscript $t - 1$. Before presenting detailed analysis, here we describe the high level ideas. It is well known that if $\nabla F(\theta)$ is available, then the gradient descent algorithm will converge to θ^* exponentially fast. The main idea of our proof is to show that, regardless of the number of attackers, the distance between $G(\theta)$ and $\nabla F(\theta)$ is universally bounded in Θ in both scenarios. Hence, $G(\theta)$ is a good estimate of $\nabla F(\theta)$. As the result, we can then show that the proposed algorithm converges to the neighborhood of the population minimizer.

A. Scenario 1: No Assumption on p

In this scenario, we assume that we do not know even the upperbound on the number of bad workers. We first show that $\|G(\theta) - \nabla F(\theta)\|$ is universally bounded in Θ regardless the number of attackers.

Lemma 1: For an arbitrary number of attackers, the distance between $G(\theta)$ and $\nabla F(\theta)$ is bounded as

$$\begin{aligned}\|G(\theta) - \nabla F(\theta)\| \leq (1+\xi)\|\nabla F(\theta) - \nabla\overline{f}^{(0)}(\theta)\| \\ + \xi\|\nabla F(\theta)\|, \quad \forall\theta.\end{aligned} \tag{9}$$

Proof: Please see Appendix A. ■

We next need to bound the two terms in the right hand side of (9). The term $\|\nabla F(\theta)\| = \|\nabla F(\theta) - \nabla F(\theta^*)\|$ can be bounded using the M-Lipschitz gradient assumption in Assumption 1. In the following, we show that the term $\|\nabla F(\theta) - \nabla\overline{f}^{(0)}(\theta)\|$ can also be bounded. For this, we need to present several assumptions and intermediate results. These assumptions are similar to those used in [20], [24], [27], and proofs of some lemmas follow closely that of [20].

Assumption 2: There exist positive constants σ_1 and α_1 such that for any unit vector $v \in B$, $\langle\nabla f(X, \theta^*), v\rangle$ is sub-exponential with σ_1 and α_1, that is,

$$\sup_{v\in B}\mathbb{E}[\exp(\lambda\langle\nabla f(X,\theta^*), v\rangle)] \leq e^{\sigma_1^2\lambda^2/2}, \quad \forall|\lambda| \leq 1/\alpha_1,$$

where B denotes the unit sphere $\{v : \|v\|_2 = 1\}$.

With this assumption, we first have the following lemma that shows $\frac{1}{|\mathcal{S}_0|}\sum_{i\in\mathcal{S}_0}\nabla f(X_i, \theta^*)$ concentrates around $\nabla F(\theta^*)$.

Lemma 2: Under Assumption 2, for any $\delta \in (0, 1)$, let

$$\Delta_1 = \sqrt{2}\sigma_1\sqrt{(d\log 6 + \log(3/\delta))/|\mathcal{S}_0|}, \tag{10}$$

and if $\Delta_1 \leq \sigma_1^2/\alpha_1$, then

$$\Pr\left\{\left\|\frac{1}{|\mathcal{S}_0|}\sum_{i\in\mathcal{S}_0}\nabla f(X_i,\theta^*) - \nabla F(\theta^*)\right\| \geq 2\Delta_1\right\} \leq \frac{\delta}{3}.$$

Proof: Please see Appendix B. ■

Second, we define gradient difference $h(x, \theta) \triangleq \nabla f(x, \theta) - \nabla f(x, \theta^*)$ and assume that for every θ, $h(x, \theta)$ normalized by $\| \theta - \theta^* \|$ is also sub-exponential.

Assumption 3: There exist positive constants σ_2 and α_2 such that for any $\theta \in \Theta$ with $\theta \neq \theta^*$ and any unit vector $v \in B$, $\langle h(X, \theta) - \mathbb{E}[h(X, \theta)], v\rangle / \| \theta - \theta^* \|$ is sub-exponential with

σ_2 and α_2, that is,

$$\sup_{\theta\in\Theta, v\in B} \mathbb{E}\left[\exp\left(\frac{\lambda\langle h(X,\theta) - E[h(X,\theta)], v\rangle}{\|\theta-\theta^*\|}\right)\right] \leq e^{\sigma_2^2\lambda^2/2}, \quad \forall|\lambda| \leq \frac{1}{\alpha_2}.$$

This allows us to show that $\frac{1}{|\mathcal{S}_0|}\sum_{i\in\mathcal{S}_0} h(X_i,\theta)$ concentrates on $\mathbb{E}[h(X,\theta)]$ for every fixed θ.

Assumption 2 and 3 ensure that random gradient $\nabla f(\theta)$ has good concentration properties, i.e., an average of $|\mathcal{S}_0|$ $i.i.d$ random gradients $\frac{1}{|\mathcal{S}_0|}\sum_{i\in\mathcal{S}_0}\nabla f(X_i,\theta)$ sharply concentrates on $\nabla F(\theta)$ for every fixed θ.

Lemma 3: If Assumption 3 holds, for any $\delta \in (0,1)$ and any fixed $\theta\in\Theta$, let $\Delta_1' = \sqrt{2}\sigma_2\sqrt{(d\log 6 + \log(3/\delta))/|\mathcal{S}_0|}$, and if $\Delta_1' \leq \sigma_2^2/\alpha_2$, then

$$\Pr\left\{\left\|\frac{1}{|\mathcal{S}_0|}\sum_{i\in\mathcal{S}_0} h(X_i,\theta) - \mathbb{E}[h(X,\theta)]\right\| \geq 2\Delta_1'\|\theta-\theta^*\|\right\} \leq \frac{\delta}{3}.$$

Proof: Please see Appendix C. ■

Assumption 4: For any $\delta\in(0,1)$, there exists an $M' = M'(\delta)$ such that

$$\Pr\left\{\sup_{\theta,\theta'\in\Theta:\theta\neq\theta'} \frac{\|\nabla f(X,\theta) - \nabla f(X,\theta')\|}{\|\theta-\theta'\|} \leq M'\right\} \geq 1 - \frac{\delta}{3}.$$

Assumption 4 ensures that $\nabla f(X,\theta)$ is M'-Lipschitz with high probability.

With these assumptions and intermediate lemmas, we are ready to state our universal bound for $\|\nabla F(\theta) - \nabla\overline{f}^{(0)}(\theta)\|$.

Proposition 1: Suppose Assumptions 2–4 hold, and $\Theta \subset \{\theta :\| \theta - \theta^* \|\leq r\sqrt{d}\}$ for some $r > 0$. For any $\delta_1\in(0,1)$,

$$\Pr\{\forall\theta : \|\nabla F(\theta) - \nabla\overline{f}^{(0)}(\theta)\| \leq 8\Delta_2\|\theta-\theta^*\| + 4\Delta_1\} \geq 1-\delta_1, \tag{11}$$

in which $\Delta_1 = \sqrt{2}\sigma_1\sqrt{(d\log 6 + \log(3/\delta_1))/|\mathcal{S}_0|}$ and $\Delta_2 = \sqrt{2}\sigma_2\sqrt{(\tau_1+\tau_2)/|\mathcal{S}_0|}$, with $\tau_1 = d\log 18 + d\log((M\vee M')/\sigma_2)$, and $\tau_2 = 0.5\, d\log(|\mathcal{S}_0|/d) + \log(3/\delta_1) + \log\left(\frac{2r\sigma_2^2\sqrt{|\mathcal{S}_0|}}{\alpha_2\sigma_1}\right)$.

Proof: (Outline): The proof relies on the typical ϵ-net argument. Let $\Theta_\epsilon = \{\theta_1,\ldots,\theta_{N_\epsilon}\}$ be an ϵ-cover of Θ, i.e., for fix any $\theta\in\Theta$, there exists a $\theta_j\in\Theta_\epsilon$ such that $\| \theta - \theta_j \|\leq \epsilon$. By triangle inequality,

$$\begin{aligned}\left\|\frac{1}{|\mathcal{S}_0|}\sum_{i\in\mathcal{S}_0}\nabla f(X_i,\theta) - \nabla F(\theta)\right\| &\leq \|\nabla F(\theta) - \nabla F(\theta_j)\| \\ &+ \left\|\frac{1}{|\mathcal{S}_0|}\sum_{i\in\mathcal{S}_0}(\nabla f(X_i,\theta) - \nabla f(X_i,\theta_j))\right\| \\ &+ \left\|\frac{1}{|\mathcal{S}_0|}\sum_{i\in\mathcal{S}_0}\nabla f(X_i,\theta_j) - \nabla F(\theta_j)\right\|.\end{aligned}$$

Then first term can be upper bounded using assumption 1. The second term can be bounded using assumption 4, and the third term can be bounded using Lemma 3. We can then employ union bound over Θ_ϵ to finish the argument. Please see Appendix D for details. ■

Combining Lemma 1 and Proposition 1, we know that $G(\theta)$ is a good approximation of $\nabla F(\theta)$. Using this fact, we have the following convergence result.

Theorem 1: If Assumptions 1–4 hold, and $\Theta\subset\{\theta :\| \theta - \theta^* \|\leq r\sqrt{d}\}$ for some $r>0$, choose $0<\eta<L/M^2$, then regardless of the number of attackers with probability at least $1-\delta_1$ that

$$\|\theta_t - \theta^*\| \leq (1-\rho_1)^t\|\theta_0 - \theta^*\| + (4\eta\Delta_1 + 4\eta\xi\Delta_1)/\rho_1,$$

in which

$$\rho_1 = 1 - \left(\sqrt{1+\eta^2M^2 - \eta L} + 8\Delta_2\eta + \eta\xi(8\Delta_2 + M)\right). \tag{12}$$

Proof: Please see Appendix E. ■

This theorem shows that under an event that happens with a high probability, the estimated θ can converge to the neighborhood of θ^* exponentially fast. However, the convergence accuracy bound is not tighter than the bound one could obtain if the algorithm uses gradient descent calculated from $\mathcal{S}_0$ only. This is because we are working with an adversarial setup, for which we need to derive a bound that holds in the worst-case scenario. When there is no assumption on p, the worst-case scenario is when all workers are under attack, which corresponds to the case where the server can only trust the data from $\mathcal{S}_0$ only but it still uses data from these Byzantine workers since these data pass the comparison test. Our numerical results in Section V will illustrate that the actual performance of the proposed algorithm is better than the case with using data from $\mathcal{S}_0$ only and it can benefit from the presence of honest workers even when more than half of the workers are Byzantine workers.

