AKSEL: Fast Byzantine SGD

Amine Boussetta[1]
Mohammed VI Polytechnic University, Ben Guerir, Morocco
amine.boussetta@um6p.ma

El-Mahdi El-Mhamdi
EPFL, Lausanne, Switzerland
elmahdielmhamdi@gmail.com

Rachid Guerraoui
EPFL, Lausanne, Switzerland
rachid.guerraoui@epfl.ch

Alexandre Maurer
Mohammed VI Polytechnic University, Ben Guerir, Morocco
alexandre.maurer@um6p.ma

Sébastien Rouault
EPFL, Lausanne, Switzerland
sebastien.rouault@epfl.ch

Abstract

Modern machine learning architectures distinguish servers and workers. Typically, a d-dimensional model is hosted by a server and trained by n workers, using a distributed *stochastic gradient descent* (SGD) optimization scheme. At each SGD step, the goal is to estimate the gradient of a cost function. The simplest way to do this is to *average* the gradients estimated by the workers. However, averaging is not resilient to even one single Byzantine failure of a worker. Many alternative *gradient aggregation rules* (GARs) have recently been proposed to tolerate a maximum number f of Byzantine workers. These GARs differ according to (1) the complexity of their computation time, (2) the maximal number of Byzantine workers despite which convergence can still be ensured (breakdown point), and (3) their accuracy, which can be captured by (3.1) their angular error, namely the angle with the true gradient, as well as (3.2) their ability to aggregate full gradients. In particular, many are not *full gradients* for they operate on each dimension separately, which results in a coordinate-wise blended gradient, leading to low accuracy in practical situations where the number (s) of workers that are actually Byzantine in an execution is small ($s << f$).

We propose Aksel, a new scalable median-based GAR with optimal time complexity ($\mathcal{O}(nd)$), optimal breakdown point ($n > 2f$) and the lowest upper bound on the *expected angular error* ($\mathcal{O}(\sqrt{d})$) among *full gradient* approaches. We also study the *actual angular error* of Aksel when the gradient distribution is normal and show that it only grows in $\mathcal{O}(\sqrt{d}\log n)$, which is the first logarithmic upper bound ever proven on the number of workers n assuming an optimal breakdown point. We also report on an empirical evaluation of Aksel on various classification tasks, which we compare to alternative GARs against state-of-the-art attacks. Aksel is the only GAR reaching top accuracy when there is actually none or few Byzantine workers while maintaining a good defense even under the extreme case ($s = f$). For simplicity of presentation, we consider a scheme with a single server. However, as we explain in the paper, Aksel can also easily be adapted to multi-server architectures that tolerate the Byzantine behavior of a fraction of the servers.

2012 ACM Subject Classification Computing methodologies → Batch learning; Security and privacy → Distributed systems security; Theory of computation → Nonconvex optimization

Keywords and phrases Machine learning, Stochastic gradient descent, Byzantine failures

Digital Object Identifier 10.4230/LIPIcs.OPODIS.2020.8

[1] Corresponding author

24th International Conference on Principles of Distributed Systems (OPODIS 2020).
Editors: Quentin Bramas, Rotem Oshman, and Paolo Romano; Article No. 8; pp. 8:1–8:16
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Machine learning (ML) has gained a lot of attention during the last decades, where data collection and processing have reached outstanding levels in terms of volume, variety and velocity. Public awareness of machine learning, especially after the renaissance of neural networks with the backpropagation algorithm [16], increased greatly when companies like IBM and DeepMind created computer programs that beat world class champions in various games. Machine learning started being incorporated within many applications such as transportation, healthcare, finance, agriculture, retail, and customer service.

Essentially, training a supervised ML algorithm consists in determining the set of parameters that minimize the error between the model prediction and the actual output, a scheme formally called *empirical risk minimization* [27]. In a single machine, it is common to use *Gradient Descent* (GD) to minimize the cost function (which depends on the entire dataset) by computing its gradient. For modern applications however, even the best and most expensive hardware would eventually become insufficient.

