Signal Processing 213 (2023) 109222

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

C-RSA: Byzantine-robust and communication-efficient distributed learning in the non-convex and non-IID regime

Xuechao He [a], Heng Zhu [b], Qing Ling [a,*]

[a] School of Computer Science and Engineering and Guangdong Provincial Key Laboratory of Computational Science, Sun Yat-Sen University, Guangzhou, Guangdong 510006, China

[b] Department of Electrical and Computer Engineering, University of California, San Diego, CA 92093, USA

ARTICLE INFO

Keywords:
Byzantine-robustness
Communication efficiency
Distributed learning
Non-convex
Non-IID

ABSTRACT

The emerging federated learning applications raise challenges of *Byzantine-robustness* and *communication efficiency* in distributed *non-convex* learning over *non-IID* data. To address these issues, in this paper we propose a compressed Byzantine-robust stochastic model aggregation method, abbreviated as C-RSA. C-RSA utilizes robust stochastic model aggregation to obtain *Byzantine-robustness* over *non-IID* data, and compresses the transmitted messages for achieving high *communication efficiency*. Theoretically, we exploit Moreau envelope and proximal point projection as technical tools to analyze the convergence of C-RSA for distributed *non-convex* learning. Extensive numerical experiments are conducted on neural network training tasks to demonstrate the superior performance of C-RSA.

1. Introduction

The past decade has witnessed the explosion of data generated from various distributed devices: mobile phones, intelligent sensors, to name a few. Such a huge amount of distributed data provide both chances and challenges. With them, we are able to train more powerful models than those trained from data of isolated devices. But meanwhile, privacy concerns prevent collecting raw data from the distributed devices to data centers. In this context, federated learning emerges as a favorable tool to balance learning performance and data privacy. The distributed devices (that we term as workers) participating federated learning hold their data privately, and communicate with a central controller (that we term as master node) to collaboratively train a model [1]. A downloaded message from the master node can be the global model, while an uploaded message from a worker can be the local model or the local stochastic gradient [2–4]. Due to the heterogeneous nature of the distributed devices, the data are usually *non-IID* (independent and identically distributed) across the workers. In addition, with the popularization of deep learning, the underlying learning tasks are often formulated as *non-convex* optimization problems. Besides the privacy issue, *communication efficiency* and *robustness* have become main considerations in designing federated learning algorithms [5–7].

Due to the intrinsic bandwidth constraint in federated learning, transmitting messages between the workers and the master node is a bottleneck. To address this issue, one can reduce the communication frequency by allowing the workers to run multiple local iterations before transmissions [8,9], or censoring a number of relatively less informative transmissions [10]. Our focus in this paper is another approach, namely, reducing the communication size through compression, including quantization and sparsification [11–14]. Note that combining these two orthogonal approaches to boost the communication efficiency is also feasible [15].

However, messages received by the master node from the workers are not necessarily trustful. They may encounter packet losses, communication corruptions and even malicious attacks. To characterize such uncertainties, we resort to the classical Byzantine attacks model [16,17]. It assumes that some unreliable workers (that we term as Byzantine workers) can send arbitrarily malicious messages to the master node. The identities of the Byzantine workers are unknown, while we can roughly estimate an upper bound for the number. The Byzantine attacks model describes the worst-case attacks, under which most of the existing distributed learning methods fail [18]. Take the stochastic gradient descent (SGD) method as an example. At each iteration, the master node expects to average the stochastic gradients received from the workers to construct an update direction. Even when only one Byzantine worker exists, it can send a wrong message with large elements such that the iterate blows up, or send a wrong message to nullify the stochastic gradients from the regular workers such that the iterate stops.

To defend against Byzantine attacks, existing Byzantine-robust distributed learning methods often replace the mean aggregation in SGD

* Corresponding author.
E-mail address: lingqing556@mail.sysu.edu.cn (Q. Ling).

https://doi.org/10.1016/j.sigpro.2023.109222
Received 23 April 2023; Received in revised form 12 July 2023; Accepted 8 August 2023
Available online 12 August 2023

with robust aggregation rules. Some robust aggregation rules yield an outlier-tolerating surrogate of the mean, such as median, geometric median and trimmed mean [18–20]. Some others choose a reliable representative stochastic gradient from all the received messages, such as Krum [21]. These methods have guaranteed performance only when the stochastic gradients received from the regular workers are IID, inferring that the regular workers have IID data distribution. Under the IID assumption, Byzantine-robust distributed non-convex learning has also been investigated [22–24]. A class of special Byzantine attacks for non-convex problems are discussed in [25,26], where the Byzantine workers construct fake local minima and the master node must escape from them. However, the IID assumption contradicts with the federated learning setting.

Although federated learning over non-IID data has attracted extensive research interest in these years, existing works generally do not consider Byzantine-robustness [14,27]. When the data distribution is non-IID, the stochastic gradients calculated by the regular workers are also non-IID. Such statistic heterogeneity brings a big challenge to defending against Byzantine attacks. The works of [28–31] adopt robust stochastic model aggregation, which avoids aggregating the stochastic gradients, to address this challenge. The works of [28,29] consider distributed learning, while the works of [30,31] consider decentralized learning. In [32], heterogeneous data are clustered, such that each cluster nearly satisfies the IID assumption. The work of [33] proposes to resample the received stochastic gradients so as to reduce the heterogeneity, before inputting them to robust aggregation.

In this context, our goal is to jointly address the challenges of *Byzantine-robustness* and *communication efficiency* for distributed *non-convex* learning, given that the data are *non-IID*. This is the first work that jointly consider the four factors, to the best of our knowledge. The works of [34–37] develop Byzantine-robust and communication-efficient distributed methods, but are confined to the IID regime. The work of [38] adopts robust stochastic model aggregation [28] to achieve Byzantine-robustness when the data are non-IID, and skips less informative messages [10] to enhance communication efficiency. Nevertheless, the theoretical analysis in [38] still rely on the convex assumption. Besides, the approach of skipping less informative messages to reduce the communication frequency is complementary to our approach of compressing messages to reduce the communication size.

Our contributions are three-fold. First, we propose a compressed robust stochastic model aggregation method, abbreviated as C-RSA, which achieves *Byzantine-robustness* over *non-IID* data. Compared to its uncompressed variant [28], the communication size is significantly reduced and the resulting method is *communication-efficient*. Second, we utilize Moreau envelope and proximal point projection as technical tools to analyze the convergence of C-RSA when the distributed learning problem is *non-convex*. Third, we conduct extensive numerical experiments that show the superior performance of C-RSA in training neural networks under Byzantine attacks.

Compared to the short, conference version [39], in this long, journal version, we have significantly rewritten the content. We have given the complete theoretical analysis of the proposed C-RSA, along with extensive numerical experiments. We have also proved the convergence of the uncompressed version of C-RSA in the non-convex case, which is of independent interest.

2. Problem statement

We consider a distributed learning system in which $K = R + B$ workers are coordinated by one master node. Therein, R workers are regular and the other B are Byzantine. Again, we stress that the master node does not know the number and identities of the Byzantine workers, but only knows K, the number of total workers. For notational convenience, denote $\mathcal{R}$ as the set of regular workers and $\mathcal{B}$ as the set of Byzantine workers. The distributed non-convex learning problem that we solve is

$$\tilde{\mathbf{x}}^* = \arg\min_{\tilde{\mathbf{x}}} \; F(\tilde{\mathbf{x}}), \quad F(\tilde{\mathbf{x}}) \triangleq \sum_{k\in\mathcal{R}} \mathbb{E}[f(\tilde{\mathbf{x}}, \xi_k)] + f_0(\tilde{\mathbf{x}}). \tag{1}$$

In (1), $\tilde{\mathbf{x}} \in \mathbb{R}^d$ stands for the model to learn, $f(\tilde{\mathbf{x}}, \xi_k)$ is the non-convex cost function privately owned by regular worker k, $\xi_k \sim \mathcal{D}_k$ is the random variable for local sample selection, with $\mathcal{D}_k$ being the data distribution of regular worker k, and $f_0(\tilde{\mathbf{x}})$ is the regularization term known to the master node, either convex or non-convex. The expectation is taken over the random variables ξ_k. This regularized expectation risk minimization form appears widely in machine learning applications, including neural network training. We assume the overall data distribution is non-IID, namely, $\mathcal{D}_k$ are different across the regular workers k.

Solving (1) is challenging due to the following reasons. (i) The unknown *Byzantine* workers can send arbitrarily malicious messages to the master node for the sake of biasing the learning process. (ii) The data distribution across the regular workers is *non-IID*, such that it is nontrivial to distinguish the messages from the Byzantine workers and those from the regular ones. (iii) The problem is *non-convex*, which makes developing learning methods with performance guarantee a difficult task. (iv) Messages transmitted between the master node and the distributed workers are through bandwidth-limited channels, which asks for *communication-efficient* learning algorithms.

To realize communication-efficient distributed learning, we propose to compress the messages transmitted between the master node and the workers [11,12]. The following definition characterizes a general compression operator.

Definition 1 (*Compression Operator*)**.** Given a function $C(\cdot) : \mathbb{R}^d \to \mathbb{R}^d$, we call it as a compression operator, if for all $\tilde{\mathbf{x}} \in \mathbb{R}^d$, there exists a constant $\gamma \in (0, 1]$ such that

$$\mathbb{E}\|\tilde{\mathbf{x}} - C(\tilde{\mathbf{x}})\|^2 \le (1-\gamma)\|\tilde{\mathbf{x}}\|^2. \tag{2}$$

Here the expectation is taken over the possible randomness appeared in the compression operator.

This paper mainly considers a compression operator based on rand-l sparsification. For a d-dimensional input vector, the compression operator randomly selects l elements to remain the same and forces the other $d-l$ elements to be zero, where l is an integer between 1 and d. Therefore, the constant $\gamma = \frac{l}{d}$. We focus on rand-l sparsification because of its low complexity of implementation, especially when the model parameter is high-dimensional. In addition, transmitting the output of rand-l sparsification is simple; it suffices to transmit l remaining real values and one random seed that encodes their indices.

Another commonly used sparsification-based compression operator is top-l. For a d-dimensional input vector, the compression operator selects l elements with the largest absolute values to remain the same. Compared with rand-l, top-l retains more information of the input vector. However, it brings extra complexity in sorting the elements, and is hence unfavorable for high-dimensional applications. Besides, transmitting the output of top-l sparsification involves the selected l indices, other than one random seed as in rand-l. Different to rand-l, top-l is a biased compression operator such that its analysis in Byzantine-robust distributed learning is complicated. We empirically test both rand-l and top-l in the numerical experiments, but in theoretical analysis, we focus on rand-l.

3. Algorithm development

For IID data distribution and without considering communication efficiency, distributed SGD methods equipped with Byzantine-robust stochastic gradient aggregation rules have provable performance guarantee in solving (1). In general, these methods operate synchronously. At time t, the master node transmits its current global model $\tilde{\mathbf{x}}^t$ to all

the workers. Upon receiving the global model, each regular worker $k \in \mathcal{R}$ randomly selects a mini-batch of local samples characterized by a random variable $\xi_k^t \sim \mathcal{D}_k$, computes the corresponding stochastic gradient $\nabla f(\tilde{\mathbf{x}}^t, \xi_k^t)$, and then transmits the stochastic gradient to the master node. However, the Byzantine workers do not follow these steps. For each Byzantine worker $k' \in \mathcal{B}$, it generates an arbitrarily malicious message $\mathbf{z}_{k'}^t \in \mathbb{R}^d$, and then transmits $\mathbf{z}_{k'}^t$ to the master node. After collecting all the messages, either trustful or malicious, the master node performs robust aggregation. The output of robust aggregation and $\nabla f_0(\tilde{\mathbf{x}}^t)$, the gradient of the regularization term, are combined to yield a direction for updating $\tilde{\mathbf{x}}^{t+1}$. Nevertheless, the performance of such Byzantine-robust stochastic gradient aggregation significantly depends on the IID assumption. When the data distribution is non-IID, the stochastic gradients of the regular workers are also non-IID, and thus the output of robust aggregation can be very different to the averaged stochastic gradient of the regular workers [28,38].