B. Scenario 2: Known p

In this section, we assume that we know an upper bound on the number of Byzantine workers. Note that p is not the exact number of Byzantine workers, it is merely an upper bound. Furthermore, p could be larger than $m/2$. Hence, this is not a too restrictive assumption. With this additional knowledge, we can derive a tighter convergence result. To proceed, we use $\mathcal{H}_t$ to denote the set of honest workers whose gradient information are accepted by the server at iteration t, and $\mathcal{A}_t$ denote the set of attackers whose information are accepted by the server at iteration t. The values of these two sets are unknown. We only know that $|\mathcal{A}_t|\leq p$. Let $k = |\mathcal{H}_t| + |\mathcal{A}_t|$. Using this notation, the gradient used for updating at the server can be written as

$$G(\theta_t) = \sum_{j\in\mathcal{H}_t\cap\mathcal{U}_t} w_j\nabla\overline{f}^{(j)}(\theta_t) + w_0\nabla\overline{f}^{(0)}(\theta_t) + \sum_{j\in\mathcal{A}_t\cap\mathcal{U}_t} w_j g^{(j)}(\theta_t), \tag{13}$$

in which $\mathcal{U}_t$ is defined in Section III.

Similar to Section IV-A, we will prove results that hold simultaneously for all $\theta\in\Theta$ with a high probability. We will show that all gradient information from all honest workers have a high probability to be accepted, hence we will drop the subscript t from $\mathcal{H}_t$ to $\mathcal{H}$. By exploiting the knowledge of p, we provide a tighter bound on $\|G(\theta) - \nabla F(\theta)\|$ than the one presented in Lemma 1.

Lemma 4: If there are up to p attackers, at iteration t, the distance between $G(\theta)$ and $\nabla F(\theta)$ is bounded as

$$\begin{aligned}&\|G(\theta)-\nabla F(\theta)\|\\ &\quad\le \frac{\sum_{j\in\mathcal{H}\cap\mathcal{U}_t}|\mathcal{S}_j|+|\mathcal{S}_0|}{\sum_{j\in\mathcal{U}_t}|\mathcal{S}_j|+|\mathcal{S}_0|}\|C_t(\theta)-\nabla F(\theta)\|\\ &\qquad+\sum_{j\in\mathcal{A}_t\cap\mathcal{U}_t} w_j\|g^{(j)}(\theta)-\nabla F(\theta)\|,\end{aligned}\tag{14}$$

where $C_t\left(\theta\right)=\sum_{j\in\mathcal{H}\cap\mathcal{U}_t}\beta_j\,\nabla\,\overline{f}^{(j)}\left(\theta\right)+\beta_0\,\nabla\,\overline{f}^{(0)}\left(\theta\right)$, $\beta_j=\frac{|\mathcal{S}_j|}{\sum_{i\in\mathcal{H}\cap\mathcal{U}_t}|\mathcal{S}_i|+|\mathcal{S}_0|}$ and $w_j=\frac{|\mathcal{S}_j|}{\sum_{i\in\mathcal{U}_t}|\mathcal{S}_i|+|\mathcal{S}_0|}$.

Proof: Please see Appendix F. ■

Before further simplifying (14), we present several supporting lemmas.

The following lemma shows that, by choosing ξ properly, the gradient information from an honest user will be accepted by the server with a high probability.

Lemma 5: Suppose we set ξ as $c|\mathcal{S}_0|^{-1/4}$ and $|\mathcal{S}_0|$ sufficiently large, then for each honest worker j and $\delta_1\in(0,1)$

$$\|\nabla\overline{f}^{(j)}(\theta)-\nabla\overline{f}^{(0)}(\theta)\|\le\xi\|\nabla\overline{f}^{(0)}(\theta)\|,\forall\theta\in\Theta\tag{15}$$

holds with probability $(1-\delta_1)^2-\delta_1$.

Proof: Please refer to Appendix G. ■

From Lemma 5, we can see that data sent by a honest worker has a high probability to pass the comparison test in the server. We can define event Υ_1 such that information from all $|\mathcal{H}|$ good workers satisfies (15). Using union bound, we know that information from these $|\mathcal{H}|$ honest workers will all be accepted with $\Pr\{\Upsilon_1\}\ge 1-\delta_3$, where $\delta_3=1-\{[(1-\delta_1)^2-\delta_1]|\mathcal{H}|-(|\mathcal{H}|-1)\}$.

We now bound the first term in (14), namely $\|C_t(\theta)-\nabla F(\theta)\|$ at iteration t. Towards this goal, consider a set $\mathcal{NC}_t=\{\mathcal{C}_t^1,\mathcal{C}_t^2,\mathcal{C}_t^3,\ldots\}$, each of which represents one possibility of choosing $|\mathcal{H}\cap\mathcal{U}_t|$ workers from $|\mathcal{H}|$ at iteration t. We have $|\mathcal{NC}_t|=\binom{|\mathcal{H}|}{|\mathcal{H}\cap\mathcal{U}_t|}$.

Proposition 2: Suppose Assumptions 2–4 hold, and $\Theta\subset\{\theta:\|\theta-\theta^*\|\le r\sqrt{d}\}$ for some positive parameter r, at iteration t, for any $\delta_2\in(0,1)$

$$\Pr\{\forall\theta:\|C_t(\theta)-\nabla F(\theta)\|\le 8\Delta_6\|\theta-\theta^*\|+4\Delta_5\}\ge 1-\delta_2,$$

in which

$$\Delta_5=\sqrt{2}\sigma_1\sqrt{(d\log 6+\log(3|\mathcal{NC}_t|/\delta_2))/|\mathcal{S}_t|},\tag{16}$$

$$\Delta_6=\sqrt{2}\sigma_2\sqrt{(\tau_1+\tau_2)/|\mathcal{S}_t|},\tag{17}$$

with $\tau_1=d\ \log\ 18+d\ \log\left(\left(\ M\vee M'\ \right)/\sigma_2\ \right)$,$\tau_2=0.5\ d$ $\log(\frac{\max_{\mathcal{C}_t^l\in\mathcal{NC}_t}(\sum_{j\in\mathcal{C}_l}|\mathcal{S}_j|+|\mathcal{S}_0|)}{d})+\log(3/\delta_2)+\log(\frac{2r\sigma_2^2\sqrt{|\mathcal{S}_t|}}{\alpha_2\sigma_1})$, and $|\mathcal{S}_t|=\min_{\mathcal{C}_t^l\in\mathcal{NC}_t}(\sum_{j\in\mathcal{C}_t^l}|\mathcal{S}_j|+|\mathcal{S}_0|)$.

Proof:

$$\|C_t(\theta)-\nabla F(\theta)\|\le\sup_{\mathcal{C}_t^l\in\mathcal{NC}_t}\|Q_t^l(\theta)-\nabla F(\theta)\|,\tag{18}$$

where

$$Q_t^l(\theta)=\sum_{j\in\mathcal{C}_t^l}\beta_j\nabla\overline{f}^{(j)}(\theta)+\beta_0\nabla\overline{f}^{(0)}(\theta),\tag{19}$$

where $\beta_j=\frac{|\mathcal{S}_j|}{\sum_{i\in\mathcal{C}_t^l}|\mathcal{S}_i|+|\mathcal{S}_0|.}$ Then by union bound, at iteration t, we need to prove

$$\Pr\{\forall\theta:\left\|Q_t^l(\theta)-\nabla F(\theta)\right\|\le 8\Delta_6\|\theta-\theta^*\|+4\Delta_5\}\ge 1-\frac{\delta_2}{|\mathcal{NC}_t|}.$$

The remaining proof is similar to the proof of Proposition 1 and hence is omitted for brevity. ■

For the second term in (14), each Byzantine gradient information in $\mathcal{U}_t$ must follow the inequality

$$\|g(\theta)-\nabla F(\theta)\|\le\max_{j\in\mathcal{H}}\left\|\nabla\overline{f}^{(j)}(\theta)-\nabla F(\theta)\right\|.\tag{20}$$

Using this fact, we have the following proposition.

Proposition 3: Suppose Assumptions 2–4 hold, and $\Theta\subset\{\theta:\|\theta-\theta^*\|\le r\sqrt{d}\}$ for some positive parameter r. For any $\delta_2\in(0,1)$

$$\begin{aligned}\Pr\Big\{\forall\theta:\max\nolimits_{j\in\mathcal{H}}\left\|\nabla\overline{f}^{(j)}(\theta)-\nabla F(\theta)\right\|\le 8\Delta_8\|\theta-\theta^*\|+4\Delta_7\Big\}\\ \ge 1-\delta_2,\end{aligned}$$

in which

$$\Delta_7=\sqrt{2}\sigma_1\sqrt{(d\log 6+\log\left(3|\mathcal{H}|/\delta_2\right))\,(\min_{j\in\mathcal{H}}|\mathcal{S}_j|)},\tag{21}$$

and

$$\Delta_8=\sqrt{2}\sigma_2\sqrt{(\tau_3+\tau_4)/\min_{j\in\mathcal{H}}|\mathcal{S}_j|},\tag{22}$$

with $\tau_3=d\log 18+d\log\left(\left(\ M\vee M'\right)/\sigma_2\right)$, and $\tau_4=0.5\,d$ $\log\left(\frac{\max_{j\in\mathcal{H}}|\mathcal{S}_j|}{d}\right)+\log(3/\delta_2)+\log\left(\frac{2r\sigma_2^2\sqrt{\min_{j\in\mathcal{H}}|\mathcal{S}_j|}}{\alpha_2\sigma_1}\right)$.

Proof: The proof is similar to the proof of Proposition 2 and hence is omitted for brevity. ■

Using these two propositions, we now further bound (14) from above by examining the worst-case scenario with regards to $\mathcal{U}_t$. At iteration t, the right-hand side of (14) has a high probability to be bounded by

$$\begin{aligned}&8\Delta_6\|\theta-\theta^*\|+4\Delta_5\\ &\quad+8(\Delta_8-\Delta_6)\|\theta-\theta^*\|\frac{\sum_{j\in\mathcal{B}\cap\mathcal{U}_t}|\mathcal{S}_j|}{\sum_{j\in\mathcal{U}_t}|\mathcal{S}_j|+|\mathcal{S}_0|}\\ &\quad+4(\Delta_7-\Delta_5)\frac{\sum_{j\in\mathcal{B}\cap\mathcal{U}_t}|\mathcal{S}_j|}{\sum_{j\in\mathcal{U}_t}|\mathcal{S}_j|+|\mathcal{S}_0|}\end{aligned}\tag{23}$$

When $|\mathcal{H}\cap\mathcal{U}_t|\ge 1$, we can find $\Delta_8\ge\Delta_6$ and $\Delta_7\ge\Delta_5$, as Δ_8 and Δ_7 have a smaller denominator. Hence, the coefficient of $(\sum_{j\in\mathcal{B}\cap\mathcal{U}_t}|\mathcal{S}_j|)/(\sum_{j\in\mathcal{U}_t}|\mathcal{S}_j|+|\mathcal{S}_0|)$ in the second term and third term of (23) are non-negative. As the result, (23) is a nondecreasing function of $(\sum_{j\in\mathcal{B}\cap\mathcal{U}_t}|\mathcal{S}_j|)/(\sum_{j\in\mathcal{U}_t}|\mathcal{S}_j|+|\mathcal{S}_0|)$. We now consider two different cases with fixed denominator.