Almost every industry grade machine learning algorithm is nowadays implemented in a distributed manner. Most rely on *stochastic gradient descent* (SGD) [25], a variant of GD that supports parallelization. However, a distributed architecture induces many challenges, in particular the risk of partial failures. The classical way to model various failures (e.g. software bug, arbitrary behavior of the hardware…) is the *Byzantine* abstraction and the classical way to deal with them is to use a state machine replication protocol [26], but this solution entails heavy communication and computational costs.

More specifically, distributed implementations of SGD typically consist of parameter servers and workers. For simplicity of presentation, we consider the now classical ML scheme with a single parameter server and several workers [1] (but our result can easily be extended to a setting with multiple servers). The dataset is distributed over these workers, each of which computes an estimation of the gradient step based on their share of the data. The parameter server aggregates all the received gradient estimations and updates the parameter vector accordingly. The goal is to come up with an estimate of the (true) gradient that would have been computed on a single machine using GD. The simplest and best way to aggregate the vectors is through averaging [23] which comes very close to the true gradient. However, averaging cannot withstand a single Byzantine failure of a worker [4].

To solve this problem, many *gradient aggregation rules* (GARs) have been proposed to tolerate a (maximum) number f of Byzantine workers (as we discuss later in "Related work"). They can be classified in two main families: *full-GARs*, that select and average gradients of responsive workers keeping the whole information on the descent direction, and *blended-GARs*, that perform coordinate-wise operations on the set of collected gradients, inevitably losing some information (as illustrated by Figure 2 in Section 6). The former are particularly appealing in a practical setting because, even if a GAR is devised to tolerate extreme situations and provide a reasonably good accuracy despite a large number of Byzantine workers, it is important that the GAR provides very good accuracy in most frequent situations where the number (s) of actual Byzantine workers in an execution is small ($s << f$). In this sense, full gradients inherently enable graceful degradation.

The motivation of this work was to ask whether it is possible to derive a full gradient aggregation rule defending against 50% of Byzantine workers ($n > 2f$) with a low time complexity ($\mathcal{O}(nd)$), which are both optimal, but with an angular error close to that of averaging (which is not Byzantine-resilient). We answer positively by presenting AKSEL[2], a

[2] Aksel (known as Kusaila in Arabic and Caecilius in Latin) was an Amazigh leader of the 7th century

new scalable median-based approach to aggregate the gradients. Essentially, Aksel is unique in the sense that it is a full-gradient GAR using indirectly the power of coordinate-wise operations to reduce the angular error.

Looking for optimal breakdown point and time complexity is self justifying. But why seek a low angular error? In fact, this is directly linked to the quality of the solution and the speed of convergence. Intuitively, a large angle makes enough room for Byzantine workers to corrupt the machine learning model. Moreover, two models with different GARs can converge to the same solution, but with different speed. We establish in Corollary 9 the link between the angle value and the convergence slowdown occasioned by the robust GAR compared to averaging.

Related work. Most approaches that have been proposed to improve the Byzantine resilience of gradient descent (and its variants) rely on robust statistics, whilst some use historical information to identify correct workers. KRUM [4] selects the vector with the minimum score defined as the sum of euclidean distances with its neighbors. m-KRUM [9] consists in averaging m KRUM outputs without replacement. BULYAN [12] applies a variant of the trimmed mean on a selection of vectors obtained from m-KRUM.