When the regular workers have non-IID data, Byzantine-robust stochastic model aggregation (RSA) is a viable option [28]. Different from Byzantine-robust distributed SGD where only the master node maintains a global model, RSA lets the master node maintain a local model $\mathbf{x}_0 \in \mathbb{R}^d$ and each regular worker $k \in \mathcal{R}$ maintain a local model $\mathbf{x}_k \in \mathbb{R}^d$. Collect all the local models in a vector

$$\mathbf{x} = [\mathbf{x}_0; \mathbf{x}_{k_1}; \cdots; \mathbf{x}_{k_R}] \in \mathbb{R}^{(R+1)d}, \tag{3}$$

where $k_1, \ldots, k_R$ are regular workers. We have an equivalent form of (1), given by

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \sum_{k\in\mathcal{R}} \mathbb{E}[f(\mathbf{x}_k, \xi_k)] + f_0(\mathbf{x}_0), \quad s.t. \ \ \mathbf{x}_0 = \mathbf{x}_k, \ \forall k \in \mathcal{R}. \tag{4}$$

Imposing the ℓ_1-norm penalization to the consensus constraints $\mathbf{x}_0 = \mathbf{x}_k, \forall k \in \mathcal{R}$, we approximate (4) by

$$\min_{\mathbf{x}} \sum_{k\in\mathcal{R}} \mathbb{E}[f(\mathbf{x}_k, \xi_k)] + f_0(\mathbf{x}_0) + \sum_{k\in\mathcal{R}} \lambda \|\mathbf{x}_k - \mathbf{x}_0\|_1, \tag{5}$$

with $\lambda > 0$ being the penalty parameter. Introducing the ℓ_1-norm penalization enforces the local models to be similar. But unlike the squared ℓ_2-norm penalization, it is less sensitive to possible outliers and thus useful for designing Byzantine-robust algorithms [28].

To solve (5), we apply the stochastic subgradient method. For time t we have

$$\mathbf{x}_0^{t+1} = \mathbf{x}_0^t - \alpha \left[\nabla f_0(\mathbf{x}_0^t) + \lambda \left(\sum_{k\in\mathcal{R}} \operatorname{sign}(\mathbf{x}_0^t - \mathbf{x}_k^t) \right) \right], \tag{6}$$

$$\mathbf{x}_k^{t+1} = \mathbf{x}_k^t - \alpha \left[\nabla f(\mathbf{x}_k^t, \xi_k^t) + \lambda \operatorname{sign}(\mathbf{x}_k^t - \mathbf{x}_0^t) \right], \ \forall k \in \mathcal{R}, \tag{7}$$

with $\alpha > 0$ being the step size. The output of the element-wise sign function $\operatorname{sign}(\cdot)$ is 1 for positive input, -1 for negative input, and 0 for zero input. At presence of the Byzantine workers, (7) does not change but (6) turns to

$$\mathbf{x}_0^{t+1} = \mathbf{x}_0^t - \alpha \left[\nabla f_0(\mathbf{x}_0^t) + \lambda \left(\sum_{k\in\mathcal{R}} \operatorname{sign}(\mathbf{x}_0^t - \mathbf{x}_k^t) \right) + \lambda \left(\sum_{k'\in\mathcal{B}} \operatorname{sign}(\mathbf{x}_0^t - \mathbf{z}_{k'}^t) \right) \right], \tag{8}$$

with $\mathbf{z}_{k'}^t \in \mathbb{R}^d$ standing for the arbitrarily malicious message sent by Byzantine worker $k' \in \mathcal{B}$ at time t. According to (8), any worker, no matter regular or Byzantine, can at most change each element of $\mathbf{x}_0^{t+1}$ by λ. Therefore, RSA is able to effectively control the impact of possible Byzantine attacks.

However, the communication efficiency of RSA is not satisfactory. Although uploading $\operatorname{sign}(\mathbf{x}_0^t - \mathbf{x}_k^t)$ from regular worker k to the master node only needs to transmit d integers,[1] downloading $\mathbf{x}_0^t$ from the master node to a regular worker needs to transmit d real numbers. When the problem dimension d is high, for example, in large-scale neural network training, the communication cost is remarkable.

To achieve communication-efficient learning with Byzantine-robustness, we introduce rand-l sparsification to the downloading. The procedure with top-l sparsification is similar, and we omit it for simplicity. Denote the compression operator as $C(\cdot)$. Instead of sending $\mathbf{x}_0^t$, at time t the master node sends $C(\mathbf{x}_0^t)$ to all the workers. For implementation, it means that the master node sends the remaining l elements of $\mathbf{x}_0^t$ along with a seed to stand for their indices. For convenience, define ω_0^t as the index set of the remaining l elements. Also define $P_{\omega_0^t}(\cdot) : \mathbb{R}^d \to \mathbb{R}^d$ as the projection operator that preserves the input elements whose indices are in ω_0^t and sets the rest as zeros. With these definitions, $P_{\omega_0^t}(\mathbf{x}_0^t) = C(\mathbf{x}_0^t)$. After receiving the compressed $C(\mathbf{x}_0^t)$, each regular worker $k \in \mathcal{R}$ modifies its local model as

$$\mathbf{x}_k^{t+1} = \mathbf{x}_k^t - \alpha \left[\nabla f(\mathbf{x}_k^t, \xi_k^t) + \lambda P_{\omega_0^t}(\operatorname{sign}(\mathbf{x}_k^t - C(\mathbf{x}_0^t))) \right]. \tag{9}$$

For the uploading, the workers no longer send the full sign messages as in RSA. Each regular worker $k \in \mathcal{R}$ transmits $P_{\omega_0^t}(\operatorname{sign}(C(\mathbf{x}_0^t) - \mathbf{x}_k^t))$ to the master node, while each Byzantine worker $k' \in \mathcal{B}$ transmits $P_{\omega_0^t}(\operatorname{sign}(C(\mathbf{x}_0^t) - \mathbf{z}_{k'}^t))$. We emphasize that the Byzantine workers must exactly follow the compression operator; otherwise their identities will be exposed. However, $\mathbf{z}_{k'}^t$ is still arbitrarily malicious. Then, the local model of the master node is updated as

$$\mathbf{x}_0^{t+1} = \mathbf{x}_0^t - \alpha \left[\nabla f_0(\mathbf{x}_0^t) + \lambda \left(\sum_{k\in\mathcal{R}} P_{\omega_0^t}(\operatorname{sign}(C(\mathbf{x}_0^t) - \mathbf{x}_k^t)) \right) + \lambda \left(\sum_{k'\in\mathcal{B}} P_{\omega_0^t}(\operatorname{sign}(C(\mathbf{x}_0^t) - \mathbf{z}_{k'}^t)) \right) \right]. \tag{10}$$

The proposed method is abbreviated as C-RSA, standing for compressed Byzantine-robust stochastic model aggregation. The updates of the master node and the workers are outlined in Algorithm 1 and illustrated in Fig. 1. At time t, the master node transmits to each worker l real values and one seed, and each regular worker only needs to transmit l integers. Since $l < d$, the communication cost is remarkably reduced compared to the uncompressed RSA.

Note that applying other compression rules in C-RSA is also doable. One such example is quantization. We can further reduce the communication frequency, on top of reducing the communication size. These valuable extensions will be investigated in our future work.

C-RSA has low computational complexity. At each time t, calculating the gradient $\nabla f_0(\mathbf{x}_0^t)$ and the stochastic gradients $\nabla f(\mathbf{x}_k^t, \xi_k^t)$ is the same as the other methods. The rand-l sparsification operation and the element-wise sign operation are almost costless. In contrast, the existing Byzantine-robust distributed stochastic gradient aggregation methods for the non-IID data distribution have much higher iteration-wise computational complexity. In [32], the master node performs K-means clustering that has $O(d)$ computational complexity and runs trimmed mean that has $O(dR(R-2B))$ computational complexity. In [33], the computational complexity is $O(R(R-B)\log(R))$ for resampling the received messages and $O(R^2(d+\log R))$ for Krum aggregation. For high-dimensional problems with a large d and/or with a large number of regular workers R, the proposed C-RSA enjoys favorable computational efficiency, in addition to other advantages.

4. Theoretical analysis

This section is devoted to the convergence analysis of C-RSA with rand-l sparsification. Since the ℓ_1-norm penalized cost function in (5) is non-convex and non-smooth, the convergence analysis is challenging — we are unable to utilize common convergence measures, such as distance to the optimal solution/value for convex functions or gradient norm for non-convex smooth functions. To address this issue, we leverage the weak convexity of the cost function, and adopts Moreau envelop and proximal point projection [40] as our main technical tools.

[1] Note that in (5), we can also use ℓ_p-norm penalization with $p > 1$ [28]. However, its subgradient is no longer binary and we have to compress the subgradient for improving the communication efficiency, too. Therefore, we choose ℓ_1-norm penalization in (5).

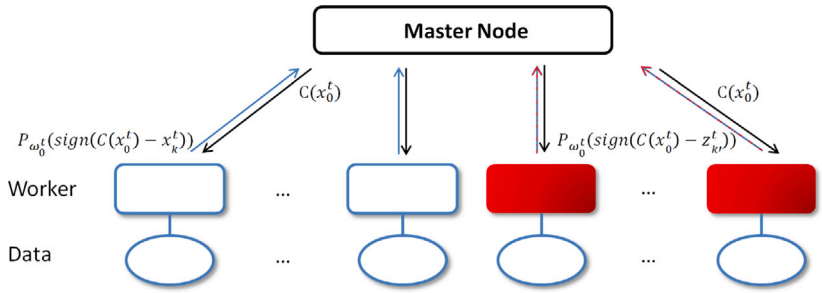

Fig. 1. Schematic diagram of C-RSA. At every time t, the workers download compressed model $C(\mathbf{x}_0^t)$ from the master node. Each regular worker $k \in \mathcal{R}$ uploads the compressed model difference $P_{\omega_0^t}(\text{sign}(C(\mathbf{x}_0^t) - \mathbf{x}_k^t))$, while each Byzantine worker $k' \in \mathcal{B}$ uploads the compressed malicious message in the form of $P_{\omega_0^t}(\text{sign}(C(\mathbf{x}_0^t) - \mathbf{z}_{k'}^t))$.

Algorithm 1 C-RSA at Time t

MASTER NODE

Transmit $C(\mathbf{x}_0^t)$ to all workers
Receive $P_{\omega_0^t}(\text{sign}(C(\mathbf{x}_0^t) - \mathbf{x}_k^t))$ from regular workers and $P_{\omega_0^t}(\text{sign}(C(\mathbf{x}_0^t) - \mathbf{z}_{k'}^t))$ from Byzantine workers
Modify $\mathbf{x}_0^{t+1}$ according to (10), as

$$\mathbf{x}_0^{t+1} = \mathbf{x}_0^t - \alpha \left[\nabla f_0(\mathbf{x}_0^t) + \lambda \left(\sum_{k \in \mathcal{R}} P_{\omega_0^t}(\text{sign}(C(\mathbf{x}_0^t) - \mathbf{x}_k^t)) \right) + \lambda \left(\sum_{k' \in \mathcal{B}} P_{\omega_0^t}(\text{sign}(C(\mathbf{x}_0^t) - \mathbf{z}_{k'}^t)) \right) \right].$$

REGULAR WORKER

Receive $C(\mathbf{x}_0^t)$ from master node
Transmit $P_{\omega_0^t}(\text{sign}(C(\mathbf{x}_0^t) - \mathbf{x}_k^t))$ to master node
Modify $\mathbf{x}_k^{t+1}$ according to (9), as

$$\mathbf{x}_k^{t+1} = \mathbf{x}_k^t - \alpha \left[\nabla f(\mathbf{x}_k^t, \xi_k^t) + \lambda P_{\omega_0^t}(\text{sign}(\mathbf{x}_k^t - C(\mathbf{x}_0^t))) \right].$$

BYZANTINE WORKER

Receive $C(\mathbf{x}_0^t)$ from master node
Create arbitrarily malicious message $\mathbf{z}_{k'}^t$
Transmit $P_{\omega_0^t}(\text{sign}(C(\mathbf{x}_0^t) - \mathbf{z}_{k'}^t))$ to master node

We also provide the convergence analysis of the uncompressed RSA as comparison. This convergence analysis is different to that in [28] since the latter assumes strongly convex cost functions, and thus of independent interest.

4.1. Assumptions

We define $f_k(\mathbf{x}_k) = \mathbb{E}[f(\mathbf{x}_k, \xi_k)]$ as the local cost function of regular worker $k \in \mathcal{R}$, and $\mathbf{g}_k^t = \nabla f(\mathbf{x}_k^t, \xi_k^t)$ as the corresponding stochastic gradient at time t. The following assumptions are made on the local cost functions and the stochastic gradients.

Assumption 1 (*Weak Convexity*)**.** The local cost functions $f_k(\tilde{\mathbf{x}})$ of regular workers k and the regularization term $f_0(\tilde{\mathbf{x}})$ are ρ-weakly convex. Namely, for any $\tilde{\mathbf{x}}, \tilde{\mathbf{y}} \in \mathbb{R}^d$, it holds that

$$f_k(\tilde{\mathbf{y}}) \geq f_k(\tilde{\mathbf{x}}) + \langle \nabla f_k(\tilde{\mathbf{x}}), \tilde{\mathbf{y}} - \tilde{\mathbf{x}} \rangle - \frac{\rho}{2} \|\tilde{\mathbf{y}} - \tilde{\mathbf{x}}\|^2, \quad \forall k \in \mathcal{R} \cup \{0\}. \tag{11}$$

With Assumption 1, the cost functions are not arbitrarily non-convex, which allows us to exploit their structures in the convergence analysis. Actually, weakly convex functions appear in various signal processing applications, including but not limited to phase retrieval, nonlinear least squares and robust principal component analysis [40].