Case 1: $p<m/2$. In this case, $p<m-p$, hence $\max\{\sum_{j\in\mathcal{B}\cap\mathcal{U}_t}|\mathcal{S}_j|\}$ with setting $|\mathcal{B}\cap\mathcal{U}_t|=p$ will maximize (23) and also maximize the right-hand side of (14) in iteration

t. This implies that, when $p < m/2$, the worst-case scenario is that there are p Byzantine workers and these gradients are all in $\mathcal{U}_t$ all the time.

Case 2: $p \geq m/2$. In this case, since $|\mathcal{B} \cap \mathcal{U}_t| \leq m - p$ and $m - p \leq p$, $\max\{\sum_{j \in \mathcal{B} \cap \mathcal{U}_t} |\mathcal{S}_j|\}$ with setting $|\mathcal{B} \cap \mathcal{U}_t| = m - p - 1$ will maximize (23) as argued above. We need to consider additional value when $\max\{\sum_{j \in \mathcal{B} \cap \mathcal{U}_t} |\mathcal{S}_j|\}$ with $|\mathcal{B} \cap \mathcal{U}_t| = m - p$, which means $m - p$ gradient information in $\mathcal{U}_t$ are all from Byzantine workers. In this case $\Delta_8 < \Delta_6$ and $\Delta_7 < \Delta_5$, since we assume $\min_{j \in \mathcal{H}} |\mathcal{S}_j| \geq |\mathcal{S}_0|$. The obtained bound with $\max\{\sum_{j \in \mathcal{B} \cap \mathcal{U}_t} |\mathcal{S}_j|\}$ by setting $|\mathcal{B} \cap \mathcal{U}_t| = m - p$ is larger than the bound obtained by setting $|\mathcal{B} \cap \mathcal{U}_t| = m - p - 1$ in (23). This implies that, when $p \geq m/2$, the worst-case occurs when all $m - p$ gradient information in $\mathcal{U}_t$ are from Byzantine workers.

From the discussion above, we know that with a high probability, the gradient information from all honest works will be accepted. Furthermore, regardless of the true number of attackers, the right-hand side of (14) is bounded by the scenario where $\min\{m - p, p\}$ number of gradient in $\mathcal{U}_t$ are from Byzantine workers all the time.

With these supporting lemmas and propositions, we are ready for our main convergence result under 2 cases.

Theorem 2: If there are up to p attackers, Assumptions 1–4 hold and $\Theta \subset \{\theta : \| \theta - \theta^* \| \leq r\sqrt{d}\}$ for some $r > 0$, choose $0 < \eta < L/M^2$, we have

$$\|\theta_t - \theta^*\| \leq (1 - \rho_2)^t \|\theta_0 - \theta^*\| + (\eta\gamma_1)/\rho_2, \tag{24}$$

hold simultaneously for all θ_t with probability at least $1 - 2\delta_2 - \delta_3$. Here

$$\rho_2 = 1 - \left(\sqrt{1 + \eta^2 M^2 - \eta L} + \eta\gamma_2\right), \tag{25}$$

$$\gamma_1 = 4(1 - w_{\max})\Delta_5 + 4w_{\max}\Delta_7, \tag{26}$$

and

$$\gamma_2 = 8(1 - w_{\max})\Delta_6 + 8w_{\max}\Delta_8, \tag{27}$$

with $w_{\max} = \max\{(\sum_{j \in \mathcal{B} \cap \mathcal{U}_t} |\mathcal{S}_j|)/(\sum_{j \in \mathcal{U}_t} |\mathcal{S}_j| + |\mathcal{S}_0|)\}$ and $|\mathcal{B} \cap \mathcal{U}_t| = \min\{m - p, p\}$ and $|\mathcal{U}_t| = m - p$.

Proof: Please see Appendix H. ■

Theorem 2 shows that under the event which would happen with highly probability, the estimated θ can converge to the neighborhood of θ^* exponentially fast.

From the discussion above, since $\sum_{j \in \mathcal{H} \cap \mathcal{U}} |\mathcal{S}_j| + |\mathcal{S}_0|$ is greater or equal to $|\mathcal{S}_0|$, $\Delta_6 \leq \Delta_2$. Then, $\gamma_2 \leq (8\Delta_2 + 8\xi\Delta_2 + \xi M)$ and $\rho_2 \geq \rho_1$, Hence, the convergence performance benefits from knowing an upperbound on the number of Byzantine workers.

V. Numerical Results

In this section, we provide numerical examples, with both synthesized data and real data, to illustrate the analytical results.

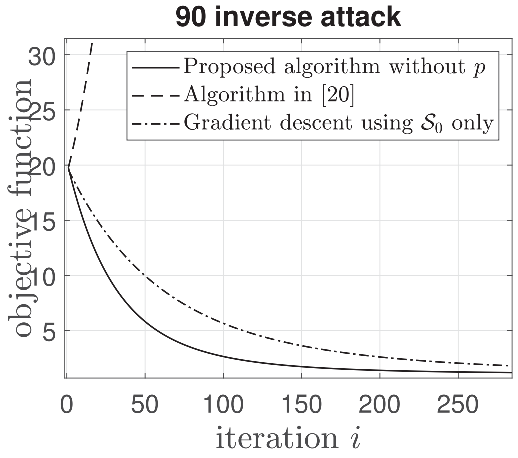


Fig. 2. Synthesized data: 90 Inverse attack.

A. Synthesized Data

We first use synthesized data. In this example, we focus on linear regression, in which

$$Y_i = X_i^T \theta^* + \epsilon_i, i = 1, 2, \ldots, N,$$

where $X_i \in \mathbb{R}^d$, θ^* is a $d \times 1$ vector and ϵ_i is the noise. We set $\mathbf{X} = [X_1, \ldots, X_N]$ as $d \times N$ data matrix. In the simulation, we set the dimension $d = 20$, the total number of data $N = 100000$, the number of workers $m = 100$, and evenly distribute data among these machines. We set $\epsilon_i \overset{i.i.d.}{\sim} \mathcal{N}(0, 1)$. Here $\mathcal{N}(\mu, \sigma^2)$ denotes Gaussian variables with mean μ and variance σ^2. Furthermore, we set $|\mathcal{S}_0| = 1000, \xi = 1.5|\mathcal{S}_0|^{-\frac{1}{4}} = 0.2667$. We use $\mathcal{N}(0, 4)$ to independently generate each entry of θ^*. After θ^* is generated, we fix it. The data matrix $\mathbf{X}$ is generated randomly by Gaussian distribution with $\mu = 0$ and fixed known maximal and minimal eigenvalues of the correlation matrix $\mathbf{X}^T\mathbf{X}$. Let $\lambda_{\max}(\cdot)$ and $\lambda_{\min}(\cdot)$ denote the maximal and minimal eigenvalue of $\mathbf{X}^T\mathbf{X}$ respectively. In the following figures, we use $\lambda_{\max}(\mathbf{X}^T\mathbf{X}) = 200$ and $\lambda_{\min}(\mathbf{X}^T\mathbf{X}) = 2$ to generate the data matrix $\mathbf{X}$, then generate Y_i using the linear relationship mentioned above. We illustrate our results with two different attacks: 1) Inverse attack, in which each attacker first calculates the gradient information $\nabla \overline{f}^{(j)}(\theta_{t-1})$ based on the its local data but sends the inversed version $-\nabla \overline{f}^{(j)}(\theta_{t-1})$ to the server; and 2) Random attack, in which the attacker randomly generates gradient value. In our simulation, we compare three algorithms: 1) Gradient descent using only data from $\mathcal{S}_0$, i.e., the server ignores information from all workers; 2) Algorithm proposed in [20]; and 3) The proposed algorithm described in Table I.

Figs. 2 and 3 plot the value of the loss function vs iteration with 90 and 48 inverse attacks respectively. When the attacker number is 48, which is less than half of the total number, all three algorithms can converge. However, from Fig. 2, it is clear that the algorithm in [20] does not converge as the number of attackers is more than half of the total number of machines. The proposed algorithm, however, still converges in the presence of 90 attackers. Furthermore, even though there are only 10 honest workers and the server does not know the identities of

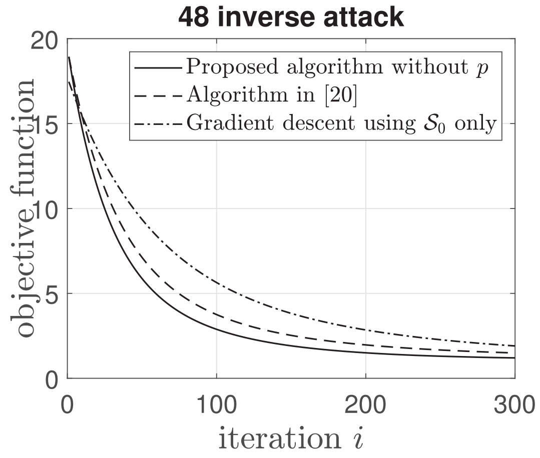


Fig. 3. Synthesized data: 48 Inverse attack.

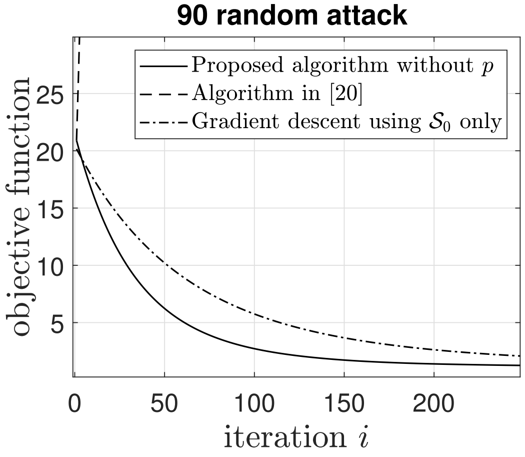


Fig. 4. Synthesized data: 90 Random attack.

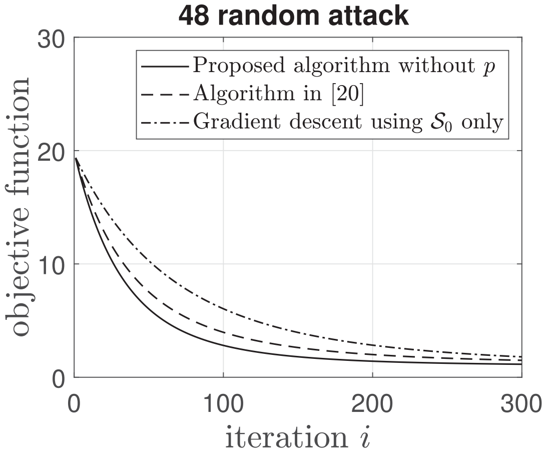


Fig. 5. Synthesized data: 48 Random attack.