MEDIAN and b-TRMEAN [31] apply robust statistics on each coordinate of the n gradients. Trimmed mean (b-TRMEAN) removes the smallest and the largest b values and averages the remaining $n - 2b$ values, whereas the median MEDIAN is a special case of Trimmed mean where $b = \lfloor \frac{n}{2} \rfloor$. b-PHOCAS [29] averages the $n - b$ closest values to b-TrMean in each coordinate. MEAMED [28] is a special case of Phocas where the trimmed mean is replaced with the median. GEOMETRIC MEDIAN OF MEANS [8] computes the average of m batches of gradients, then computes the geometric median of those averages. Since no exact algorithm is available for GEOMED, the $(1 + \epsilon)$-approximation is used instead. DRACO [7] uses coding theory and a redundancy scheme to aggregate the gradients. BYZANTINESGD [2] and KARDAM [10] both use historical information on the gradients and construct filters that allow to distinguish bad workers from honest ones. Recent techniques from Multidimensional approximate agreement [14, 20] are also good candidates because the output of the correct workers remains inside the convex hull of the correct workers input, which is a desirable property for the problem at hand.

The median is particularly interesting for it constitutes a straightforward mechanism to deal with outliers. Yet, although the median is guaranteed to be inside the set of correct scalar values, its multidimensional variant (Coordinate-wise Median) may not lie within the convex hull of correct vectors. Second, the median heavily protects against outliers at the expense of statistical meaning. As a matter of fact, the median throws away many interesting values which makes it less efficient, as we explain later. b-TRMEAN and b-PHOCAS are very efficient when the truncation parameter b is greater than the number of Byzantine workers. However, to defend against $s = \lceil \frac{n}{2} \rceil - 1$, the value of b must be equal to its upper bound and the two GARs are reduced to their special cases, namely, MEDIAN and MEAMED. Otherwise, they become as vulnerable as averaging, whose deviation under attack is unbounded. One common aspect about these blended-GARs is the fact that they defend against dimensional attacks [28] but cannot reach top accuracy in honest settings with none or few Byzantine workers. The only full-GARs proposed to this day are KRUM, m-KRUM and BULYAN. These are all powerful, but they have a high time complexity (at least $\mathcal{O}(n^2 d)$) and their breakdown point is far from optimal. DRACO is the only aggregation rule not suffering from

who resisted the conquest of North Africa while making pragmatic alliances with the Byzantines.

vulnerabilities of common statistics. However, it only defends against a very limited number of Byzantine workers because of the redundancy scheme. Also, Draco cannot be used in settings where privacy matters, because of the matrix allocation mechanism needed before the encoding phase. ByzantineSGD and Kardam are different from the first category of GARs because they use information on past gradients to filter the Byzantine estimates. Although theoretical guarantees have been provided for convergence, ByzantineSGD requires too many parameters to be tuned, which make it less practical. Kardam is the only GAR tolerating asynchrony, but it only works for Lipschitz loss functions, and defends only against $n > 3f$. Finally, multidimensional approximate agreement algorithms are round based, which means that, at each SGD iteration, many rounds ($\mathcal{O}(\log \frac{\Delta}{\epsilon})$, Δ being the initial diameter of the correct set of workers) need to be executed in order to agree on a gradient with an error rate ϵ. These techniques may be advantageous in coordinator-free settings (fully decentralized learning). Table 1 compares various GARs to Aksel according to several properties.

Basically, the full-GARs achieve top accuracy when $s << f$ but are not optimal in terms of complexity and break down point. They also have a big angular error. In contrast, blended-GARs have optimal complexity and break down point with a small angular error, but do not achieve top accuracy when $s << f$. Aksel achieves the best of both worlds.

Table 1 Comparing the time complexity (TC), the breakdown point (BDP) and the expected angular error of gradient aggregation rules (GARs). Parameter f denotes the maximal number of Byzantine workers. Parameter m is specific to m-Krum (which consists in averaging m Krum outputs without replacement). Parameter b is specific to Phocas and TrMean and sets the level of truncation. Aksel is the best full-GAR for all three properties.