Assumption 2 (*Bounded Gradients*)**.** For any regular worker $k \in \mathcal{R}$, its stochastic gradient at any $\boldsymbol{x}_k^t \in \mathbb{R}^d$ satisfies

$$\mathbb{E} \left\| \boldsymbol{g}_k^t \right\|^2 \leq M^2. \tag{12}$$

For the master node, its gradient at any $\boldsymbol{x}_0 \in \mathbb{R}^d$ satisfies

$$\left\| \nabla f_0(\boldsymbol{x}_0) \right\|^2 \leq M^2. \tag{13}$$

Assumption 2 is common in machine learning. In particular, during training deep neural networks we often use gradient clipping to control the norms of stochastic gradients. Therefore, this assumption is natural.

4.2. Moreau envelope and proximal point projection

Consider a continuous, weakly convex function $h(\tilde{\mathbf{x}})$. Given any point $\tilde{\mathbf{x}} \in \mathbb{R}^d$ and any constant $\beta > 0$, we define

$$h_\beta(\tilde{\mathbf{x}}) := \min_{\tilde{\mathbf{y}}} h(\tilde{\mathbf{y}}) + \frac{1}{2\beta} \|\tilde{\mathbf{y}} - \tilde{\mathbf{x}}\|^2 \tag{14}$$

as its Moreau envelope and

$$\text{prox}_{\beta h}(\tilde{\mathbf{x}}) := \arg\min_{\tilde{\mathbf{y}}} h(\tilde{\mathbf{y}}) + \frac{1}{2\beta} \|\tilde{\mathbf{y}} - \tilde{\mathbf{x}}\|^2 \tag{15}$$

as its proximal point projection. The connection between the Moreau envelope and the proximal point projection is

$$\nabla h_\beta(\tilde{\mathbf{x}}) = \frac{1}{\beta}(\tilde{\mathbf{x}} - \text{prox}_{\beta h}(\tilde{\mathbf{x}})). \tag{16}$$

For any point $\tilde{\mathbf{x}} \in \mathbb{R}^d$, its proximal point $\hat{\mathbf{x}} = \text{prox}_{\beta h}(\tilde{\mathbf{x}})$ satisfies

$$\|\partial h(\hat{\mathbf{x}})\| \leq \left\| \nabla h_\beta(\tilde{\mathbf{x}}) \right\|, \tag{17}$$

$$\|\hat{\mathbf{x}} - \tilde{\mathbf{x}}\| = \beta \left\| \nabla h_\beta(\tilde{\mathbf{x}}) \right\|, \tag{18}$$

with $\partial h(\hat{\mathbf{x}})$ standing for any subgradient of $h(\cdot)$ at $\hat{\mathbf{x}}$. Therefore, if the gradient norm $\left\| \nabla h_\beta(\tilde{\mathbf{x}}) \right\|$ is small, we can reach the following two conclusions. First, by (17), $\|\partial h(\hat{\mathbf{x}})\|$ is small such that $\hat{\mathbf{x}}$ is close to a stationary point of $h(\cdot)$. Second, by (18), $\|\hat{\mathbf{x}} - \tilde{\mathbf{x}}\|$ is small such that $\tilde{\mathbf{x}}$ is close to its proximal point $\hat{\mathbf{x}}$. These two conclusions reveal that if the

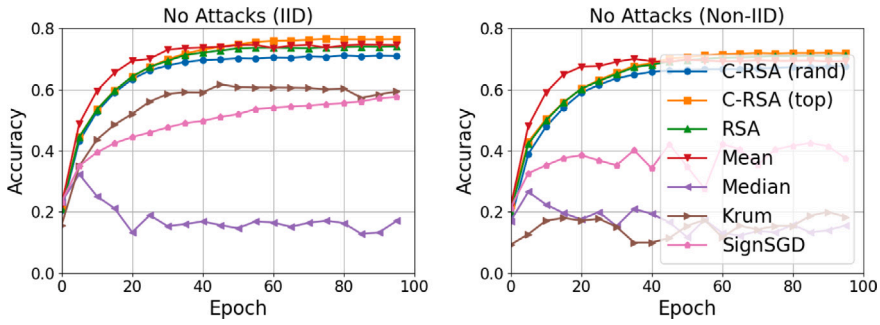

Fig. 2. Performance of C-RSA with rand-l and top-l sparsification, as well as other aggregation rules when there are no Byzantine attacks. The left panel is for the IID case and the right panel is for non-IID case. C-RSA performs constantly well.

gradient norm $\left\|\nabla h_\beta(\tilde{\mathbf{x}})\right\|$ is small, then $\tilde{\mathbf{x}}$ is close to a stationary point of $h(\cdot)$. Consequently, $\left\|\nabla h_\beta(\tilde{\mathbf{x}})\right\|^2$ is a proper measure for the distance between $\tilde{\mathbf{x}}$ and a stationary point of $h(\cdot)$ in the convergence analysis.

4.3. Convergence and asymptotical learning error of non-convex RSA

We start from the analysis of non-convex RSA. Define

$$h(\mathbf{x}) = \sum_{k\in\mathcal{R}} f_k(\mathbf{x}_k) + f_0(\mathbf{x}_0) + \sum_{k\in\mathcal{R}} \lambda \left\|\mathbf{x}_k - \mathbf{x}_0\right\|_1. \tag{19}$$

Observe that $h(\mathbf{x})$ is ρ-weakly convex since $f_k(\mathbf{x}_k)$ and $f_0(\mathbf{x}_0)$ are ρ-weakly convex, while $\left\|\mathbf{x}_k - \mathbf{x}_0\right\|_1$ is convex. We next define $h_{1/\bar{\rho}}(\mathbf{x})$ as the Moreau envelop of $h(\mathbf{x})$, with $\bar{\rho} > \rho$ being a constant.

The following theorem shows that the RSA iterate $\mathbf{x}^t$ converges to a neighborhood of a stationary point of $h_{1/\bar{\rho}}(\cdot)$, and hence to that of $h(\cdot)$ according to the statement given at the end of Section 4.2. The proof is left to Appendix A.

Theorem 1 (*Convergence and Learning Error of Non-convex RSA*). *Suppose that Assumptions 1, 2 hold. Consider the RSA updates (7) and (8) with constant step size* $\alpha = \frac{1}{\sqrt{T}}$. *For any constant* $\bar{\rho} > \rho$, *it holds that*

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left\|\nabla h_{1/\bar{\rho}}(\boldsymbol{x}^t)\right\|^2 \le \frac{\Delta_1}{\sqrt{T}} + \Delta_2, \tag{20}$$

where

$$\Delta_1 := \frac{\bar{\rho}}{(\bar{\rho}-\rho-\epsilon)}\left(h_{1/\bar{\rho}}(\boldsymbol{x}^0) - h_m\right) + \frac{\bar{\rho}^2}{\bar{\rho}-\rho-\epsilon}\left[(R+1)M^2 + (R+K^2)\lambda^2 d\right], \tag{21}$$

$$\Delta_2 := \frac{\bar{\rho}^2\lambda^2 B^2 d}{\epsilon(\bar{\rho}-\rho-\epsilon)}. \tag{22}$$

Therein, $h_m \triangleq \min_{\boldsymbol{x}} h_{1/\bar{\rho}}(\boldsymbol{x})$ *and* ϵ *is an arbitrary constant satisfying* $\bar{\rho}-\rho-\epsilon > 0$.

When the Byzantine workers are absent, $\mathbb{E}\left\|\nabla h_{1/\bar{\rho}}(\mathbf{x}^t)\right\|^2$ asymptotically converges to zero. Nevertheless, at presence of the Byzantine workers, they can transmit arbitrarily malicious messages and lead the iterate to a neighborhood of the stationary point of $h_{1/\bar{\rho}}(\cdot)$. The asymptotical learning error Δ_2 is proportional to B^2, the squared number of Byzantine workers, and also depends on the squared penalty parameter λ^2. When we choose a small λ, then the asymptotical learning error in terms of the running average of $\mathbb{E}\|\nabla h_{1/\bar{\rho}}(\mathbf{x}^t)\|^2$ is small. But on the other hand, it also refers to weak ℓ_1-norm penalties. In this case, the iterate tends to violate the consensus constraints $\mathbf{x}_0 = \mathbf{x}_k$, $k \in \mathcal{R}$.

4.4. Convergence and asymptotical learning error of non-convex C-RSA

In analyzing the convergence of C-RSA, we need to investigate the influence of compressing $\mathbf{x}_0^t$. If the compression operator $C(\cdot)$ is rand-l sparsification, we have

$$\mathbb{E}P_{\omega_0^t}(\mathrm{sign}(\mathbf{x}_k^t - C(\mathbf{x}_0^t))) = \gamma\,\mathrm{sign}(\mathbf{x}_k^t - \mathbf{x}_0^t), \tag{23}$$

$$\mathbb{E}P_{\omega_0^t}(\mathrm{sign}(C(\mathbf{x}_0^t) - \mathbf{x}_k^t)) = \gamma\,\mathrm{sign}(\mathbf{x}_0^t - \mathbf{x}_k^t). \tag{24}$$

Here, we take the expectation with respect to the randomness in $C(\cdot)$ and the compression ratio is $\gamma = \frac{l}{d}$. We remark that it is possible to extend the following analysis to other random compression operators satisfying (23) and (24).

Let us define

$$\tilde{h}(\mathbf{x}) = \sum_{k\in\mathcal{R}} f_k(\mathbf{x}_k) + f_0(\mathbf{x}_0) + \sum_{k\in\mathcal{R}} \gamma\lambda \left\|\mathbf{x}_k - \mathbf{x}_0\right\|_1. \tag{25}$$

Similar to the arguments in analyzing RSA, $\tilde{h}(\mathbf{x})$ is ρ-weakly convex. We define $\tilde{h}_{1/\bar{\rho}}(\mathbf{x})$ as the Moreau envelop of $\tilde{h}(\mathbf{x})$ with $\bar{\rho} > \rho$ being a constant. Note that we define different $h(\mathbf{x})$ and $\tilde{h}(\mathbf{x})$ for the convergence analysis, but the goal of RSA and C-RSA is the same (4).

The following theorem states that the iterate of C-RSA converges to a neighborhood of a stationary point of $\tilde{h}_{1/\bar{\rho}}(\cdot)$, and hence that of $\tilde{h}(\cdot)$. The proof is left to Appendix B.

Theorem 2 (*Convergence and Learning Error of Non-convex C-RSA*). *Suppose that Assumptions 1, 2 hold. Consider the C-RSA updates (9) and (10) with constant step size* $\alpha = \frac{1}{\sqrt{T}}$. *For any constant* $\bar{\rho} > \rho$, *it holds*

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left\|\nabla \tilde{h}_{1/\bar{\rho}}(\boldsymbol{x}^t)\right\|^2 \le \frac{\Delta_1'}{\sqrt{T}} + \Delta_2', \tag{26}$$

where

$$\Delta_1' := \frac{\bar{\rho}}{(\bar{\rho}-\rho-\epsilon)}\left(\tilde{h}_{1/\bar{\rho}}(\boldsymbol{x}^0) - \tilde{h}_m\right) + \frac{\bar{\rho}^2}{\bar{\rho}-\rho-\epsilon}\left[(R+1)M^2 + (R+K^2)\gamma\lambda^2 d\right], \tag{27}$$

$$\Delta_2' := \frac{\bar{\rho}^2\gamma\lambda^2 B^2 d}{(\bar{\rho}-\rho-\epsilon)\epsilon}. \tag{28}$$

Therein, $\tilde{h}_m \triangleq \min_{\boldsymbol{x}} \tilde{h}_{1/\bar{\rho}}(\boldsymbol{x})$ *and* ϵ *is an arbitrary constant satisfying* $\bar{\rho}-\rho-\epsilon > 0$.

Similar to RSA, the asymptotical learning error of C-RSA is also proportional to B^2, the squared number of Byzantine workers, as well as λ^2, the squared penalty parameter. There is a tradeoff between asymptotical learning error and consensus in tuning λ. The impact of λ will be further investigated in the numerical experiments.