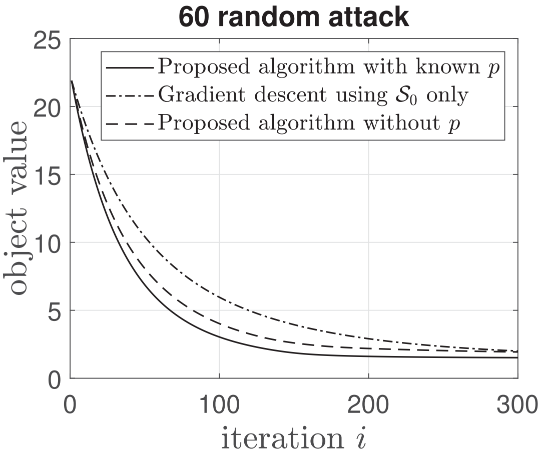


Fig. 6. Synthesized data: 60 Random attack.

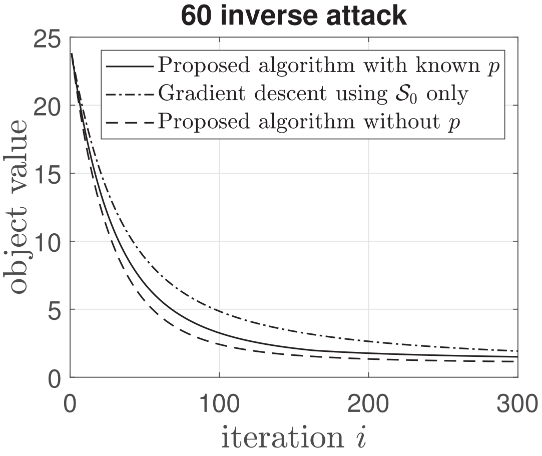


Fig. 7. Synthesized data: 60 Inverse attack.

TABLE III
RUNNING TIME COMPARISON

loss function 1.9	time/iter	iteration	time
algorithm without p	1.0904×10^{-4}	140	0.0153
algorithm with p	1.5575×10^{-4}	174	0.0271
GD using S_0 only	9.5889×10^{-5}	300	0.0288

these honest workers, the proposed algorithm can still benefit from these workers, as the proposed algorithm outperforms the algorithm that only relies on information from $\mathcal{S}_0$.

Figs. 4 and 5 plot the value of the loss function vs iteration with 90 and 48 random attacks respectively. Similar to the scenario with inverse attack, all three algorithms can converge when there are less than half of the total number attackers. However, when there are 90 attackers, our algorithm outperforms the algorithm that uses $\mathcal{S}_0$ only, while the algorithm in [20] diverges.

Figs. 6 and 7 plot the value of the loss function vs iteration with 60 random and 60 inverse attacks respectively for the cases with and without knowledge of p. For the case with knowledge about p, we set $p = 75$. From Fig. 6, we can see that the proposed algorithm without knowing p has a lower convergence accuracy and convergence rate when comparing with the proposed algorithm knowing p. The main reason is that, when facing random attack, some attack vectors can pass the comparison test. In Fig. 7, since the attacks are inverse attack, the proposed algorithm can successfully reject all the information from attackers, then the proposed algorithm without knowing p has more data to update the parameter.

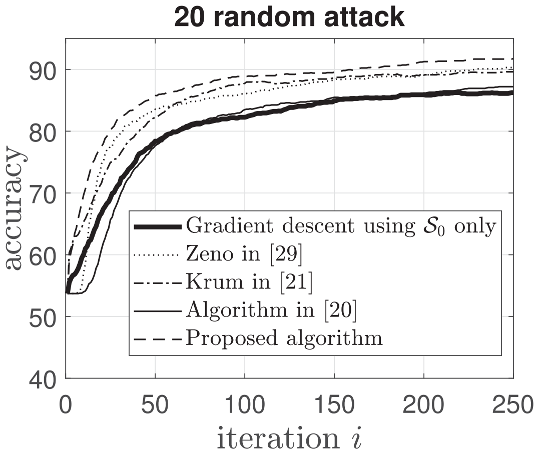


Fig. 8. MNIST: 20 Random attack.

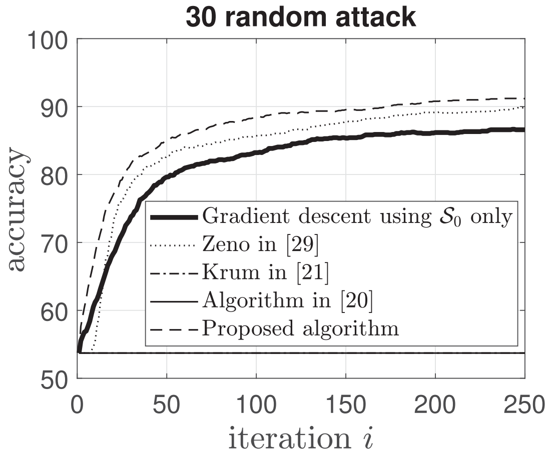


Fig. 9. MNIST: 30 Random attack.

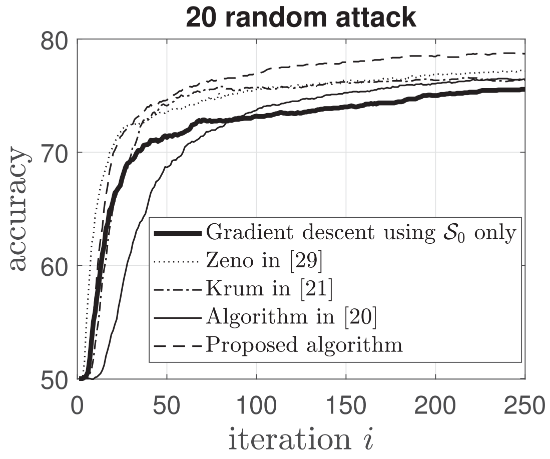


Fig. 10. CIFAR-10: 20 Random attack.

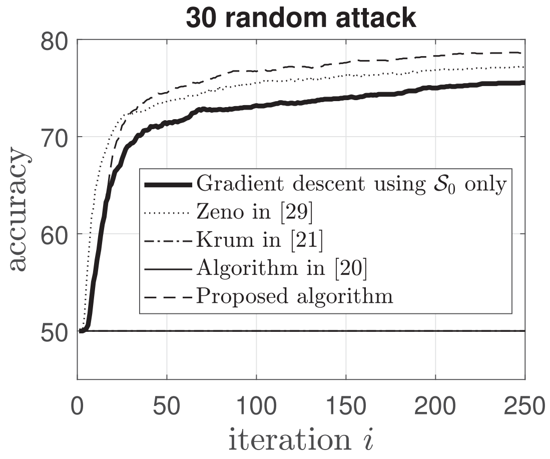


Fig. 11. CIFAR-10: 30 Random attack.

Table III lists the running time for three algorithms under 60 inverse attacks, and we measure the number of iterations needed for the loss function to reach 1.9. In Table III, the simulations are produced under the same testing environment. From Table III, we can see that, compared with the case of using data from $\mathcal{S}_0$ only, the proposed algorithms have a higher complexity per iteration, but they reduce the number of iterations. Both proposed algorithms have a better performance than the gradient descent using $|\mathcal{S}_0|$ only, since they can benefit from the gradient information received from workers, even though the server does not know whether the workers are honest or not.

B. Real Data

Now we test our algorithms on real datasets MNIST [32] and CIFAR-10 [33], and compare our algorithms with various existing work [20], [21], [29]. MNIST is a widely used computer vision dataset that consists of 70,000 28 × 28 pixel images of handwritten digits 0 to 9. We use the handwritten images of 3 and 5, which are the most difficult to distinguish in this dataset, to build a logistic regression model. After picking all 3 and 5 images from the dataset, the total number of images is 13454. It is divided into a training subset of size 12000 and a testing subset of size 1454. The CIFAR-10 dataset consists of 60,000 32 × 32 images in 10 classes. For CIFAR-10 dataset, we pick the images of car and plane, and build a training subset of size 10000 and a testing subset of 2000. For these two datasets, we set the number of workers to be 50 and we random pick 200 images from both subset to build $\mathcal{S}_0$, and set the step size to be 0.01 for MNIST and 0.005 for CIFAR-10. Similar to the synthesized data scenario, we illustrate our results with two different attacks, namely inverse attack and random attack, and compare the performance of five algorithms: Zeno [29], where we set the cutoff number (a design parameter in Zeno) to be 5, Krum [21], median-mean [20], proposed algorithm without known p and the algorithm that server using only data $\mathcal{S}_0$. The following figures show how the testing accuracy varies with training iteration.

Figs. 8, 9, 10, and 11 illustrate the impact of 20 and 30 random attacks on different algorithms respectively. Figs. 8, 9 are generated using MNIST, while Figs. 10, and 11 are generated using CIFAR-10. Figs. 8 and 10 show that all algorithms have high accuracy when there are 20 attacks. Gradient descent using $\mathcal{S}_0$ only have the lowest accuracy since it uses a small size of data for training. The proposed algorithm has the best performance even though less than half of the workers are attackers. Figs. 9

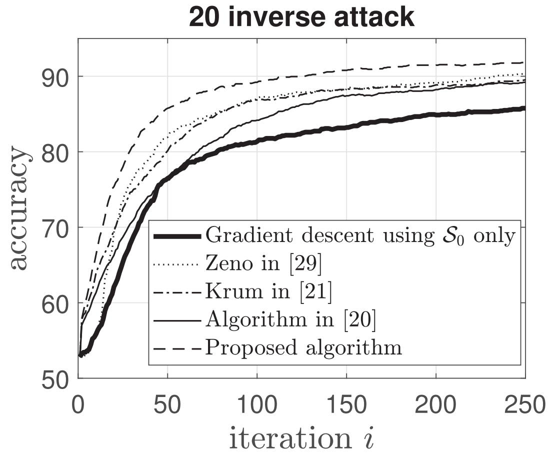


Fig. 12. MNIST: 20 Inverse attack.