GARs	TC	BDP	Angular error $f = \mathcal{O}(1)$	Angular error $f = \mathcal{O}(n)$
Averaging	$\mathcal{O}(nd)$	$f = 0$	$\mathcal{O}(\sqrt{\frac{d}{n}})$	$\mathcal{O}(\sqrt{\frac{d}{n}})$
Full-aggregathors				
Krum	$\mathcal{O}(n^2d)$	$n > 2f + 1$	$\mathcal{O}(\sqrt{nd})$	$\mathcal{O}(n\sqrt{d})$
m-Krum	$\mathcal{O}(n^2d)$	$n > 2f + 2$ $m < n - f - 2$	$\mathcal{O}(\sqrt{nd})$	$\mathcal{O}(n\sqrt{d})$
Bulyan	$\mathcal{O}(n^2d)$	$n > 4f + 2$	$\mathcal{O}(\sqrt{nd})$	$\mathcal{O}(n\sqrt{d})$
Aksel	$\mathcal{O}(nd)$	$n > 2f$	$\mathcal{O}(\sqrt{d})$	$\mathcal{O}(\sqrt{d})$
Blended-aggregathors				
Median	$\mathcal{O}(nd)$	$n > 2f$	$\mathcal{O}(\sqrt{d})$	$\mathcal{O}(\sqrt{d})$
$(1+\epsilon)$-GeoMed	$\mathcal{O}(nd)$	$n > 2f$	$\mathcal{O}(\sqrt{nd})$	$\mathcal{O}(\sqrt{nd})$
b-Phocas	$\mathcal{O}(nd)$	$n > 2f$ $b > f$	$\mathcal{O}(\sqrt{\frac{d}{n}})$	$\mathcal{O}(\sqrt{d})$
b-TrMean	$\mathcal{O}(nd)$	$n > 2f$ $b > f$	$\mathcal{O}(\sqrt{\frac{d}{n}})$	$\mathcal{O}(\sqrt{d})$
MeaMed	$\mathcal{O}(nd)$	$n > 2f$	$\mathcal{O}(\sqrt{d})$	$\mathcal{O}(\sqrt{d})$

Contributions. We present in this paper Aksel, a new median based algorithm which is the first to have the 4 following properties simultaneously:

- Optimal time complexity $\mathcal{O}(nd)$
- Optimal breakdown point $n > 2f$
- Full gradient aggregation (high accuracy reachable for $s << f$)
- Constant upper bound ($\mathcal{O}(d)$ in the number of workers n, see Lemma 10) on the expected angular error (scalability)

On the theoretical side, we prove (1) the (α, f)-Byzantine resilience of Aksel; (2) its convergence for non convex and strongly convex losses; and (3) a logarithmic upper bound of the real angular error of Aksel.

On the practical side, we report on an empirical evaluation of our distributed implementation of Aksel. In particular, we consider two state-of-the-art attacks [3, 30] on academic classification tasks (MNIST, Fashion-MNIST and CIFAR-10). Aksel reaches the top accuracy when $s << f$, and maintains a good accuracy in the extreme case $s = f$. Aksel does also have some advantages that may appeal to practitioners: it requires no parameter tuning for the aggregation (a time consuming task in general) and no knowledge of the number of Byzantine workers (which can be fatal if underestimated, e.g. b-Phocas and b-TrMean). Aksel is also based on simple mathematical functions (i.e. median, subtraction, sum-of-squares, averaging) which makes it simple to analyze.

A recent paper [11] proposed a genuinely distributed scheme with multiple servers, tolerating the Byzantine failures of a fraction of them by composing established GARs such as Krum, m-Krum and Bulyan. For pedagogical reasons, we present here Aksel in a single-server setting, focusing on improving resilience to failures of workers. However, Aksel satisfies the properties required by [11] from a GAR, and could therefore be used also in a multi-server setting instead of Krum, m-Krum and Bulyan in [11].

Outline. The paper is organized as follow. We first present our model in Section 2. After some preliminaries in Section 3, we motivate the design of our algorithm and present it in Section 4. Theoretical guarantees on its Byzantine resilience and convergence are presented in Section 5. Section 6 reports on a selection of empirical results. We conclude the paper by discussing some open issues in Section 7. For space limitations, we defer all the proofs and the full empirical evaluation to the appendix.