It is worth noting that the asymptotical learning error, in terms of the running average of $\mathbb{E}\|\nabla\tilde{h}_{1/\bar{\rho}}(\mathbf{x}^t)\|^2$, is proportional to the compression ratio γ. We explain this seemly counter-intuitive result as follows. Observe that the definition of $\tilde{h}(\mathbf{x})$ is related to γ, while the asymptotical learning error is defined upon such $\tilde{h}(\mathbf{x})$. A small γ means weak coupling between $\mathbf{x}_0$ and $\mathbf{x}_k$ for all $k \in \mathcal{R}$. In this circumstance, a small asymptotical learning error means that the master node and the regular workers reach the neighborhoods of their local stationary points. For

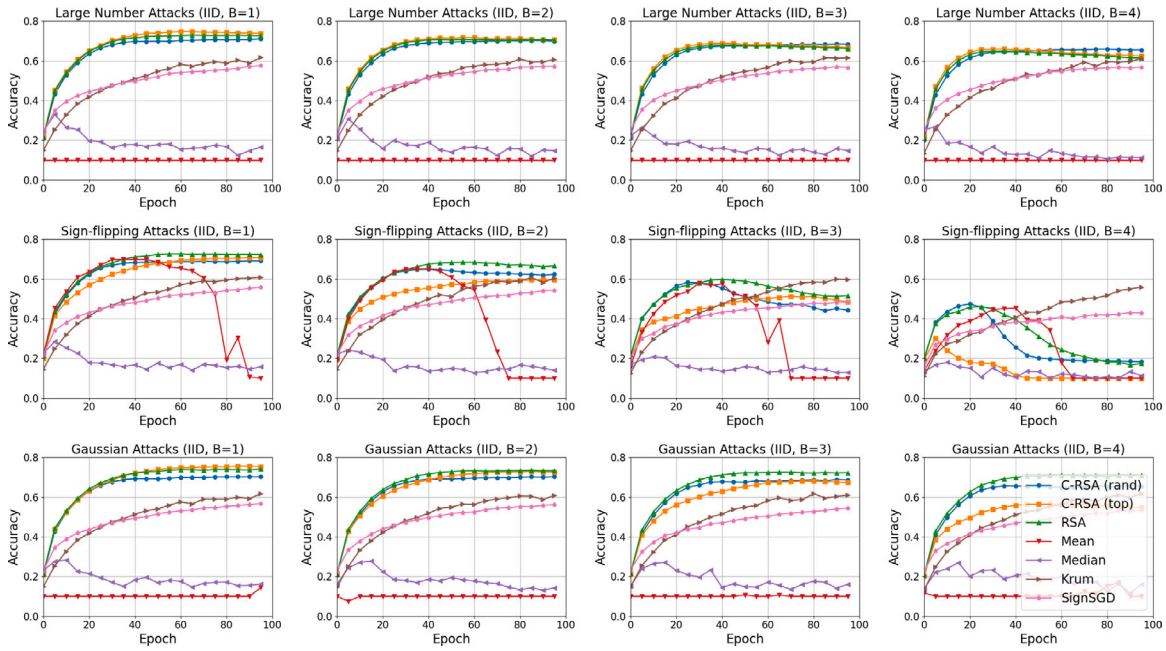

Fig. 3. Performance under different Byzantine attacks for the IID data distribution. There are three attacks from top to bottom: large-number, sign-flipping and Gaussian. The number of Byzantine workers increases from left to right: $B = 1, 2, 3, 4$. RSA and C-RSA are the best in almost all the numerical experiments. The difference between RSA and C-RSA is not significant, showing that compression does not hurt the Byzantine-robustness.

the extreme case of $\gamma = 0$, the master node and the regular workers do not communicate. They eventually converge to the stationary points of their local cost functions, respectively. In the numerical experiments, we will show that a larger compression ratio γ generally leads to higher accuracy in classification problems.

5. Numerical experiments

In this section, we demonstrate the performance of C-RSA with numerical experiments. The task is to train a convolutional neural network (CNN) in a distributed manner. The dataset is on CIFAR10, which has 50,000 training samples and 10,000 testing samples, with each sample being a 32×32 image. These samples belong to 10 categories. The batch size during the training stage is set as 10. The trained CNN is composed of 2 convolutional layers and 3 fully connected layers, having totally 368,052 parameters. We use cross-entropy as the cost function. The regularization term is set as $f_0(\tilde{\mathbf{x}}) := \frac{\mu}{2}\|\tilde{\mathbf{x}}\|^2$ with $\mu = 0.01$.

The distributed learning system consists of one master node and $K = 10$ workers. Both IID and non-IID data distributions are investigated. For the IID data distribution, the training samples are evenly and randomly assigned to all the workers. For the non-IID data distribution, we follow [41] such that half of each category's training samples are assigned to a corresponding worker, while the rest half are evenly and randomly assigned to all the workers.

Three commonly used Byzantine attacks are investigated in the numerical experiments [18].

Large-number attacks. The Byzantine workers change each element of the true messages to 10,000. Note that these attacks are particularly effective to the traditional mean aggregation.

Sign-flipping attacks. The Byzantine workers multiplies each element of the true messages by -1. These attacks are difficult to identify, and may lead the model updates at the master node to wrong directions.

Gaussian attacks. The Byzantine workers modify each element of the true messages to a random variable following the standard normal distribution $\mathcal{N}(0, 1)$. These attacks disturb the convergence of the trained models.

In the numerical experiments, we observe that the elements of the true messages are mostly in the order of $10^{-4} \sim 10^{-2}$. Therefore, these attacks are sufficiently strong.

We compare with four algorithms.

SGD with mean aggregation. This standard SGD implementation is vulnerable to Byzantine attacks.

SGD with median aggregation [19]. It uses element-wise median for stochastic gradient aggregation, and is developed for the IID case.

SGD with Krum aggregation [21]. This algorithm uses stochastic gradient aggregation by selecting one representative stochastic gradient with the smallest summed squared distance to its $K - B - 2$ nearest neighbors. It is also developed for the IID case, and requires to know B, the exact number of Byzantine workers.

SignSGD with majority voting [34,35]. It sums the signs of the stochastic gradients, and then calculates the signs of the sum, which amounts to majority voting.

RSA [28]. It uses stochastic model aggregation and is developed for both the IID and the non-IID data distributions.

In the proposed C-RSA, the compression ratio set as $\gamma = 0.5$ by default, such that the iteration-wise communication cost is roughly reduced by half. We mainly focus on C-RSA with rand-l sparsification, but also show the results of C-RSA with top-l sparsification in some of the numerical experiments. The performance metric is the top-1 classification accuracy of the model at the master node.

The parameters of C-RSA are the same as those in RSA. The step size is set as $\alpha = 0.001$ and the penalty parameter is set as $\lambda = 0.001$ by default. For the other algorithms, all the parameters of are hand-tuned to the best.[2] To initialize, the master node generates the elements of $\mathbf{x}_0^0$ following uniform distributions according to the standard PyTorch model initialization technique, and then sends $\mathbf{x}_0^0$ to the workers such that all $\mathbf{x}_k^0 = \mathbf{x}_0^0$.

We begin with the scenario where no Byzantine attacks are present; see Fig. 2. Not surprisingly, SGD with mean aggregation is the best for

[2] Code available at https://github.com/hexchao/C-RSA.

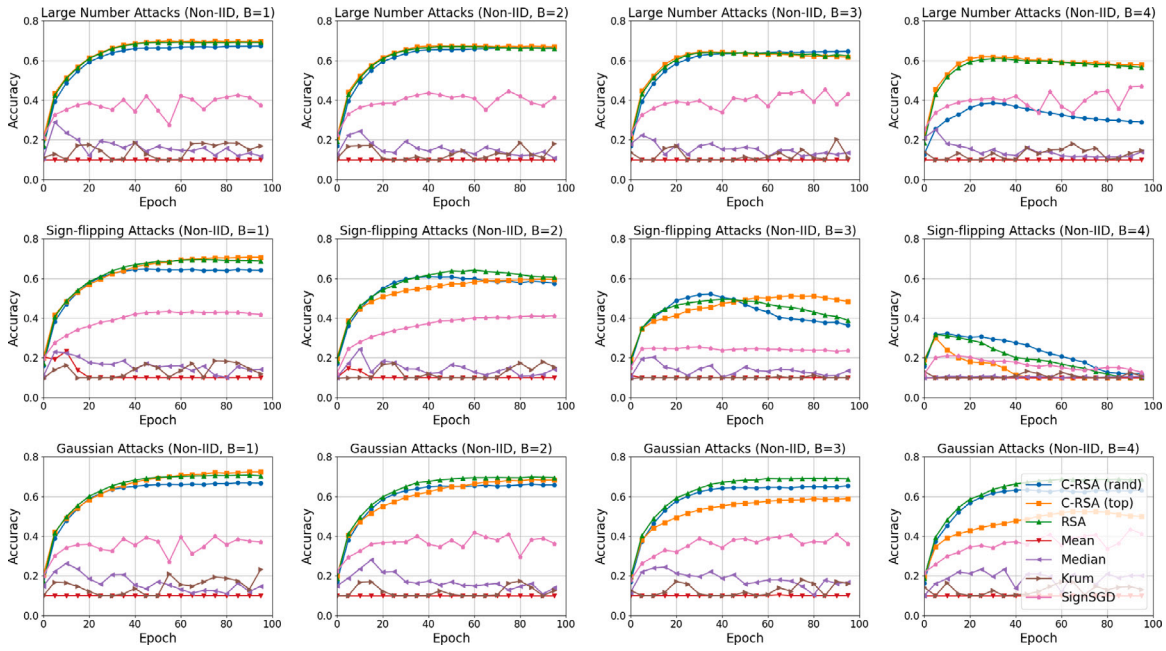

Fig. 4. Performance under different Byzantine attacks for the non-IID data distribution. There are three attacks from top to bottom: large-number, sign-flipping and Gaussian. The number of Byzantine workers increases from left to right: $B = 1, 2, 3, 4$. Mean, median and Krum aggregation rules fail under all attacks. RSA and C-RSA that are developed to handle the non-IID data distribution work well.

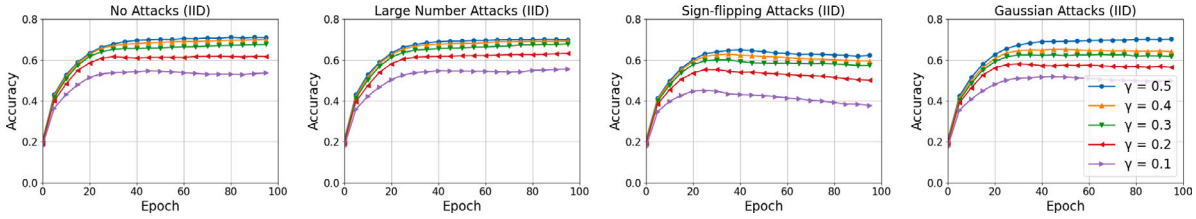

Fig. 5. Impact of compression ratio γ of C-RSA with rand-l sparsification for the IID data distribution. Introducing compression in C-RSA is able to improve the communication efficiency, while still obtain acceptable classification accuracy.

both the IID and the non-IID data distributions. RSA and C-RSA are close to SGD with mean aggregation. However, RSA incurs higher communication cost than C-RSA. SGD with median and Krum aggregation rules and SignSGD fail for the non-IID data distribution, since their aggregation outputs are severely biased when the stochastic gradient inputs are non-IID. For the IID case Krum and SignSGD also do not perform well, while median fails. This phenomenon coincides with the observation in [21,26,38]. It has been conjectured in [38] that median may lead the iterates to unfavorable saddle points when the cost functions are non-convex.

Fig. 3 shows the performance under different Byzantine attacks for the IID data distribution. Even there is only one Byzantine worker, SGD with mean aggregation quickly fails. RSA and C-RSA are the best in almost all the numerical experiments, except for sign-flipping attacks with $B = 3$ and $B = 4$. The difference between RSA and C-RSA is not significant, showing that compression does not hurt the Byzantine-robustness. Krum and SignSGD perform consistently for different numbers of Byzantine workers, as long as $B < \frac{K}{2}$. Nevertheless, we should stress that B must be exactly known in Krum, while RSA and C-RSA do not rely on this prior information.

When the data distribution is non-IID, the results are depicted in Fig. 4. SGD with mean, median and Krum aggregation rules fail under all attacks. SignSGD has better performance, but is still unsatisfactory. RSA and C-RSA that are developed to handle the non-IID data distribution work well. Performance degradation can be observed under sign-flipping attacks when $B = 3$ and $B = 4$, as well as under large number attacks when $B = 4$. This coincides with the theoretical analysis that the worst-case asymptotical learning errors of RSA and C-RSA both increase sharply when B becomes large.

To investigate the influence of compression ratio γ on the performance of C-RSA with rand-l sparsification, we change γ from 0.5 to 0.4, 0.3, 0.2, and 0.1. The number of Byzantine workers is set as $B = 2$. As demonstrated in Figs. 5 and 6, for both the IID and non-IID data distributions, the accuracy monotonically decreases when γ reduces.

The ℓ_1-norm penalty parameter λ plays an important role in C-RSA. We depict its impact on C-RSA with rand-l sparsification with both the IID and non-IID data distributions, shown in Figs. 7 and 8, respectively. With a small λ, the master node and the regular workers tend to minimize their own cost functions and cooperate little. Therefore, the classification accuracy is unsatisfactory even when no Byzantine attacks are present. When there exist Byzantine workers, the asymptotical learning error increases sharply when λ becomes large, according to the theoretical analysis. This theoretical finding is validated by the numerical experiments. Choosing a proper λ helps the distributed learning system obtain the best accuracy.