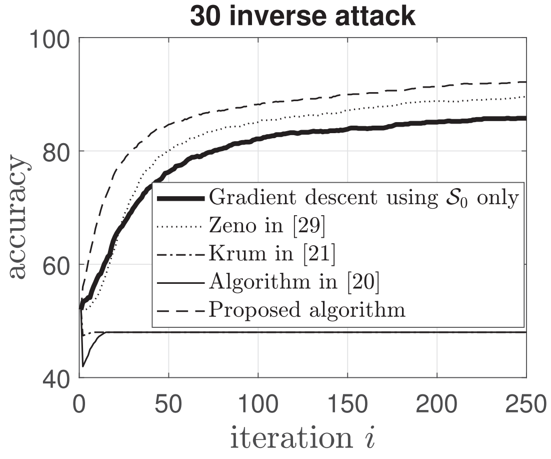


Fig. 13. MNIST: 30 Inverse attack.

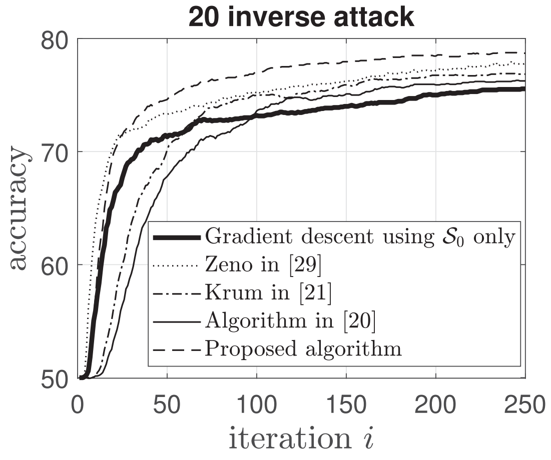


Fig. 14. CIFAR-10: 20 Inverse attack.

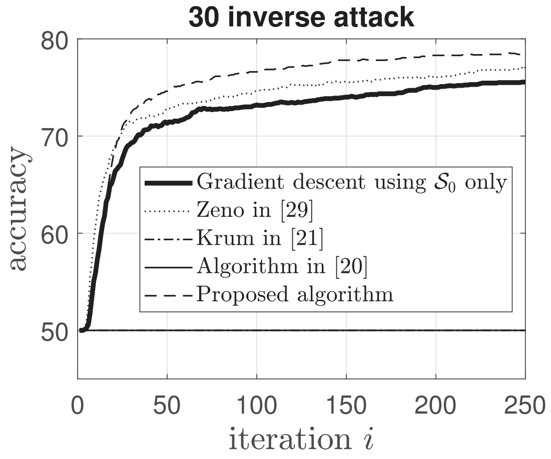


Fig. 15. CIFAR-10: 30 Inverse attack.

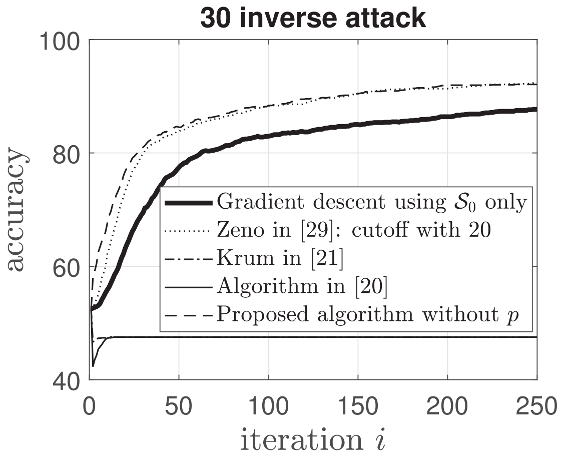


Fig. 16. MNIST: 30 Inverse attack.

and 11 show the algorithm using median-mean and Krum fail to predict if there are 30 attackers. Our proposed algorithm and Zeno still show high accuracy, and outperform the algorithm that only relies on information from $\mathcal{S}_0$. Furthermore, the proposed algorithm has better performance than Zeno.

We plot the impact of different number of inverse attacks on real data in Figs. 12, 13, 14, and 15 using MNIST and CFAIR10 respectively. All algorithms can converge when there are 20 inverse attacks. However, as the number of attackers is very close to half of the total number, the algorithm in [20] converges very slowly. Again, the proposed algorithm has the best performance even though less than half of the workers are attackers. Furthermore, if there are 30 Byzantine workers, Krum and median-mean algorithm cannot properly work. The algorithm that only based on information from $\mathcal{S}_0$ still performs well, since it does not use the information from all workers. Our proposed algorithm and Zeno can still work well. They can benefit from the 20 good workers, and outperform the scheme with $\mathcal{S}_0$ only. Our proposed algorithm also outperforms Zeno.

In Figs. 13 and 16, we plot the impact of choosing different cutoff values in Zeno. In Fig. 16, the cutoff value is 20, Zeno and our proposed algorithm both use all good gradient information, so both algorithms have similar performance. In Fig. 13, the cutoff value is 5. Although there are 20 good workers, Zeno can only benefit from 5 good workers, but our proposed algorithm can still benefit from all good workers and has a better performance.

Figs. 17 and 18 illustrate the testing accuracy v.s. training time under 20 and 30 inverse attacks with different algorithms.

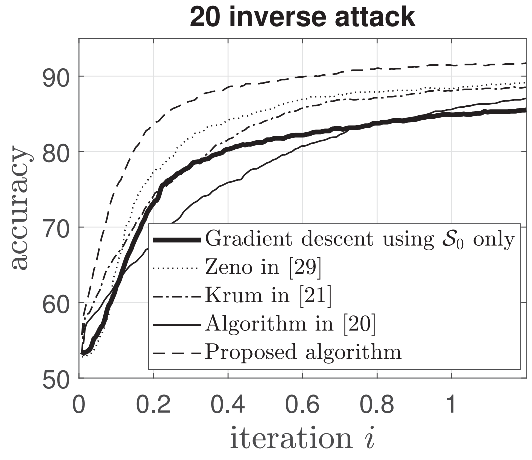


Fig. 17. MNIST: 20 Inverse attack.

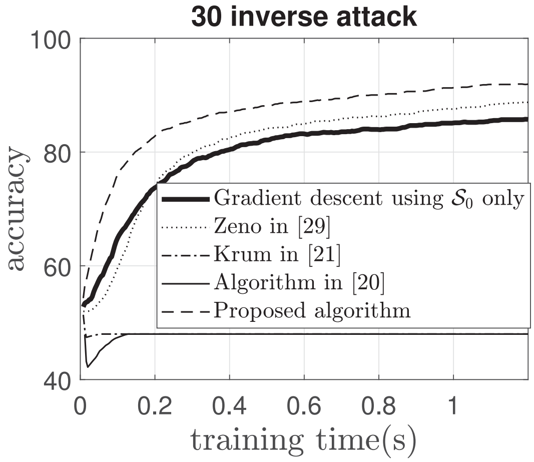


Fig. 18. MNIST: 30 Inverse attack.

All algorithms can converge when there are 20 inverse attacks. Since algorithms have higher complexity, some algorithms converges slower than the gradient descent using $|\mathcal{S}_0|$ only. But our proposed algorithm has a better performance in general.

VI. Conclusion

In this paper, we have proposed a robust gradient descent algorithm that can tolerant an arbitrary number of Byzantine attackers. We have shown that the proposed algorithm converges regardless the number of Byzantine attackers and have provided numerical examples to illustrate the performance of the proposed algorithm. In terms of future work, we hope to extend the analysis to scenarios with non-convex cost functions.

Appendix A
Proof of Lemma 1

$$
\begin{aligned}
&\|G(\theta)-\nabla F(\theta)\| \\
&= \left\| \sum_{l\in\mathcal{V}_t} w_l q_t^{(l)}(\theta_{t-1}) + w_0 \nabla \overline{f}^{(0)}(\theta_{t-1}) - \nabla F(\theta) \right\| \\
&= \left\| \sum_{l\in\mathcal{V}_t} w_l (q_t^{(l)}(\theta_{t-1}) - \nabla \overline{f}^{(0)}(\theta_{t-1})) + \nabla \overline{f}^{(0)}(\theta) - \nabla F(\theta) \right\| \\
&\le \sum_{l\in\mathcal{V}_t} w_l \|q_t^{(l)}(\theta) - \nabla \overline{f}^{(0)}(\theta)\| + \|\nabla \overline{f}^{(0)}(\theta) - \nabla F(\theta)\| \\
&\le \sum_{l\in\mathcal{V}_t} w_l \xi \|\nabla \overline{f}^{(0)}(\theta)\| + \|\nabla \overline{f}^{(0)}(\theta) - \nabla F(\theta)\| \\
&\le \sum_{l\in\mathcal{V}_t} w_l \xi \|\nabla \overline{f}^{(0)}(\theta) - \nabla F(\theta)\| + \sum_{l\in\mathcal{V}_t} w_l \xi \|\nabla F(\theta)\| \\
&\quad + \|\nabla \overline{f}^{(0)}(\theta) - \nabla F(\theta)\| \\
&\le (1+\xi)\|\nabla \overline{f}^{(0)}(\theta) - \nabla F(\theta)\| + \xi \|\nabla F(\theta)\|. \qquad (28)
\end{aligned}
$$

Appendix B
Proof of Lemma 2

Let $V=\{v_1,v_2,\ldots,v_{N_{1/2}}\}$ denote an $\frac{1}{2}$-cover of unit sphere B, i.e., for fix any $v\in B$, there exists a $v_j\in V$ such that $\parallel v-v_j \parallel \le \frac{1}{2}$. From [34], we have $\log N_{1/2}\le d\log 6$, and

$$
\begin{aligned}
&\left\| \frac{1}{|\mathcal{S}_0|} \sum_{i\in\mathcal{S}_0} \nabla f(X_i,\theta^*) - \nabla F(\theta^*) \right\| \\
&\le 2 \sup_{v\in V} \left\{ \frac{1}{|\mathcal{S}_0|} \sum_{i\in\mathcal{S}_0} \langle \nabla f(X_i,\theta^*) - \nabla F(\theta^*), v\rangle \right\}. \qquad (29)
\end{aligned}
$$

By Assumption 2 and the condition $\Delta_1\le\sigma_1^2/\alpha_1$, it follows from concentration inequalities for sub-exponential random variables [35] that

$$
\begin{aligned}
&\Pr\left\{ \frac{1}{|\mathcal{S}_0|} \sum_{i\in\mathcal{S}_0} \langle \nabla f(X_i,\theta^*) - \nabla F(\theta^*), v\rangle \ge \Delta_1 \right\} \\
&\le \exp(-|\mathcal{S}_0|\Delta_1^2/(2\sigma_1^2)). \qquad (30)
\end{aligned}
$$

By union bound and (29), we have

$$
\begin{aligned}
&\Pr\left\{ \left\| \frac{1}{|\mathcal{S}_0|} \sum_{i\in\mathcal{S}_0} \nabla f(X_i,\theta^*) - \nabla F(\theta^*) \right\| \ge 2\Delta_1 \right\} \\
&\le \exp(-|\mathcal{S}_0|\Delta_1^2/(2\sigma_1^2) + d\log 6). \qquad (31)
\end{aligned}
$$

Setting $\Delta_1=\sqrt{2}\sigma_1\sqrt{(d\log 6+\log(3/\delta))(|\mathcal{S}_0|)}$ in (31), we obtain the desired result.