2 Model

As discussed previously, most machine learning algorithms use gradient descent (GD) to minimize a cost function $F(\mathbf{w}_t)$ where $\mathbf{w}_t$ is a vector of parameters[3] at time t. Typically, the cost function is a sum of individual errors run through many examples of the data set. Vanilla GD runs the sum through the entire dataset. However, this takes a lot of time to compute, and this is not realistic with huge datasets involving hundreds of billions of examples. Another variant is the stochastic gradient descent algorithm (SGD) which only uses a single example in each iteration. This method is very fast but noisy. A compromise is to construct a mini batch, namely a small subset of the dataset, run the sum of individual errors over this mini batch, and compute the gradient. A randomly sampled mini batch typically contains redundant examples, which can be useful to smooth out noisy gradients. Mini batch SGD is a good choice to compute quality gradients in a reasonable time. Besides, mini batch SGD is highly parallelizable. One can generate n mini batches and compute n gradients, then average them to get a very good estimate of the true gradient. In a distributed setting, randomly sampled mini batches are allocated to n workers (compute nodes), and a server aggregates those gradients then updates the parameter vector $\mathbf{w}_t$.

[3] For instance, the weights and biases of a neural network.

2.1 Distributed SGD

We follow the classical distributed SGD model [1] where a parameter server (PS) broadcasts, in each *synchronous* round t, the parameter vector $\mathbf{w}_t \in \mathbb{R}^d$ to n workers. We consider that f among these n workers can be Byzantine. Each correct worker i computes an estimate $\mathbf{V_i}^t = \mathbf{G}(\mathbf{w}_t, \xi_i^t)$ of the gradient $\nabla F(\mathbf{w}_t)$ of the cost function F, where ξ_i^t is an independent and identically distributed (i.i.d.) random variable representing the subset of the dataset, drawn randomly for worker i. The PS aggregates the n received gradients $(\mathbf{V_1^t}, \mathbf{V_2^t}, ..., \mathbf{V_n^t})$ using its choice function $\mathcal{A}$ called *aggregation rule*, then updates the parameter vector using the following SGD equation:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \gamma \mathcal{A}(\mathbf{V_1^t}, \mathbf{V_2^t}, ..., \mathbf{V_n^t})$$

where γ is an arbitrary constant called *learning rate*.

2.2 Adversary

A Byzantine worker has the full knowledge of the system, including the aggregation rule and the vectors proposed by other workers. It can collude with other Byzantine workers to perform attacks against the aggregation rule and prevent convergence, or make the model converge to ineffective solutions. The Byzantine workers can for instance send arbitrary values, strategically-chosen values that exploit the environment, or null values corresponding to a classical crash failure. Since we are working in a synchronous system, when a vector is not received, the PS assumes that it is a null vector. Dimensional attacks were presented in [28], meaning that corruption can happen anywhere in the gradient matrix as long as each dimension contains a majority of correct values. However, we believe that such scenario could be avoided by introducing cryptography schemes (e.g. RSA signatures / AES encryption and decryption / Diffie-Hellman secure exchange of keys…) to make sure that impersonation is not possible, and keep the same threat model as in [4].

2.3 Assumptions

We now state the (rather standard) assumptions made in this paper by default: in the rest of the paper, all assumptions, except Assumption 5, are always assumed to be true, unless specified otherwise.