6. Conclusion

In this paper, we jointly address the following four challenges in distributed learning: (i) There exist Byzantine workers to disturb the learning process; (ii) The communication bandwidths are limited; (iii) The data distribution across the workers is non-IID; (iv) The cost functions are non-convex. Addressing these challenges are of practical importance to secure and efficient distributed learning. We propose

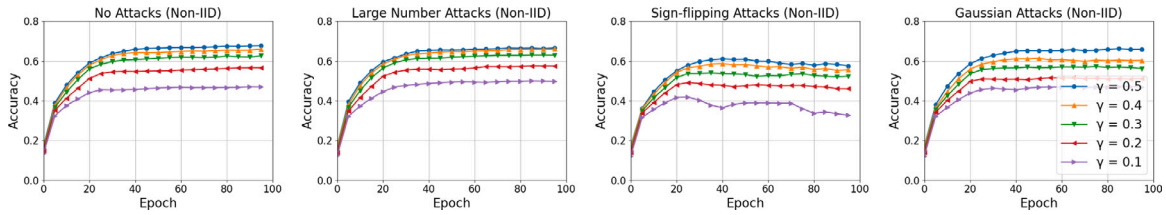

Fig. 6. Impact of compression ratio γ of C-RSA with rand-l sparsification for the non-IID data distribution. Introducing compression in C-RSA is able to improve the communication efficiency, while still obtain acceptable classification accuracy.

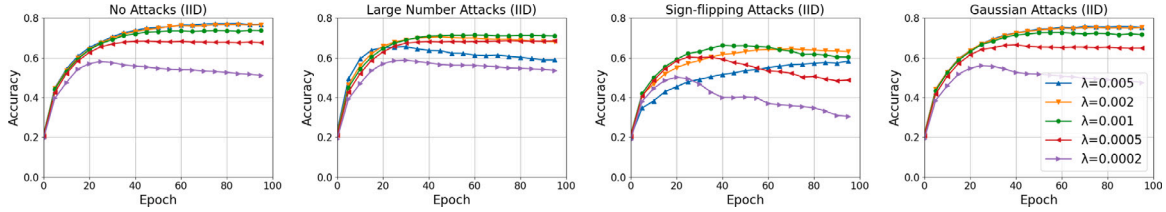

Fig. 7. Performance with different penalty parameters λ of C-RSA for the IID data distribution. Choosing a proper λ helps the distributed learning system obtain the best accuracy.

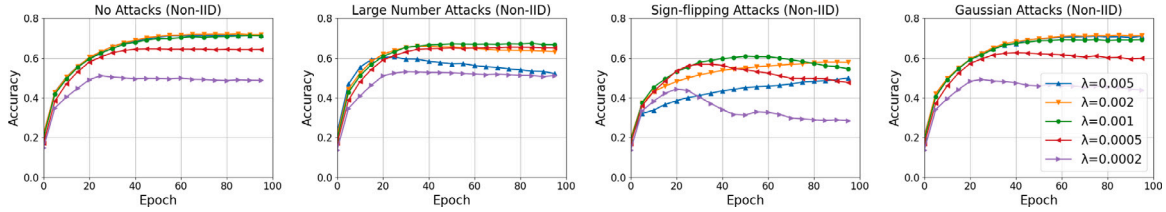

Fig. 8. Performance with different penalty parameters λ of C-RSA for the non-IID data distribution. Choosing a proper λ helps the distributed learning system obtain the best accuracy.

C-RSA, compressed Byzantine-robust stochastic model aggregation, to attain Byzantine-robust and communication-efficient distributed learning over non-IID data. Theoretically, we analyze the convergence of C-RSA with non-convex cost functions, and show that the asymptotical learning error is dependent on the number of Byzantine workers and the penalty parameter. Extensive numerical experiments are conducted to validate the effectiveness of C-RSA, especially for non-IID data.

Our focus in this paper is the compression operator based on rand-l sparsification, which is very easy to implement. We emphasize that other compression rules, including quantization, are also applicable in C-RSA. However, the theoretical analysis needs to be modified for the other compression operators. We will leave them in our future work. Further, we will also investigate the extension to decentralized learning without coordination of a master node.

CRediT authorship contribution statement

Xuechao He: Designed the algorithm, Performed the theoretical analysis, Conducted the numerical experiments, Wrote the original draft. **Heng Zhu:** Performed the theoretical analysis and revised the paper. **Qing Ling:** Organized the research and revised the paper.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The dataset that has been used is publically available.

Acknowledgments

The work of Qing Ling (corresponding author) is supported in part by the National Natural Science Foundation of China under grants 61973324 and 12126610, the Guangdong Basic and Applied Basic Research Foundation under grant 2021B1515020094, and the Guangdong Provincial Key Laboratory of Computational Science, Sun Yat-Sen University, China under grant 2020B1212060032.

Appendix A. Proof of Theorem 1

Proof. First, according to Assumption 2, we obtain two useful inequalities

$$\left\|\nabla f_0(\mathbf{x}_0^t)+\lambda\sum_{k\in\mathcal{R}}\operatorname{sign}(\mathbf{x}_0^t-\mathbf{x}_k^t)+\lambda\sum_{k'\in\mathcal{B}}\operatorname{sign}(\mathbf{x}_0^t-\mathbf{z}_{k'}^t)\right\|^2$$

$$\leq 2\left\|\nabla f_0(\mathbf{x}_0^t)\right\|^2+2\lambda^2\left\|\sum_{k\in\mathcal{R}}\operatorname{sign}(\mathbf{x}_0^t-\mathbf{x}_k^t)+\sum_{k'\in\mathcal{B}}\operatorname{sign}(\mathbf{x}_0^t-\mathbf{z}_{k'}^t)\right\|^2$$

$$\leq 2M^2+2\lambda^2K^2d, \tag{A.1}$$

and

$$\mathbb{E}\left\|\mathbf{g}_k^t+\lambda\operatorname{sign}(\mathbf{x}_k^t-\mathbf{x}_0^t)\right\|^2\leq 2\mathbb{E}\left\|\mathbf{g}_k^t\right\|^2+2\left\|\lambda\operatorname{sign}(\mathbf{x}_k^t-\mathbf{x}_0^t)\right\|^2$$

$$\leq 2M^2+2\lambda^2d. \tag{A.2}$$

In the second inequality of (A.1), the second term is due to bounding the K signs.

Further, because $h(\cdot)$ is ρ-weakly convex, we have

$$h(y)\geq h(\mathbf{x})+\langle\partial h(\mathbf{x}),y-x\rangle-\frac{\rho}{2}\|y-x\|^2, \tag{A.3}$$

with $\partial h(\mathbf{x})$ denoting one subgradient of $h(\mathbf{x})$. Let us consider a particular subgradient

$$\partial h(\mathbf{x}) = \Big[\nabla f_{k_1}(\mathbf{x}_{k_1}) + \lambda \text{sign}(\mathbf{x}_{k_1} - \mathbf{x}_0); \cdots; \nabla f_{k_R}(\mathbf{x}_{k_R}) + \lambda \text{sign}(\mathbf{x}_{k_R} - \mathbf{x}_0); \nabla f_0(\mathbf{x}_0) + \lambda \sum_{k\in\mathcal{R}} \text{sign}(\mathbf{x}_0 - \mathbf{x}_k) \Big]. \tag{A.4}$$

Here $k_1, \ldots, k_R$ are indices of the R regular workers.

Let $\hat{\mathbf{x}}^t := \text{prox}_{h,1/\bar{\rho}}(\mathbf{x}^t)$. According to the definition of Moreau envelop, we have

$$\begin{aligned}
\mathbb{E}[h_{1/\bar{\rho}}(\mathbf{x}^{t+1})] \le & h(\hat{\mathbf{x}}^t) + \frac{\bar{\rho}}{2}\mathbb{E}\left\|\hat{\mathbf{x}}^t - \mathbf{x}^{t+1}\right\|^2 \\
= & h(\hat{\mathbf{x}}^t) + \frac{\bar{\rho}}{2}\mathbb{E}\left\|\hat{\mathbf{x}}^t - \mathbf{x}^t - (\mathbf{x}^{t+1} - \mathbf{x}^t)\right\|^2 \\
= & h(\hat{\mathbf{x}}^t) + \frac{\bar{\rho}}{2}\left\|\hat{\mathbf{x}}^t - \mathbf{x}^t\right\|^2 + \bar{\rho}\mathbb{E}\left\langle \hat{\mathbf{x}}^t - \mathbf{x}^t, \mathbf{x}^t - \mathbf{x}^{t+1}\right\rangle \\
& + \frac{\bar{\rho}}{2}\mathbb{E}\left\|\mathbf{x}^{t+1} - \mathbf{x}^t\right\|^2 \\
= & h_{1/\bar{\rho}}(\mathbf{x}^t) + \bar{\rho}\mathbb{E}\left\langle \hat{\mathbf{x}}^t - \mathbf{x}^t, \mathbf{x}^t - \mathbf{x}^{t+1}\right\rangle + \frac{\bar{\rho}}{2}\mathbb{E}\left\|\mathbf{x}^{t+1} - \mathbf{x}^t\right\|^2.
\end{aligned} \tag{A.5}$$

For the second term at the right-hand side of (A.5), noticing that $\mathbf{x}^t = [\mathbf{x}_0^t; \mathbf{x}_{k_1}^t; \cdots; \mathbf{x}_{k_R}^t]$, we can obtain

$$\begin{aligned}
\mathbb{E}\left\langle \hat{\mathbf{x}}^t - \mathbf{x}^t, \mathbf{x}^t - \mathbf{x}^{t+1}\right\rangle = & \sum_{k\in\mathcal{R}} \mathbb{E}\left\langle \hat{\mathbf{x}}_k^t - \mathbf{x}_k^t, \mathbf{x}_k^t - \mathbf{x}_k^{t+1}\right\rangle + \mathbb{E}\left\langle \hat{\mathbf{x}}_0^t - \mathbf{x}_0^t, \mathbf{x}_0^t - \mathbf{x}_0^{t+1}\right\rangle \\
= & \sum_{k\in\mathcal{R}} \mathbb{E}\left\langle \hat{\mathbf{x}}_k^t - \mathbf{x}_k^t, \alpha\left(\mathbf{g}_k^t + \lambda\text{sign}(\mathbf{x}_k^t - \mathbf{x}_0^t)\right)\right\rangle \\
& + \left\langle \hat{\mathbf{x}}_0^t - \mathbf{x}_0^t, \alpha\left(\nabla f_0(\mathbf{x}_0^t) + \lambda\sum_{k\in\mathcal{R}}\text{sign}(\mathbf{x}_0^t - \mathbf{x}_k^t)\right)\right\rangle \\
& + \left\langle \hat{\mathbf{x}}_0^t - \mathbf{x}_0^t, \alpha\left(\lambda\sum_{k'\in\mathcal{B}}\text{sign}(\mathbf{x}_0^t - \mathbf{z}_{k'}^t)\right)\right\rangle \\
= & \sum_{k\in\mathcal{R}} \alpha\left\langle \hat{\mathbf{x}}_k^t - \mathbf{x}_k^t, \nabla f_k(\mathbf{x}_k^t) + \lambda\text{sign}(\mathbf{x}_k^t - \mathbf{x}_0^t)\right\rangle \\
& + \alpha\left\langle \hat{\mathbf{x}}_0^t - \mathbf{x}_0^t, \nabla f_0(\mathbf{x}_0^t) + \lambda\sum_{k\in\mathcal{R}}\text{sign}(\mathbf{x}_0^t - \mathbf{x}_k^t)\right\rangle \\
& + \alpha\left\langle \hat{\mathbf{x}}_0^t - \mathbf{x}_0^t, \lambda\sum_{k'\in\mathcal{B}}\text{sign}(\mathbf{x}_0^t - \mathbf{z}_{k'}^t)\right\rangle \\
= & \alpha\left\langle \hat{\mathbf{x}}^t - \mathbf{x}^t, \partial h(\mathbf{x}^t)\right\rangle \\
& + \alpha\left\langle \hat{\mathbf{x}}_0^t - \mathbf{x}_0^t, \lambda\sum_{k'\in\mathcal{B}}\text{sign}(\mathbf{x}_0^t - \mathbf{z}_{k'}^t)\right\rangle \\
\le & -\alpha\left(h(\mathbf{x}^t) - h(\hat{\mathbf{x}}^t) - \frac{\rho}{2}\left\|\mathbf{x}^t - \hat{\mathbf{x}}^t\right\|^2\right) \\
& + \alpha\left\langle \hat{\mathbf{x}}_0^t - \mathbf{x}_0^t, \lambda\sum_{k'\in\mathcal{B}}\text{sign}(\mathbf{x}_0^t - \mathbf{z}_{k'}^t)\right\rangle,
\end{aligned} \tag{A.6}$$

where the second equality is due to the updates of $\{\mathbf{x}_k\}_{k\in\mathcal{R}}$ and $\mathbf{x}_0$, the third equality is due to the definition of $\nabla f_k(\mathbf{x}_k^t) = \mathbb{E}\mathbf{g}_k^t = \mathbb{E}\nabla f_k(\mathbf{x}_k, \xi_k^t)$, and the last inequality is due to (A.3).