Appendix C
Proof of Lemma 3

Define a set V using the same way in Appendix B. We have

$$
\begin{aligned}
&\frac{\|\frac{1}{|\mathcal{S}_0|}\sum_{i\in\mathcal{S}_0} h(X_i,\theta) - \mathbb{E}[h(X,\theta)]\|}{\|\theta-\theta^*\|} \\
&\le 2\sup_{v\in V} \left\{ \frac{1}{|\mathcal{S}_0|} \sum_{i\in\mathcal{S}_0} \frac{\langle h(X_i,\theta) - \mathbb{E}[h(X,\theta)], v\rangle}{\parallel \theta-\theta^* \parallel} \right\}. \qquad (32)
\end{aligned}
$$

By Assumption 3 and the condition $\Delta_1'\le\sigma_2^2/\alpha_2$, it follows from concentration inequalities for sub-exponential random variables [35] that

$$\Pr\left\{\frac{1}{|\mathcal{S}_0|}\sum_{i\in\mathcal{S}_0}\frac{\langle h(X_i,\theta)-E[h(X,\theta)],v\rangle}{\parallel\theta-\theta^*\parallel}\geq\Delta_1'\right\}$$
$$\leq\exp(-|\mathcal{S}_0|(\Delta_1')^2/(2\sigma_2^2)). \tag{33}$$

By union bound and (32),

$$\Pr\left\{\left\|\frac{1}{|\mathcal{S}_0|}\sum_{i\in\mathcal{S}_0}h(X_i,\theta)-\mathbb{E}[h(X,\theta)]\right\|\geq 2\Delta_1'\left\|\theta-\theta^*\right\|\right\}$$
$$\leq\exp(-|\mathcal{S}_0|(\Delta_1')^2/(2\sigma_2^2)+d\log 6).$$

By setting $\Delta_1'=\sqrt{2}\sigma_2\sqrt{(d\log 6+\log(3/\delta))(|\mathcal{S}_0|)}$, the proof is complete.

Appendix D
Proof of Proposition 1

Suppose Assumption 2, Assumption 3 and Assumption 4 hold, $\delta_1\in(0,1)$ and $\Theta\subset\{\theta:\parallel\theta-\theta^*\parallel\leq r\sqrt{d}\}$ for some positive parameter r. let

$$\tau=\frac{\alpha_2\sigma_1}{2\sigma_2^2}\sqrt{\frac{d}{|\mathcal{S}_0|}},u^*=\left\lceil\frac{r\sqrt{d}}{\tau}\right\rceil, \tag{34}$$

We define Θ_u for any positive integer $1\leq u\leq u^*$. $\Theta_u\triangleq\{\theta:\parallel\theta-\theta^*\parallel\leq\tau u\}$. Suppose that $\theta_1,\ldots,\theta_{N_\epsilon}$ is an ϵ-cover of Θ_τ, where ϵ is given by

$$\epsilon=\frac{\sigma_2\tau u}{M\vee M'}\sqrt{\frac{d}{|\mathcal{S}_0|}}. \tag{35}$$

Then $\log N_\epsilon\leq d\log(3\tau u/\epsilon)$. Fix any $\theta\in\Theta_u$, there exists a $1\leq j\leq N_\epsilon$ that $\parallel\theta-\theta_j\parallel\leq\epsilon$. By triangle's inequality,

$$\left\|\frac{1}{|\mathcal{S}_0|}\sum_{i\in\mathcal{S}_0}\nabla f(X_i,\theta)-\nabla F(\theta)\right\|\leq\|\nabla F(\theta)-\nabla F(\theta_j)\|$$
$$+\left\|\frac{1}{|\mathcal{S}_0|}\sum_{i\in\mathcal{S}_0}(\nabla f(X_i,\theta)-\nabla f(X_i,\theta_j))\right\|$$
$$+\left\|\frac{1}{|\mathcal{S}_0|}\sum_{i\in\mathcal{S}_0}\nabla f(X_i,\theta_j)-\nabla F(\theta_j)\right\|. \tag{36}$$

By Assumption 1,

$$\|\nabla F(\theta)-\nabla F(\theta_j)\|\leq M\|\theta-\theta_j\|\leq M\epsilon. \tag{37}$$

Define event

$$\varepsilon_1=\left\{\sup_{\theta,\theta'\in\Theta:\theta\neq\theta'}\frac{\|\nabla f(X,\theta)-\nabla f(X,\theta')\|}{\|\theta-\theta'\|}\leq M'\right\}. \tag{38}$$

By Assumption 4, $\Pr\{\varepsilon_1\}\geq 1-\frac{\delta_1}{3}$, and on event ε_1,

$$\sup_{\theta\in\Theta_\tau}\left\|\frac{1}{|\mathcal{S}_0|}\sum_{i\in\mathcal{S}_0}(\nabla f(X_i,\theta)-\nabla f(X_i,\theta_j))\right\|$$
$$\leq M'\parallel\theta-\theta_j\parallel\leq M'\epsilon.$$

By triangle's inequality,

$$\left\|\frac{1}{|\mathcal{S}_0|}\sum_{i\in\mathcal{S}_0}\nabla f(X_i,\theta_j)-\nabla F(\theta_j)\right\|$$
$$\leq\left\|\frac{1}{|\mathcal{S}_0|}\sum_{i\in\mathcal{S}_0}\nabla f(X_i,\theta^*)-\nabla F(\theta^*)\right\|$$
$$+\left\|\frac{1}{|\mathcal{S}_0|}\sum_{i\in\mathcal{S}_0}h(X_i,\theta_j)-\mathbb{E}[h(X,\theta_j)]\right\|.$$

Define event

$$\varepsilon_2=\left\{\left\|\frac{1}{|\mathcal{S}_0|}\sum_{i\in\mathcal{S}_0}\nabla f(X_i,\theta^*)-\nabla F(\theta^*)\right\|\geq 2\Delta_1\right\}, \tag{39}$$

and event

$$\mathcal{F}_u=\left\{\left\|\frac{1}{|\mathcal{S}_0|}\sum_{i\in\mathcal{S}_0}h(X_i,\theta_j)-E[h(X,\theta_j)]\right\|\geq 2\tau u\Delta_2\right\}, \tag{40}$$

where

$$\Delta_1=\sqrt{2}\sigma_1\sqrt{\frac{d\log 6+\log(3/\delta_1)}{|\mathcal{S}_0|}}, \tag{41}$$

$\Delta_2=\sqrt{2}\sigma_2\sqrt{(\tau_1+\tau_2)(|\mathcal{S}_0|)}$, with

$$\tau_1=d\log 18+d\log((M\vee M')/\sigma_2),$$
$$\tau_2=\frac{1}{2}d\log(|\mathcal{S}_0|/d)+\log(3/\delta_1)+\log\left(\frac{2r\sigma_2^2\sqrt{|\mathcal{S}_0|}}{\alpha_2\sigma_1}\right). \tag{42}$$

Since $\Delta_1\leq\sigma_1^2/\alpha_1$, by Lemma 2, $\Pr\{\varepsilon_2\}\leq\delta_1/3$. Similarly, by Lemma 3, $\Pr\{\mathcal{F}_u\}\leq\delta_1/(3u^*)$.

In conclusion, it follows that on event $\varepsilon_1\cap\varepsilon_2^c\cap\mathcal{F}_u^c$,

$$\sup_{\theta\in\Theta_\tau}\left\|\frac{1}{|\mathcal{S}_0|}\sum_{i\in\mathcal{S}_0}\nabla f(X_i,\theta)-\nabla F(\theta)\right\|$$
$$\leq(M+M')\epsilon+2\Delta_1+2\Delta_2\tau\leq 4\Delta_2\tau u+2\Delta_1, \tag{43}$$

where the last inequality holds due to $(M\vee M')\epsilon\leq\Delta_2\tau u$. Let

$$\varepsilon=\varepsilon_1\cap\varepsilon_2^c\cap(\cap_{\tau=1}^{u*}\mathcal{F}_u^c). \tag{44}$$

It follows from the union bound, $\Pr\{\varepsilon\}\geq 1-\delta_1$. On event ε, for all $\theta\in\Theta_{u^*}$, there exist a u such that $(u-1)\tau\leq\|\theta-\theta^*\|\leq u\tau$. For $u\geq 2$, $u\leq 2(u-1)$, then

$$\sup_{\theta\in\Theta_r}\left\|\frac{1}{|\mathcal{S}_0|}\sum_{i\in\mathcal{S}_0}\nabla f(X_i,\theta)-\nabla F(\theta)\right\|\leq 8\Delta_2\left\|\theta-\theta^*\right\|+2\Delta_1.$$

For $u=1$, since $\Delta_1\geq\sigma_1\sqrt{d/|\mathcal{S}_0|}$ and $\Delta_2\leq\sigma_2^2/\alpha_2$, by using τ in (34), we get

$$\sup_{\theta\in\Theta_r}\left\|\frac{1}{|\mathcal{S}_0|}\sum_{i\in\mathcal{S}_0}\nabla f(X_i,\theta)-\nabla F(\theta)\right\|\leq 4\Delta_1.$$

Then on event ε, we have

$$\sup_{\theta\in\Theta_r}\left\|\frac{1}{|\mathcal{S}_0|}\sum_{i\in\mathcal{S}_0}\nabla f(X_i,\theta)-\nabla F(\theta)\right\|\le 8\Delta_2\,\|\theta-\theta^*\|+4\Delta_1.$$

As $\Delta_1\le\sigma_1^2/\alpha_1$ and $\Delta_2\le\sigma_2^2/\alpha_2$, then

$$\Pr\{\forall\theta:\|\nabla F(\theta)-\nabla\overline{f}^{(0)}(\theta)\|\le 8\Delta_2\|\theta-\theta^*\|+4\Delta_1\}\ge 1-\delta_1, \tag{45}$$

is proved by the assumption that $\Theta\subset\Theta_r$.

Appendix E
Proof of Theorem 1

Under Proposition 1, fix any $t\ge 1$,

$$\begin{aligned}
&\|\theta_t-\theta^*\|\\
&=\left\|\theta_{t-1}-\eta\left[\sum_{l\in\mathcal{V}_t}w_l q_t^{(l)}(\theta_{t-1})+w_0\nabla\overline{f}^{(0)}(\theta_{t-1})\right]-\theta^*\right\|\\
&=\|\theta_{t-1}-\eta\nabla F(\theta_{t-1})-\theta^*+\eta(\nabla F(\theta_{t-1})-\nabla\overline{f}^{(0)}(\theta_{t-1}))\\
&\quad+\eta\left[\sum_{l\in\mathcal{V}_t}w_l(\overline{f}^{(0)}(\theta_{t-1})-q_t^{(l)}(\theta_{t-1}))\right]\Bigg\|\\
&\le\|\theta_{t-1}-\eta\nabla F(\theta_{t-1})-\theta^*\|+\eta\|\nabla F(\theta_{t-1})-\nabla\overline{f}^{(0)}(\theta_{t-1})\|\\
&\quad+\eta\sum_{l\in\mathcal{V}_t}w_l\left\|\overline{f}^{(0)}(\theta_{t-1})-q_t^{(l)}(\theta_{t-1})\right\| \qquad (46)\\
&\le\|\theta_{t-1}-\eta\nabla F(\theta_{t-1})-\theta^*\|+\eta\|\nabla F(\theta_{t-1})-\nabla\overline{f}^{(0)}(\theta_{t-1})\|\\
&\quad+\eta\xi\sum_{l\in\mathcal{V}_t}w_l\|\overline{f}_t^{(0)}(\theta_{t-1})\|\\
&\le\|\theta_{t-1}-\eta\nabla F(\theta_{t-1})-\theta^*\|+\eta\|\nabla F(\theta_{t-1})-\nabla\overline{f}_t^{(0)}(\theta_{t-1})\|\\
&\quad+\eta\xi\sum_{l\in\mathcal{V}_t}w_l(\|\overline{f}_t^{(0)}(\theta_{t-1})-\nabla F(\theta_{t-1})\|\\
&\quad+\|\nabla F(\theta_{t-1})-\nabla F(\theta^*)\|)\\
&\le\left(\sqrt{1+\eta^2M^2-\eta L}+8\Delta_2\eta+\eta\xi(8\Delta_2+M)\right)\|\theta_{t-1}-\theta^*\|\\
&\quad+(\eta 4\Delta_1+\eta\xi 4\Delta_1). \qquad (47)
\end{aligned}$$

Then

$$\|\theta_t-\theta^*\|\le(1-\rho_1)^t\|\theta_0-\theta^*\|+(4\eta\Delta_1+4\eta\xi\Delta_1)/\rho_1, \tag{48}$$

where

$$\rho_1=1-\left(\sqrt{1+\eta^2M^2-\eta L}+8\Delta_2\eta+\eta\xi(8\Delta_2+M)\right).$$

Appendix F
Proof of Lemma 4

$$\begin{aligned}
&\|G(\theta)-\nabla F(\theta)\|\\
&=\left\|\left(\sum_{j\in\mathcal{H}\cap\mathcal{U}_t}w_j\nabla\overline{f}^{(j)}(\theta)+w_0\nabla\overline{f}^{(0)}(\theta)\right.\right.\\
&\quad\left.\left.+\sum_{j\in\mathcal{A}\cap\mathcal{U}_t}w_jg^{(j)}(\theta)\right)-\nabla F(\theta)\right\|\\
&\le\frac{\sum_{j\in\mathcal{H}\cap\mathcal{U}_t}|\mathcal{S}_j|+|\mathcal{S}_0|}{\sum_{j\in\mathcal{U}_t}|\mathcal{S}_j|+|\mathcal{S}_0|}\|\mathcal{C}_t(\theta)-\nabla F(\theta)\|\\
&\quad+\left\|(\sum_{j\in\mathcal{A}_t\cap\mathcal{U}_t}w_jg^{(j)}(\theta)-\frac{\sum_{j\in\mathcal{A}_t\cap\mathcal{U}_t}|\mathcal{S}_j|}{\sum_{j\in\mathcal{U}_t}|\mathcal{S}_j|+|\mathcal{S}_0|}\nabla F(\theta))\right\|\\
&\le\frac{\sum_{j\in\mathcal{H}\cap\mathcal{U}_t}|\mathcal{S}_j|+|\mathcal{S}_0|}{\sum_{j\in\mathcal{U}_t}|\mathcal{S}_j|+|\mathcal{S}_0|}\|\mathcal{C}_t(\theta)-\nabla F(\theta)\|\\
&\quad+\sum_{j\in\mathcal{A}_t\cap\mathcal{U}_t}w_j\|g^{(j)}(\theta)-\nabla F(\theta)\|.
\end{aligned}$$

Appendix G
Proof of Lemma 5

By triangle inequality,

$$\begin{aligned}
\|\nabla\overline{f}^{(j)}(\theta)-\nabla\overline{f}^{(0)}(\theta)\|&\le\|\nabla\overline{f}^{(j)}(\theta)-\nabla F(\theta)\|\\
&\quad+\|\nabla F(\theta)-\nabla\overline{f}^{(0)}(\theta)\|.
\end{aligned}\tag{49}$$

From Proposition 1, we know that $\|\nabla F(\theta)-\nabla\overline{f}^{(0)}(\theta)\|$ can be universally bounded. Using the same arguments, we have that $\|\nabla F(\theta)-\nabla\overline{f}^{(j)}(\theta)\|$ is universally bounded. In particular, under the same assumption as that of Proposition 1, for any $\delta_1\in(0,1)$

$$\Pr\{\forall\theta:\|\nabla F(\theta)-\nabla\overline{f}^{(j)}(\theta)\|\le 8\Delta_4\|\theta-\theta^*\|+4\Delta_3\}\ge 1-\delta_1, \tag{50}$$

in which

$$\Delta_3=\sqrt{2}\sigma_1\sqrt{(d\log 6+\log(3/\delta_1))/|\mathcal{S}_j|}, \tag{51}$$

and $\Delta_4=\sqrt{2}\sigma_2\sqrt{(\tau_1+\tau_2)/|\mathcal{S}_j|}$, with $\tau_1=d\log 18+d\log((M\vee M')/\sigma_2)$, and $\tau_2=0.5d\,\log(|\mathcal{S}_j|/d)+\log(3/\delta_1)+\log\left(\frac{2r\sigma_2^2\sqrt{|\mathcal{S}_j|}}{\alpha_2\sigma_1}\right)$.

Combining Proposition 1 and equation (50), we know that for each good worker,

$$\begin{aligned}
&\|\nabla\overline{f}^{(j)}(\theta)-\nabla\overline{f}^{(0)}(\theta)\|\\
&\quad\le 8(\Delta_2+\Delta_4)\|\theta-\theta^*\|+4(\Delta_1+\Delta_3),\forall\theta\in\Theta
\end{aligned}\tag{52}$$

with a probability larger than $(1-\delta_1)^2$.

In the following, we provide a lower bound on $\xi\|\nabla\overline{f}^{(0)}(\theta)\|$. By triangle inequality,

$$\xi\|\nabla\overline{f}^{(0)}(\theta)\|\ge\xi\|\nabla F(\theta)\|-\xi\|\nabla F(\theta)-\nabla\overline{f}^{(0)}(\theta)\|. \tag{53}$$

The second term of (53) can be bounded using Proposition 1. Next we bound the first term of (53). Using Assumption 1, we have

$$F(\theta^*) \geq F(\theta) + < \nabla F(\theta), \theta^* - \theta > + \frac{L}{2} \parallel \theta^* - \theta \parallel^2$$
$$\geq F(\theta) - \|\nabla F(\theta)\| \|\theta^* - \theta\| + \frac{L}{2} \parallel \theta^* - \theta \parallel^2 . \quad (54)$$

Since $F(\theta^*) \leq F(\theta)$,

$$-\|\nabla F(\theta)\| \|\theta^* - \theta\| + \frac{L}{2} \parallel \theta^* - \theta \parallel^2 \leq 0, \quad (55)$$

hence,

$$\|\nabla F(\theta)\| \geq \frac{L}{2} \|\theta - \theta^*\|. \quad (56)$$

Plugging (56) and Proposition 1 to (53), we have $\forall \theta \in \Theta$

$$\xi \|\nabla \overline{f}^{(0)}(\theta)\| \geq \frac{L\xi}{2} \|\theta - \theta^*\| - 8\xi\Delta_2 \|\theta - \theta^*\| - 4\xi\Delta_1. \quad (57)$$

with probability larger than $1 - \delta_1$. Then we need to choose value of ξ to guarantee that the right-hand side of (57) will be larger than the right-hand side of (52).

$$8(\Delta_2 + \Delta_4)\|\theta - \theta^*\| + 4(\Delta_1 + \Delta_3)$$
$$\leq 16\Delta_2 \|\theta - \theta^*\| + 8\Delta_1. \quad (58)$$

Since $\xi \leq 1$,

$$(16 + 8\xi)\Delta_2 \|\theta - \theta^*\| + (8 + 4\xi)\Delta_1$$
$$\leq 24\Delta_2 \|\theta - \theta^*\| + 12\Delta_1 \leq \frac{L\xi}{2} \|\theta - \theta^*\|. \quad (59)$$

Since $|\mathcal{S}_0|^{-1/4}$ converges more slowly than $\sqrt{\frac{\log(|\mathcal{S}_0|)}{|\mathcal{S}_0|}}$, we set $\xi = c|\mathcal{S}_0|^{-1/4}$, then we can choose $c = (48\Delta_2 \|\theta - \theta^*\| + 24\Delta_1)|\mathcal{S}_0|^{1/4} / (L\|\theta - \theta^*\|)$, when $\|\theta - \theta^*\| \neq 0$. As the result,

$$\|\nabla \overline{f}^{(j)}(\theta) - \nabla \overline{f}^{(0)}(\theta)\| \leq \xi \|\nabla \overline{f}^{(0)}(\theta)\|, \forall \theta \in \Theta \quad (60)$$

holds with probability $(1 - \delta_1)^2 - \delta_1$.