Because $h(\cdot)$ is a ρ-weakly convex function and $\bar{\rho} > \rho$, we know that the function $h(\mathbf{x}) + \frac{\bar{\rho}}{2}\left\|\mathbf{x}^t - x\right\|^2$ is $(\bar{\rho}-\rho)$-strongly convex with respect to x. Therefore, for the first term at the right-hand side of (A.6), we have

$$\begin{aligned}
h(\mathbf{x}^t) - h(\hat{\mathbf{x}}^t) - \frac{\rho}{2}\left\|\mathbf{x}^t - \hat{\mathbf{x}}^t\right\|^2 = & \left(h(\mathbf{x}^t) + \frac{\bar{\rho}}{2}\left\|\mathbf{x}^t - \mathbf{x}^t\right\|^2\right) \\
& - \left(h(\hat{\mathbf{x}}^t) + \frac{\bar{\rho}}{2}\left\|\mathbf{x}^t - \hat{\mathbf{x}}^t\right\|^2\right) + \frac{\bar{\rho}-\rho}{2}\left\|\mathbf{x}^t - \hat{\mathbf{x}}^t\right\|^2 \\
\ge & (\bar{\rho}-\rho)\left\|\mathbf{x}^t - \hat{\mathbf{x}}^t\right\|^2,
\end{aligned} \tag{A.7}$$

where the second equality is due to (16).

For the second term at the right-hand side of (A.6), for any $\epsilon > 0$ we have

$$\left\langle \hat{\mathbf{x}}_0^t - \mathbf{x}_0^t, \lambda\sum_{k'\in\mathcal{B}}\text{sign}(\mathbf{x}_0^t - \mathbf{z}_{k'}^t)\right\rangle \le \epsilon\left\|\hat{\mathbf{x}}_0^t - \mathbf{x}_0^t\right\|^2 + \frac{\lambda^2}{\epsilon}\left\|\sum_{k'\in\mathcal{B}}\text{sign}(\mathbf{x}_0^t - \mathbf{z}_{k'}^t)\right\|^2$$

$$\le \epsilon\left\|\hat{\mathbf{x}}_0^t - \mathbf{x}_0^t\right\|^2 + \frac{\lambda^2 B^2 d}{\epsilon}. \tag{A.8}$$

With (A.7) and (A.8), for (A.6) we can obtain

$$\begin{aligned}
\mathbb{E}\left\langle \hat{\mathbf{x}}^t - \mathbf{x}^t, \mathbf{x}^t - \mathbf{x}^{t+1}\right\rangle \le & -\alpha(\bar{\rho}-\rho)\left\|\mathbf{x}^t - \hat{\mathbf{x}}^t\right\|^2 + \alpha\epsilon\left\|\hat{\mathbf{x}}_0^t - \mathbf{x}_0^t\right\|^2 + \frac{\alpha\lambda^2 B^2 d}{\epsilon} \\
= & -\alpha(\bar{\rho}-\rho)\sum_{k\in\mathcal{R}}\left\|\mathbf{x}_k^t - \hat{\mathbf{x}}_k^t\right\|^2 - \alpha(\bar{\rho}-\rho)\left\|\mathbf{x}_0^t - \hat{\mathbf{x}}_0^t\right\|^2 \\
& + \alpha\epsilon\left\|\hat{\mathbf{x}}_0^t - \mathbf{x}_0^t\right\|^2 + \frac{\alpha\lambda^2 B^2 d}{\epsilon} \\
= & -\alpha(\bar{\rho}-\rho)\sum_{k\in\mathcal{R}}\left\|\mathbf{x}_k^t - \hat{\mathbf{x}}_k^t\right\|^2 - \alpha(\bar{\rho}-\rho-\epsilon)\left\|\hat{\mathbf{x}}_0^t - \mathbf{x}_0^t\right\|^2 \\
& + \frac{\alpha\lambda^2 B^2 d}{\epsilon} \\
\le & -\alpha(\bar{\rho}-\rho-\epsilon)\left\|\hat{\mathbf{x}}^t - \mathbf{x}^t\right\|^2 + \frac{\alpha\lambda^2 B^2 d}{\epsilon} \\
= & -\frac{\alpha(\bar{\rho}-\rho-\epsilon)}{\bar{\rho}^2}\left\|\nabla h_{1/\bar{\rho}}(\mathbf{x}^t)\right\|^2 + \frac{\alpha\lambda^2 B^2 d}{\epsilon},
\end{aligned} \tag{A.9}$$

where the last equality is due to (16).

For the third term at the right-hand side of (A.5), we have

$$\begin{aligned}
\mathbb{E}\left\|\mathbf{x}^{t+1} - \mathbf{x}^t\right\|^2 = & \sum_{k\in\mathcal{R}}\mathbb{E}\left\|\mathbf{x}_k^{t+1} - \mathbf{x}_k^t\right\|^2 + \mathbb{E}\left\|\mathbf{x}_0^{t+1} - \mathbf{x}_0^t\right\|^2 \\
= & \sum_{k\in\mathcal{R}}\alpha^2\mathbb{E}\left\|\mathbf{g}_k^t + \lambda\text{sign}(\mathbf{x}_k^t - \mathbf{x}_0^t)\right\|^2 \\
& + \alpha^2\left\|\nabla f_0(\mathbf{x}_0^t) + \lambda\sum_{k\in\mathcal{R}}\text{sign}(\mathbf{x}_0^t - \mathbf{x}_k^t) \right. \\
& \left. + \lambda\sum_{k'\in\mathcal{B}}\text{sign}(\mathbf{x}_0^t - \mathbf{z}_{k'}^t)\right\|^2 \\
\le & \alpha^2 R(2M^2 + 2\lambda^2 d) + \alpha^2(2M^2 + 2\lambda^2 K^2 d) \\
= & 2\alpha^2\left((R+1)M^2 + (R+K^2)\lambda^2 d\right),
\end{aligned} \tag{A.10}$$

where the inequality is due to (A.1) and (A.2).

Substituting (A.9) and (A.10) into (A.5), we can obtain

$$\begin{aligned}
\mathbb{E}[h_{1/\bar{\rho}}(\mathbf{x}^{t+1})] \le & h_{1/\bar{\rho}}(\mathbf{x}^t) - \frac{\alpha(\bar{\rho}-\rho-\epsilon)}{\bar{\rho}}\left\|\nabla h_{1/\bar{\rho}}(\mathbf{x}^t)\right\|^2 \\
& + \alpha^2\bar{\rho}\left[(R+1)M^2 + (R+K^2)\lambda^2 d\right] + \frac{\alpha\bar{\rho}\lambda^2 B^2 d}{\epsilon}.
\end{aligned} \tag{A.11}$$

From now on, $\mathbb{E}\left\|\nabla h_{1/\bar{\rho}}(\mathbf{x}^t)\right\|^2$ means expectation with respect to the local stochastic gradients at time $t-1$, and conditioned on all prior randomness.

Therefore, when $\bar{\rho} - \rho - \epsilon > 0$ we can obtain

$$\begin{aligned}
& \mathbb{E}\left\|\nabla h_{1/\bar{\rho}}(\mathbf{x}^t)\right\|^2 \\
\le & \frac{\bar{\rho}}{\alpha(\bar{\rho}-\rho-\epsilon)}\left(\mathbb{E}[h_{1/\bar{\rho}}(\mathbf{x}^t)] - \mathbb{E}[h_{1/\bar{\rho}}(\mathbf{x}^{t+1})]\right) \\
& + \alpha\frac{\bar{\rho}^2}{\bar{\rho}-\rho-\epsilon}\left[(R+1)M^2 + (R+K^2)\lambda^2 d\right] + \frac{\bar{\rho}^2\lambda^2 B^2 d}{\epsilon(\bar{\rho}-\rho-\epsilon)}.
\end{aligned} \tag{A.12}$$

We use a constant step size α, let $h_m = \min_{\mathbf{x}} h_{1/\bar{\rho}}(\mathbf{x})$, and apply telescopic cancellation through $t = 0$ to $T-1$. Then, we have

$$\begin{aligned}
& \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla h_{1/\bar{\rho}}(\mathbf{x}^t)\right\|^2 \\
\le & \frac{\bar{\rho}}{T\alpha(\bar{\rho}-\rho-\epsilon)}\left(\mathbb{E}[h_{1/\bar{\rho}}(\mathbf{x}^0)] - \mathbb{E}[h_{1/\bar{\rho}}(\mathbf{x}^{T-1})]\right) \\
& + \alpha\frac{\bar{\rho}^2}{\bar{\rho}-\rho-\epsilon}\left[(R+1)M^2 + (R+K^2)\lambda^2 d\right] + \frac{\bar{\rho}^2\lambda^2 B^2 d}{\epsilon(\bar{\rho}-\rho-\epsilon)} \\
\le & \frac{\bar{\rho}}{T\alpha(\bar{\rho}-\rho-\epsilon)}\left(\mathbb{E}[h_{1/\bar{\rho}}(\mathbf{x}^0)] - h_m\right) \\
& + \alpha\frac{\bar{\rho}^2}{\bar{\rho}-\rho-\epsilon}\left[(R+1)M^2 + (R+K^2)\lambda^2 d\right] + \frac{\bar{\rho}^2\lambda^2 B^2 d}{\epsilon(\bar{\rho}-\rho-\epsilon)}.
\end{aligned} \tag{A.13}$$

Choosing the step size as $\alpha = \frac{1}{\sqrt{T}}$, we obtain

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla h_{1/\bar{\rho}}(\mathbf{x}^t)\right\|^2 \le \frac{\Delta_1}{\sqrt{T}} + \Delta_2, \tag{A.14}$$

where

$$\Delta_1 := \frac{\bar{\rho}}{(\bar{\rho}-\rho-\epsilon)}\left(h_{1/\bar{\rho}}(\mathbf{x}^0) - h_m\right) + \frac{\bar{\rho}^2}{\bar{\rho}-\rho-\epsilon}\left[(R+1)M^2 + (R+K^2)\lambda^2 d\right], \tag{A.15}$$

$$\Delta_2 := \frac{\bar{\rho}^2\lambda^2 B^2 d}{\epsilon(\bar{\rho}-\rho-\epsilon)}. \tag{A.16}$$

Note that ϵ must be chosen to satisfy $\bar{\rho}-\rho-\epsilon > 0$, for example, $\epsilon = \frac{\bar{\rho}-\rho}{2}$. ■

Appendix B. Proof of Theorem 2

Proof. The proof of C-RSA shares similarity with that of RSA, but we need to handle the influence of compression.

First, according to Assumption 2, we can obtain two useful inequalities

$$\begin{aligned}
&\left\|\nabla f_0(\mathbf{x}_0^t) + \lambda\sum_{k\in\mathcal{R}} P_{\omega_0^t}(\operatorname{sign}(C(\mathbf{x}_0^t)-\mathbf{x}_k^t)) + \lambda\sum_{k'\in\mathcal{B}} P_{\omega_0^t}(\operatorname{sign}(C(\mathbf{x}_0^t)-\mathbf{z}_{k'}^t))\right\|^2 \\
\le& 2\left\|\nabla f_0(\mathbf{x}_0^t)\right\|^2 + 2\lambda^2\left\|\lambda\sum_{k\in\mathcal{R}} P_{\omega_0^t}(\operatorname{sign}(C(\mathbf{x}_0^t)-\mathbf{x}_k^t)) + \lambda\sum_{k'\in\mathcal{B}} P_{\omega_0^t}(\operatorname{sign}(C(\mathbf{x}_0^t)-\mathbf{z}_{k'}^t))\right\|^2 \\
\le& 2M^2 + 2\gamma\lambda^2K^2d.
\end{aligned} \tag{B.1}$$

and

$$\begin{aligned}
\mathbb{E}\left\|\mathbf{g}_k^t + \lambda P_{\omega_0^t}(\operatorname{sign}(\mathbf{x}_k^t - C(\mathbf{x}_0^t)))\right\|^2 \le& 2\mathbb{E}\left\|\mathbf{g}_k^t\right\|^2 + 2\left\|\lambda P_{\omega_0^t}(\operatorname{sign}(\mathbf{x}_k^t - C(\mathbf{x}_0^t)))\right\|^2 \\
\le& 2M^2 + 2\gamma\lambda^2 d,
\end{aligned} \tag{B.2}$$

The first term in the second inequality of (B.1) is due to bounding the K signs.

Further, the fact that $\tilde{h}(\cdot)$ is ρ-weakly convex for any $k \in \mathcal{R}$ yields

$$\tilde{h}(\mathbf{y}) \ge \tilde{h}(\mathbf{x}) + \left\langle \partial\tilde{h}(\mathbf{x}), \mathbf{y}-\mathbf{x}\right\rangle - \frac{\rho}{2}\left\|\mathbf{y}-\mathbf{x}\right\|^2, \tag{B.3}$$

with $\partial\tilde{h}(\mathbf{x})$ denoting one subgradient of $\tilde{h}(\mathbf{x})$.