Appendix H
Proof of Theorem 2

From Assumption 1, Proposition 1, Proposition 2 and Lemma 4, fix any $t \geq 1$, the norm of difference between $G_t(\theta)$ and $\nabla F(\theta)$ is

$$\|G(\theta) - \nabla F(\theta)\|$$
$$\leq \frac{\sum_{j \in \mathcal{H} \cap \mathcal{U}_t} |\mathcal{S}_j| + |\mathcal{S}_0|}{\sum_{j \in \mathcal{U}_t} |\mathcal{S}_j| + |\mathcal{S}_0|} \|C_t(\theta) - \nabla F(\theta)\| + \quad (61)$$
$$+ \sum_{j \in \mathcal{A}_t \cap \mathcal{U}_t} w_j \|\nabla g^{(j)}(\theta) - \nabla F(\theta)\| \quad (62)$$
$$\leq \gamma_2 \|\theta - \theta^*\| + \gamma_1, \quad (63)$$

where

$$\gamma_1 = 4(1 - w_{\max})\Delta_5 + 4w_{\max}\Delta_7, \quad (64)$$

and

$$\gamma_2 = 8(1 - w_{\max})\Delta_6 + 8w_{\max}\Delta_8. \quad (65)$$

with $w_{\max} = \max\{\left(\sum_{j \in \mathcal{B} \cap \mathcal{U}_t} |\mathcal{S}_j|\right) / \left(\sum_{j \in \mathcal{U}_t} |\mathcal{S}_j| + |\mathcal{S}_0|\right)\}$ and $|\mathcal{B} \cap \mathcal{U}_t| = \min\{m - p, p\}$ and $|\mathcal{U}_t| = m - p$. Fix any $t \geq 1$,

$$\|\theta_t - \theta^*\| = \|\theta_{t-1} - \eta G(\theta_{t-1}) - \theta^*\|$$
$$\leq \|\theta_{t-1} - \eta \nabla F(\theta_{t-1}) - \theta^*\| + \eta \|G(\theta_{t-1}) - \nabla F(\theta_{t-1})\|$$
$$\leq \left(\sqrt{1 + \eta^2 M^2 - \eta L} + \eta\gamma_2\right) \|\theta_{t-1} - \theta^*\| + \eta\gamma_1. \quad (66)$$

Then,

$$\|\theta_t - \theta^*\| \leq (1 - \rho_2)^t \|\theta_0 - \theta^*\| + (\eta\gamma_1)/\rho_2, \quad (67)$$

where $\rho_2 = 1 - (\sqrt{1 + \eta^2 M^2 - \eta L} + \eta\gamma_2)$.

References

[1] X. Cao and L. Lai, "Robust distributed gradient descent with arbitrary number of byzantine attackers," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Calgary, AB, Canada, Apr. 2018, pp. 6373–6377.

[2] A. Crotty, A. Galakatos, and T. Kraska, "Tupleware: Distributed machine learning on small clusters," *IEEE Data Eng. Bull.*, vol. 37, no. 3, pp. 63–76, Sep. 2014.

[3] M. Jordan, J. Lee, and Y. Yang, "Communication-efficient distributed statistical inference," *J. Amer. Statistical Assoc.*, vol. 114, no. 526, pp. 668–681, Apr. 2019.

[4] F. Provost and D. Hennessy, "Scaling up: Distributed machine learning with cooperation," in *Proc. of Nat. Conf. Artificial Intell.*, vol. 1, pp. 74–79, Aug. 1996.

[5] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, pp. 1–122, Jan. 2011.

[6] P. Moritz, R. Nishihara, I. Stoica, and M. Jordan, "Sparknet: Training deep networks in spark," May 2016, *arXiv:1511.06051*.

[7] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. Int. Conf. Comput. Statist.*, Aug. 2010, pp. 177–186.

[8] J. Lee, Q. Lin, T. Ma, and T. Yang, "Distributed stochastic variance reduced gradient methods by sampling extra data with replacement," *J. Mach. Learn. Res.*, vol. 18, pp. 4404–4446, Feb. 2017.

[9] D. Friend, R. Thomas, A. MacKenzie, and L. Silva, "Distributed learning and reasoning in cognitive networks: Methods and design decisions," *Cogn. Netw., Towards Self-Aware Netw.*, pp. 223–246, Jul. 2007.

[10] C. Yu, M. van der Schaar, and A. Sayed, "Distributed learning for stochastic generalized nash equilibrium problems," *IEEE Trans. Signal Process.*, vol. 65, no. 15, pp. 3893–3908, Apr. 2017.

[11] P. Mertikopoulos, E. Belmega, R. Negrel, and L. Sanguinetti, "Distributed stochastic optimization via matrix exponential learning," *IEEE Trans. Signal Process.*, vol. 65, no. 9, pp. 2277–2290, May 2017.

[12] C. Tekin and M. van der Schaar, "Distributed online learning via cooperative contextual bandits," *IEEE Trans. Signal Process.*, vol. 63, no. 14, pp. 3700–3714, Jul. 2015.

[13] S. Chouvardas, G. Mileounis, N. Kalouptsidis, and S. Theodoridis, "Greedy sparsity-promoting algorithms for distributed learning," *IEEE Trans. Signal Process.*, vol. 63, no. 6, pp. 1419–1432, Mar. 2015.

[14] B. Swenson, S. Kar, and J. Xavier, "Empirical centroid fictitious play: An approach for distributed learning in multi-agent games," *IEEE Trans. Signal Process.*, vol. 63, no. 15, pp. 3888–3901, Aug. 2015.

[15] S. Marano, V. Matta, and P. Willett, "Nearest-neighbor distributed learning by ordered transmissions," *IEEE Trans. Signal Process.*, vol. 61, no. 21, pp. 5217–5230, Nov. 2013.

[16] Y. Zhang and X. Lin, "Disco: Distributed optimization for self-concordant empirical loss," in *Proc. Int. Conf. Mach. Learn.*, Jul. 2015, pp. 362–370.

[17] P. Richtárik, and M. Takáč, "Distributed coordinate descent method for learning with big data," *J. Mach. Learn. Res.*, vol. 17, pp. 2657–2681, Jan. 2016.

[18] P. Richtárik and M. Takáč, "Parallel coordinate descent methods for big data optimization," *Math. Program.*, vol. 156, pp. 433–484, Mar. 2016.

[19] A. Jochems *et al.*, "Distributed learning: Developing a predictive model based on data from multiple hospitals without data leaving the hospital–a real life proof of concept," *Radiotherapy Oncol.*, vol. 121, pp. 459–467, Dec. 2016.

[20] Y. Chen, L. Su, and J. Xu, "Distributed statistical machine learning in adversarial settings: Byzantine gradient descent," in *Proc. ACM Meas. Anal. Comput. Syst.*, Dec. 2017, pp. 44:1–44:25.

[21] P. Blanchard, E. Mhamdi, R. Guerraoui, and J. Stainer, "Byzantine-tolerant machine learning," Mar. 2017, *arXiv:1703.02757*.

[22] D. Alistarh, Z. Allen-Zhu, and J. Li, "Byzantine stochastic gradient descent," in *Proc. Advances Neural Inform. Process. Syst.*, Dec. 2018, pp. 4613–4623.

[23] C. Xie, O. Koyejo, and I. Gupta, "Phocas: Dimensional byzantine-resilient stochastic gradient descent," May 2018, *arXiv:1805.09682*.

[24] D. Yin, Y. Chen, K. Ramchandran, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," Mar. 2018, *arXiv:1803.01498*.

[25] L. Chen, H. Wang, Z. Charles and D. Papailiopoulos, "DRACO: Byzantine-resilient Distributed Training via Redundant Gradients," in *Proc. Intl. Conf. Mach. Learning*, Jul. 2018, pp. 1–10.

[26] G. Damaskinos, E. Mhamdi, R. Guerraoui, R. Patra, and M. Taziki, "Asynchronous byzantine machine learning," Jul. 2018, *arXiv:1802.07928*.

[27] L. Su and J. Xu, "Securing distributed machine learning in high dimensions," Jun. 2018, *arXiv:1804.10140*.

[28] D. Yin, Y. Chen, K. Ramchandran, and P. Bartlett, "Defending against saddle point attack in byzantine-robust distributed learning," Sep. 2018, *arXiv:1806.05358*.

[29] C. Xie, O. Koyejo, and I. Gupta, "Zeno: Byzantine-suspicious stochastic gradient descent," Sep. 2018, *arXiv:1805.10032*.

[30] L. Huang, A. Joseph, B. Nelson, B. Rubinstein, and J. Tygar, "Adversarial machine learning," in *Proc. ACM Workshop Secur. Artif. Intell.*, Oct. 2011, pp. 43–58.

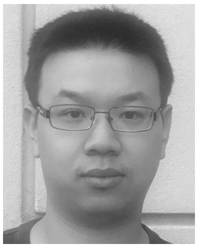
[31] L. Gordon, M. Loeb, W. Lucyshyn, and R. Richardson, "2006 CSI/FBI computer crime and security survey," *Comput. Secur. J.*, vol. 22, no. 3, pp. 1–29, 2006.

[32] Y. LeCun and C. Burges, "Mnist handwritten digit database," AT&T Labs, vol. 2, 2010. [Online]. Available: http://yann.lecun.com/exdb/mnist

[33] A. Krizhevsky, V. Nair, and G. Hinton, "The CIFAR-10 dataset," 2014. [Online]. Available: http://www.cs.toronto.edu/kriz/cifar.html

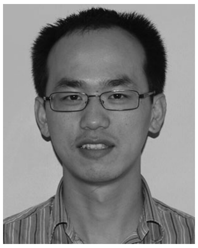
[34] R. Vershynin, "Introduction to the non-asymptotic analysis of random matrices," Nov. 2010, *arXiv:1011.3027*.

[35] J. Wainwright, *High-Dimensional Statistics: A Non-Asymptotic Viewpoint.* Oakland, CA, USA: University of California, 2015.



Xinyang Cao received the B. E. degree from Zhejiang University, Hangzhou, China, in 2014, and the M.S. degree in electrical and computer engineering from University of California, Davis, CA, USA, in 2017.

He is working toward the Ph.D. degree with the Department of Electrical and Computer Engineering, University of California, Davis, CA, USA. His research interests are in machine learning and distributed optimization.



Lifeng Lai (SM'19) received the B.E. and M.E. degrees from Zhejiang University, Hangzhou, China, in 2001 and 2004, respectively, and the Ph.D. degree from The Ohio State University, Columbus, OH, USA, in 2007. He was a Postdoctoral Research Associate with Princeton University from 2007 to 2009, an Assistant Professor with University of Arkansas, Little Rock from 2009 to 2012, and an Assistant Professor with Worcester Polytechnic Institute from 2012 to 2016. Since 2016, he has been an Associate Professor with the University of California, Davis, CA, USA. His research interests include information theory, stochastic signal processing and their applications in wireless communications, security and other related areas.

Dr. Lai was a Distinguished University Fellow of the Ohio State University from 2004 to 2007. He is a co-recipient of the Best Paper Award from the IEEE Global Communications Conference (Globecom) in 2008, the Best Paper Award from IEEE Conference on Communications (ICC) in 2011 and the Best Paper Award from IEEE Smart Grid Communications (SmartGridComm) in 2012. He was the recipient of the National Science Foundation CAREER Award in 2011, and Northrop Young Researcher Award in 2012. He was a Guest Editor for the IEEE Journal on Selected Areas in Communications, Special Issue on Signal Processing Techniques for Wireless Physical Layer Security from 2012 to 2013, and was an Editor for the IEEE Transactions on Wireless Communications from 2013 to 2018. He is currently an Associate Editor for the IEEE Transactions on Information Forensics and Security.