Let us consider a particular subgradient

$$\begin{aligned}
\partial\tilde{h}(\mathbf{x}) = \Big[&\nabla f_{k_1}(\mathbf{x}_{k_1}) + \gamma\lambda\operatorname{sign}(\mathbf{x}_{k_1}-\mathbf{x}_0); \cdots; \\
&\nabla f_{k_R}(\mathbf{x}_{k_R}) + \gamma\lambda\operatorname{sign}(\mathbf{x}_{k_R}-\mathbf{x}_0); \nabla f_0(\mathbf{x}_0) + \gamma\lambda\sum_{k\in\mathcal{R}}\operatorname{sign}(\mathbf{x}_0-\mathbf{x}_k)\Big].
\end{aligned} \tag{B.4}$$

Here $k_1, \ldots, k_R$ are indices of the R regular workers.

Let $\hat{\mathbf{x}}^t := \operatorname{prox}_{\tilde{h},1/\bar{\rho}}(\mathbf{x}^t)$. According to the definition of Moreau envelop, we have

$$\begin{aligned}
\mathbb{E}[\tilde{h}_{1/\bar{\rho}}(\mathbf{x}^{t+1})] \le& \tilde{h}(\hat{\mathbf{x}}^t) + \frac{\bar{\rho}}{2}\mathbb{E}\left\|\hat{\mathbf{x}}^t - \mathbf{x}^{t+1}\right\|^2 \\
=& \tilde{h}(\hat{\mathbf{x}}^t) + \frac{\bar{\rho}}{2}\mathbb{E}\left\|\hat{\mathbf{x}}^t - \mathbf{x}^t - (\mathbf{x}^{t+1}-\mathbf{x}^t)\right\|^2
\end{aligned} \tag{B.5}$$

$$\begin{aligned}
=& \tilde{h}(\hat{\mathbf{x}}^t) + \frac{\bar{\rho}}{2}\left\|\hat{\mathbf{x}}^t - \mathbf{x}^t\right\|^2 + \bar{\rho}\mathbb{E}\left\langle\hat{\mathbf{x}}^t - \mathbf{x}^t, \mathbf{x}^t - \mathbf{x}^{t+1}\right\rangle + \frac{\bar{\rho}}{2}\mathbb{E}\left\|\mathbf{x}^{t+1}-\mathbf{x}^t\right\|^2 \\
=& \tilde{h}_{1/\bar{\rho}}(\mathbf{x}^t) + \bar{\rho}\mathbb{E}\left\langle\hat{\mathbf{x}}^t - \mathbf{x}^t, \mathbf{x}^t - \mathbf{x}^{t+1}\right\rangle + \frac{\bar{\rho}}{2}\mathbb{E}\left\|\mathbf{x}^{t+1}-\mathbf{x}^t\right\|^2.
\end{aligned}$$

For the second term at the right-hand side of (B.5), noticing that $\mathbf{x}^t = [\mathbf{x}_0^t; \mathbf{x}_{k_1}^t; \cdots; \mathbf{x}_{k_R}^t]$, we can obtain

$$\begin{aligned}
&\mathbb{E}\left\langle\hat{\mathbf{x}}^t - \mathbf{x}^t, \mathbf{x}^t - \mathbf{x}^{t+1}\right\rangle \\
=& \sum_{k\in\mathcal{R}}\mathbb{E}\left\langle\hat{\mathbf{x}}_k^t - \mathbf{x}_k^t, \mathbf{x}_k^t - \mathbf{x}_k^{t+1}\right\rangle + \mathbb{E}\left\langle\hat{\mathbf{x}}_0^t - \mathbf{x}_0^t, \mathbf{x}_0^t - \mathbf{x}_0^{t+1}\right\rangle \\
=& \sum_{k\in\mathcal{R}}\mathbb{E}\left\langle\hat{\mathbf{x}}_k^t - \mathbf{x}_k^t, \alpha\left(\mathbf{g}_k^t + \lambda P_{\omega_0^t}(\operatorname{sign}(\mathbf{x}_k^t - C(\mathbf{x}_0^t)))\right)\right\rangle \\
&+ \left\langle\hat{\mathbf{x}}_0^t - \mathbf{x}_0^t, \alpha\left(\nabla f_0(\mathbf{x}_0^t) + \lambda\sum_{k\in\mathcal{R}} P_{\omega_0^t}(\operatorname{sign}(C(\mathbf{x}_0^t) - \mathbf{x}_k^t))\right)\right\rangle \\
&+ \left\langle\hat{\mathbf{x}}_0^t - \mathbf{x}_0^t, \alpha\left(\lambda\sum_{k'\in\mathcal{B}} P_{\omega_0^t}(\operatorname{sign}(C(\mathbf{x}_0^t) - \mathbf{z}_{k'}^t))\right)\right\rangle \\
=& \sum_{k\in\mathcal{R}}\alpha\left\langle\hat{\mathbf{x}}_k^t - \mathbf{x}_k^t, \nabla f_k(\mathbf{x}_k^t) + \gamma\lambda\operatorname{sign}(\mathbf{x}_k^t - \mathbf{x}_0^t)\right\rangle \\
&+ \alpha\left\langle\hat{\mathbf{x}}_0^t - \mathbf{x}_0^t, \nabla f_0(\mathbf{x}_0^t) + \gamma\lambda\sum_{k\in\mathcal{R}}\operatorname{sign}(\mathbf{x}_0^t - \mathbf{x}_k^t)\right\rangle \\
&+ \alpha\left\langle\hat{\mathbf{x}}_0^t - \mathbf{x}_0^t, \lambda\sum_{k'\in\mathcal{B}} P_{\omega_0^t}(\operatorname{sign}(C(\mathbf{x}_0^t) - \mathbf{z}_{k'}^t))\right\rangle \\
=& \alpha\left\langle\hat{\mathbf{x}}^t - \mathbf{x}^t, \partial\tilde{h}(\mathbf{x}^t)\right\rangle + \alpha\left\langle\hat{\mathbf{x}}_0^t - \mathbf{x}_0^t, \lambda\sum_{k'\in\mathcal{B}} P_{\omega_0^t}(\operatorname{sign}(C(\mathbf{x}_0^t) - \mathbf{z}_{k'}^t))\right\rangle \\
\le& -\alpha\left(\tilde{h}(\mathbf{x}^t) - \tilde{h}(\hat{\mathbf{x}}^t) - \frac{\rho}{2}\left\|\mathbf{x}^t - \hat{\mathbf{x}}^t\right\|^2\right) \\
&+ \alpha\left\langle\hat{\mathbf{x}}_0^t - \mathbf{x}_0^t, \lambda\sum_{k'\in\mathcal{B}} P_{\omega_0^t}(\operatorname{sign}(C(\mathbf{x}_0^t) - \mathbf{z}_{k'}^t))\right\rangle,
\end{aligned} \tag{B.6}$$

where the second equality is due to the updates of $\{\mathbf{x}_k\}_{k\in\mathcal{R}}$ and $\mathbf{x}_0$, the third equality is due to the definition of $\nabla f_k(\mathbf{x}_k^t) = \mathbb{E}\mathbf{g}_k^t = \mathbb{E}\nabla f_k(\mathbf{x}_k, \xi_k^t)$ as well as (23) and (24), while the last inequality is due to (B.3).

Because $\tilde{h}(\cdot)$ is a ρ-weakly convex function and $\bar{\rho} > \rho$, we know that the function $\tilde{h}(\mathbf{x}) + \frac{\bar{\rho}}{2}\left\|\mathbf{x}^t - \mathbf{x}\right\|^2$ is $(\bar{\rho}-\rho)$-strongly convex with respect to $\mathbf{x}$.

Therefore, for the first term at the right-hand side of (B.6), we have

$$\begin{aligned}
\tilde{h}(\mathbf{x}^t) - \tilde{h}(\hat{\mathbf{x}}^t) - \frac{\rho}{2}\left\|\mathbf{x}^t - \hat{\mathbf{x}}^t\right\|^2 =& \left(\tilde{h}(\mathbf{x}^t) + \frac{\bar{\rho}}{2}\left\|\mathbf{x}^t - \mathbf{x}^t\right\|^2\right) \\
&- \left(\tilde{h}(\hat{\mathbf{x}}^t) + \frac{\bar{\rho}}{2}\left\|\mathbf{x}^t - \hat{\mathbf{x}}^t\right\|^2\right) \\
&+ \frac{\bar{\rho}-\rho}{2}\left\|\mathbf{x}^t - \hat{\mathbf{x}}^t\right\|^2 \\
\ge& (\bar{\rho}-\rho)\left\|\mathbf{x}^t - \hat{\mathbf{x}}^t\right\|^2,
\end{aligned} \tag{B.7}$$

where the second equality is due to (16).

For the second term at the right-hand side of (B.6), for any $\epsilon > 0$ we have

$$\begin{aligned}
&\mathbb{E}\left\langle\hat{\mathbf{x}}_0^t - \mathbf{x}_0^t, \lambda\sum_{k'\in\mathcal{B}} P_{\omega_0^t}(\operatorname{sign}(C(\mathbf{x}_0^t) - \mathbf{z}_{k'}^t))\right\rangle \\
\le& \epsilon\left\|\hat{\mathbf{x}}_0^t - \mathbf{x}_0^t\right\|^2 + \frac{\lambda^2}{\epsilon}\left\|\sum_{k'\in\mathcal{B}} P_{\omega_0^t}(\operatorname{sign}(C(\mathbf{x}_0^t) - \mathbf{z}_{k'}^t))\right\|^2 \\
\le& \epsilon\left\|\hat{\mathbf{x}}_0^t - \mathbf{x}_0^t\right\|^2 + \frac{\gamma\lambda^2B^2d}{\epsilon},
\end{aligned} \tag{B.8}$$

where the last inequality is due to the fact that only $l = \gamma d$ dimensions are remaining after compression.

With (B.7) and (B.8), for (B.6) we can obtain

$$\mathbb{E}\left\langle\hat{\mathbf{x}}^t - \mathbf{x}^t, \mathbf{x}^t - \mathbf{x}^{t+1}\right\rangle \le -\alpha(\bar{\rho}-\rho)\left\|\mathbf{x}^t - \hat{\mathbf{x}}^t\right\|^2 + \alpha\epsilon\left\|\hat{\mathbf{x}}_0^t - \mathbf{x}_0^t\right\|^2 + \frac{\alpha\gamma\lambda^2B^2d}{\epsilon} \tag{B.9}$$

$$\begin{aligned}
=& -\alpha(\bar{\rho}-\rho)\sum_{k\in\mathcal{R}}\left\|\mathbf{x}_k^t - \hat{\mathbf{x}}_k^t\right\|^2 - \alpha(\bar{\rho}-\rho)\left\|\mathbf{x}_0^t - \hat{\mathbf{x}}_0^t\right\|^2 \\
&+ \alpha\epsilon\left\|\hat{\mathbf{x}}_0^t - \mathbf{x}_0^t\right\|^2 + \frac{\alpha\gamma\lambda^2B^2d}{\epsilon} \\
=& -\alpha(\bar{\rho}-\rho)\sum_{k\in\mathcal{R}}\left\|\mathbf{x}_k^t - \hat{\mathbf{x}}_k^t\right\|^2 - \alpha(\bar{\rho}-\rho-\epsilon)\left\|\hat{\mathbf{x}}_0^t - \mathbf{x}_0^t\right\|^2 \\
&+ \frac{\alpha\gamma\lambda^2B^2d}{\epsilon}
\end{aligned}$$

$$\leq -\alpha(\bar{\rho}-\rho-\epsilon)\left\|\hat{\mathbf{x}}^t-\mathbf{x}^t\right\|^2+\frac{\alpha\gamma\lambda^2B^2d}{\epsilon}$$
$$=-\frac{\alpha(\bar{\rho}-\rho-\epsilon)}{\bar{\rho}^2}\left\|\nabla\tilde{h}_{1/\bar{\rho}}(\mathbf{x}^t)\right\|^2+\frac{\alpha\gamma\lambda^2B^2d}{\epsilon},$$

where the last equality is due to (16).

For the third term at the right-hand side of (B.5), we have

$$\begin{aligned}\mathbb{E}\left\|\mathbf{x}^{t+1}-\mathbf{x}^t\right\|^2&=\sum_{k\in\mathcal{R}}\mathbb{E}\left\|\mathbf{x}_k^{t+1}-\mathbf{x}_k^t\right\|^2+\mathbb{E}\left\|\mathbf{x}_0^{t+1}-\mathbf{x}_0^t\right\|^2\\&=\sum_{k\in\mathcal{R}}\alpha^2\mathbb{E}\left\|\mathbf{g}_k^t+\lambda P_{\omega_0^t}(\mathrm{sign}(\mathbf{x}_k^t-C(\mathbf{x}_0^t)))\right\|^2\\&\quad+\alpha^2\Bigg\|\nabla f_0(\mathbf{x}_0^t)+\lambda\sum_{k\in\mathcal{R}}P_{\omega_0^t}(\mathrm{sign}(C(\mathbf{x}_0^t)-\mathbf{x}_k^t))\\&\quad+\lambda\sum_{k'\in\mathcal{B}}P_{\omega_0^t}(\mathrm{sign}(C(\mathbf{x}_0^t)-\mathbf{z}_{k'}^t))\Bigg\|^2\\&\leq\alpha^2R(2M^2+2\gamma\lambda^2d)+\alpha^2(2M^2+2\gamma\lambda^2K^2d)\\&=2\alpha^2\left((R+1)M^2+(R+K^2)\gamma\lambda^2d\right),\end{aligned}\tag{B.10}$$

where the inequality is due to (B.1) and (B.2).

Substituting (B.9) and (B.10) into (B.5), we can obtain

$$\mathbb{E}[\tilde{h}_{1/\bar{\rho}}(\mathbf{x}^{t+1})]\leq\tilde{h}_{1/\bar{\rho}}(\mathbf{x}^t)-\frac{\alpha(\bar{\rho}-\rho-\epsilon)}{\bar{\rho}}\left\|\nabla\tilde{h}_{1/\bar{\rho}}(\mathbf{x}^t)\right\|^2$$
$$+\alpha^2\bar{\rho}\left[(R+1)M^2+(R+K^2)\gamma\lambda^2d\right]+\frac{\alpha\bar{\rho}\gamma\lambda^2B^2d}{\epsilon}.\tag{B.11}$$

From now on, $\mathbb{E}\|\nabla\tilde{h}_{1/\bar{\rho}}(\mathbf{x}^t)\|^2$ means expectation with respect to the local stochastic gradients and the rand-l sparsification-based compression at time $t-1$, and conditioned on all prior randomness. Therefore, when $\bar{\rho}-\rho-\epsilon>0$ we obtain

$$\mathbb{E}\left\|\nabla\tilde{h}_{1/\bar{\rho}}(\mathbf{x}^t)\right\|^2\leq\frac{\bar{\rho}}{\alpha(\bar{\rho}-\rho-\epsilon)}\left(\mathbb{E}[\tilde{h}_{1/\bar{\rho}}(\mathbf{x}^t)]-\mathbb{E}[\tilde{h}_{1/\bar{\rho}}(\mathbf{x}^{t+1})]\right)$$
$$+\alpha\frac{\bar{\rho}^2}{\bar{\rho}-\rho-\epsilon}\left[(R+1)M^2+(R+K^2)\gamma\lambda^2d\right]+\frac{\bar{\rho}^2\gamma\lambda^2B^2d}{\epsilon(\bar{\rho}-\rho-\epsilon)}.\tag{B.12}$$

We use a constant step size α, let $\tilde{h}_m=\min_{\mathbf{x}}\tilde{h}_{1/\bar{\rho}}(\mathbf{x})$, and apply telescopic cancellation through $t=0$ to $T-1$. Then, we have

$$\begin{aligned}&\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla\tilde{h}_{1/\bar{\rho}}(\mathbf{x}^t)\right\|^2\\&\leq\frac{\bar{\rho}}{T\alpha(\bar{\rho}-\rho-\epsilon)}\left(\mathbb{E}[\tilde{h}_{1/\bar{\rho}}(\mathbf{x}^0)]-\mathbb{E}[\tilde{h}_{1/\bar{\rho}}(\mathbf{x}^{T-1})]\right)\\&\quad+\alpha\frac{\bar{\rho}^2}{\bar{\rho}-\rho-\epsilon}\left[(R+1)M^2+(R+K^2)\lambda^2d\right]+\frac{\bar{\rho}^2\lambda^2\gamma B^2d}{\epsilon(\bar{\rho}-\rho-\epsilon)}\\&\leq\frac{\bar{\rho}}{T\alpha(\bar{\rho}-\rho-\epsilon)}\left(\mathbb{E}[\tilde{h}_{1/\bar{\rho}}(\mathbf{x}^0)]-\tilde{h}_m\right)\\&\quad+\alpha\frac{\bar{\rho}^2}{\bar{\rho}-\rho-\epsilon}\left[(R+1)M^2+(R+K^2)\lambda^2d\right]+\frac{\bar{\rho}^2\lambda^2\gamma B^2d}{\epsilon(\bar{\rho}-\rho-\epsilon)}.\end{aligned}\tag{B.13}$$

Choosing the step size as $\alpha=\frac{1}{\sqrt{T}}$, we obtain

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla h_{1/\bar{\rho}}(\mathbf{x}^t)\right\|^2\leq\frac{\Delta_1'}{\sqrt{T}}+\Delta_2',\tag{B.14}$$

where

$$\Delta_1':=\frac{\bar{\rho}}{(\bar{\rho}-\rho-\epsilon)}(\tilde{h}_{1/\bar{\rho}}(\mathbf{x}^0)-\tilde{h}_m)$$
$$+\frac{\bar{\rho}^2}{\bar{\rho}-\rho-\epsilon}\left[(R+1)M^2+(R+K^2)\gamma\lambda^2d\right],\tag{B.15}$$

$$\Delta_2':=\frac{\bar{\rho}^2\gamma\lambda^2B^2d}{(\bar{\rho}-\rho-\epsilon)\epsilon}.\tag{B.16}$$

Note that ϵ must be chosen to satisfy $\bar{\rho}-\rho-\epsilon>0$, for example, $\epsilon=\frac{\bar{\rho}-\rho}{2}$. ■

References

[1] J. Konecny, H.B. McMahan, F.X. Yu, P. Richtarik, A.T. Suresh, D. Bacon, Federated learning: Strategies for improving communication efficiency, 2016, arXiv preprint arXiv:1610.05492.

[2] D. Yuan, D.W.C. Ho, S. Xu, Stochastic strongly convex optimization via distributed epoch stochastic gradient algorithm, IEEE Trans. Neural Netw. Learn. Syst. 32 (6) (2021) 2344–2357.

[3] Z. Li, B. Liu, Z. Ding, Consensus-based cooperative algorithms for training over distributed data sets using stochastic gradients, IEEE Trans. Neural Netw. Learn. Syst. 33 (10) (2022) 5579–5589.

[4] X. Wang, J. Yan, B. Jin, W. Li, Distributed and parallel ADMM for structured nonconvex optimization problem, IEEE Trans. Cybern. 51 (9) (2021) 4540–4552.

[5] Q. Yang, Y. Liu, T. Chen, Y. Tong, Federated machine learning: Concept and applications, ACM Trans. Intell. Syst. Technol. 10 (2) (2019) 1–19.

[6] P. Kairouz, H.B. McMahan, Advances and open problems in federated learning, Found. Trends Mach. Learn. 14 (2021) 1–210.

[7] L. Zhou, K.H. Yeh, G. Hancke, Z. Liu, C. Su, Security and privacy for the industrial internet of things: An overview of approaches to safeguarding endpoints, IEEE Signal Process. Mag. 35 (5) (2018) 76–87.

[8] S.U. Stich, Local SGD converges fast and communicates little, in: Proceedings of ICLR, 2019.

[9] X. He, J. Zhang, Q. Ling, Communication-efficient personalized federated learning, in: Proceedings of ICASSP, 2023.

[10] T. Chen, G.B. Giannakis, T. Sun, W. Yin, LAG: Lazily aggregated gradient for communication-efficient distributed learning, in: Proceedings of NeurIPS, 2018.

[11] D. Basu, D. Data, C. Karakus, S.N. Diggavi, Qsparse-local-SGD: Distributed SGD with quantization, sparsification, and local computations, IEEE J. Select. Areas Inf. Theory 1 (1) (2020) 217–226.

[12] S.U. Stich, J.B. Cordonnier, M. Jaggi, Sparsified SGD with memory, in: Proceedings of NeurIPS, 2018.

[13] Q. Li, R. Heusdens, M.G. Christensen, Communication efficient privacy-preserving distributed optimization using adaptive differential quantization, Signal Process. 194 (2022) 108456.

[14] J. Xu, W. Du, Y. Jin, W. He, R. Cheng, Ternary compression for communication-efficient federated learning, IEEE Trans. Neural Netw. Learn. Syst. 33 (3) (2022) 1162–1176.

[15] Y. Liu, G. Wu, Z. Tian, Q. Ling, DQC-ADMM: Decentralized dynamic ADMM with quantized and censored communications, IEEE Trans. Neural Netw. Learn. Syst. 33 (8) (2022) 3290–3304.

[16] L. Lamport, R.E. Shostak, M.C. Pease, The Byzantine generals problem, ACM Trans. Program. Lang. Syst. 4 (3) (1982) 382–401.

[17] Z. Yang, A. Gang, W.U. Bajwa, Adversary-resilient distributed and decentralized statistical inference and machine learning: An overview of recent advances under the Byzantine threat model, IEEE Signal Process. Mag. 37 (3) (2020) 146–159.

[18] Y. Chen, L. Su, J. Xu, Distributed statistical machine learning in adversarial settings: Byzantine gradient descent, in: Proceedings of the ACM on Measurement and Analysis of Computing Systems, ACM SIGMETRICS Perform. Eval. Rev. 46 (1) (2018) 1–25.

[19] D. Yin, Y. Chen, K. Ramchandran, P. Bartlett, Byzantine-robust distributed learning: Towards optimal statistical rates, in: Proceedings of ICML, 2018.

[20] X. Cao, L. Lai, Distributed gradient descent algorithm robust to an arbitrary number of Byzantine attackers, IEEE Trans. Signal Process. 67 (22) (2019) 5850–5864.

[21] P. Blanchard, E.M.E. Mhamdi, R. Guerraoui, J. Stainer, Machine learning with adversaries: Byzantine tolerant gradient descent, in: Proceedings of NeurIPS, 2017.

[22] S. Bulusu, P. Khanduri, P. Sharma, P.K. Varshney, On distributed stochastic gradient descent for nonconvex functions in the presence of Byzantines, in: Proceedings of ICASSP, 2020.

[23] C. Xie, O. Koyejo, I. Gupta, Zeno++: Robust fully asynchronous SGD, in: Proceedings of ICML, 2020.

[24] S.P. Karimireddy, L. He, M. Jaggi, Learning from history for Byzantine robust optimization, in: Proceedings of ICML, 2021.

[25] D. Yin, Y. Chen, K. Ramchandran, P. Bartlett, Defending against saddle point attack in Byzantine-robust distributed learning, in: Proceedings of ICML, 2019.

[26] Z. Allen, F. Ebrahimian, J. Li, D. Alistarh, Byzantine-resilient non-convex stochastic gradient descent, in: Proceedings of ICLR, 2021.

[27] T. Li, A.K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, V. Smith, Federated optimization in heterogeneous networks, in: Proceedings of MLSys, 2020.

[28] L. Li, W. Xu, T. Chen, G.B. Giannakis, Q. Ling, RSA: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets, in: Proceedings of AAAI, 2019.

[29] F. Lin, W. Li, Q. Ling, Stochastic alternating direction method of multipliers for Byzantine-robust distributed learning, Signal Process. 195 (2022) 108501.

[30] J. Peng, Z. Wu, Q. Ling, T. Chen, Byzantine-robust variance-reduced federated learning over distributed non-i.i.d data, Inform. Sci. 616 (2022) 367–391.

[31] J. Peng, W. Li, Q. Ling, Byzantine-robust decentralized stochastic optimization over static and time-varying networks, Signal Process. 183 (2021) 108020.

[32] A. Ghosh, J. Hong, D. Yin, K. Ramchandran, Robust federated learning in a heterogeneous environment, 2019, arXiv preprint arXiv:1906.06629.

[33] S.P. Karimireddy, L. He, M. Jaggi, Byzantine-robust learning on heterogeneous datasets via bucketing, in: Proceedings of ICLR, 2022.

[34] J. Bernstein, J. Zhao, K. Azizzadenesheli, Anandkumar, Signsgd with majority vote is communication efficient and fault tolerant, in: Proceedings of ICLR, 2019.
[35] J. Akoun, S. Meyer, Signsgd: Fault-tolerance to blind and Byzantine adversaries, 2022, arXiv preprint arXiv:2202.02085.
[36] A. Ghosh, R.K. Maity, S. Kadhe, A. Mazumdar, K. Ramachandran, Communication efficient and Byzantine tolerant distributed learning, in: Proceedings of ISIT, 2020.
[37] H. Zhu, Q. Ling, Byzantine-robust distributed learning with compression, IEEE Trans. Signal Inf. Process. Netw. 9 (2023) 280–294.
[38] Y. Dong, G.B. Giannakis, T. Chen, J. Cheng, M. Hossain, V. Leung, Communication-efficient robust federated learning over heterogeneous datasets, 2020, arXiv preprint arXiv:2006.09992.
[39] X. He, H. Zhu, Q. Ling, Byzantine-robust and communication-efficient distributed non-convex learning over non-IID data, in: Proceedings of ICASSP, 2022.
[40] D. Davis, D. Drusvyatskiy, Stochastic model-based minimization of weakly convex functions, SIAM J. Optim. 29 (1) (2019) 207–239.
[41] X. Cao, M. Fang, J. Liu, N.Z. Gong, FLTrust: Byzantine-robust federated learning via trust bootstrapping, in: Proceedings of NDSS, 2021.