▶ **Assumption 1.** *(Breakdown point) The number of Byzantine workers is strictly less than the number of correct ones: $n > 2f$*

▶ **Assumption 2.** *(Smoothness) F is $L-$smooth:*
$\forall \mathbf{w}', \mathbf{w}, \|\nabla F(\mathbf{w}') - \nabla F(\mathbf{w})\| \leq L \|\mathbf{w}' - \mathbf{w}\|$

▶ **Assumption 3.** *(Strong convexity) F is $K-$strongly convex:*
$\forall \mathbf{w}', \mathbf{w}, F(\mathbf{w}') \geq F(\mathbf{w}) + \langle \nabla F(\mathbf{w}), \mathbf{w}' - \mathbf{w} \rangle + \frac{K}{2} \|\mathbf{w}' - \mathbf{w}\|^2$

▶ **Assumption 4.** *(Bounded variance and unbiased estimators) The proposed vectors are unbiased estimates of the true gradient and their variance is bounded:*
$\forall i \in \{1, \cdots, n\}, \mathbb{E}\, \mathbf{V_i} = \nabla F$ *and* $\mathbb{E} \|\mathbf{V_i} - \nabla F\|^2 < d\sigma^2$

▶ **Assumption 5.** *(Normal distribution;* not a default assumption*) The proposed vectors are normally distributed around the true gradient ∇F: $\forall i \in \{1, \cdots, n\}, \mathbf{V_i} \sim \mathcal{N}(\nabla F, \boldsymbol{\sigma}^2)$ where $\boldsymbol{\sigma}^2 = diag(\sigma^2)$ is a $d \times d$ diagonal covariance matrix.*

Assumption 1 is very common in synchronous distributed systems. It is however worth noting that beyond the classical impossibility results in distributed computing, this assumption is a direct consequence of another impossibility result in robust statistics [24], even when all the operations are done in a *single machine*. Assumptions 2 and 4 are common in the SGD literature [5] and Assumption 3 is typically needed to prove convergence rates [6]. We also analyze Aksel and median based GARs in general under Assumption 5. This assumption is substantiated by recent empirical findings in machine learning, where many normally distributed datasets naturally yield normally distributed gradients [18]. As we detail later, our experimental findings illustrate that Aksel performs well in commonly used datasets.

3 Preliminaries

We recall in this section background results on the robustness of the median and the probabilistic absolute error between the extreme value and the mean of normal samples. These will also be useful when describing the properties of our algorithm. We also recall the measure of Byzantine resilience in the context of distributed SGD .

3.1 Robustness of the median

Mosteller and Tukey [21] defined two types of robustness: resistance and efficiency. The first notion conveys the fact that an infinite change caused by a small part of a group has a bounded impact on the value of the estimate. The second means that the estimate is close to the optimal estimate in a variety of situations and not only in a particular one. Many robust estimators have been proposed for scale and location. In this paper, we focus on the median, a robust estimator of the location which is the value that separates a sorted set into two equal parts. Formally: Let $\mathbf{X} = (x_1, x_2, \cdots, x_n)$ be a set of n values, then:

$$med(\mathbf{X}) = \arg\min_{y} \sum_{i=1}^{n} |x_i - y|$$

In high dimensions, we work with the coordinate-wise median, defined as follow: Let $\mathbf{M} = (\mathbf{V_1}, \mathbf{V_2}, \cdots, \mathbf{V_n})$ be a matrix with n column vectors $\mathbf{V_i} = (v_{1i}, v_{2i}, \cdots, v_{di})^T$ in $\mathbb{R}^d$, then: $\textsc{Median}(\mathbf{M}) = (m_1, m_2, \cdots, m_d)^T$, where $m_j = med(v_{j1}, v_{j2}, \cdots, v_{jn}), \forall j \in [1, \cdots, d]$.

The median has high efficiency for normal data (64%) [21], and most importantly, an optimal breakdown point (50%). The last point implies that corrupting 50% of the data will have only limited impact on the location parameter. Moreover, as known from the works on Byzantine tolerant approximate agreement and clock synchronization, the median always lies inside the subset of correct values when more than 50% of the data is correct. To formalize this, we restate Lemma 4 from [28] without proof.

▶ **Lemma 6.** *For a sequence composed of f Byzantine values and $n - f$ correct values $x_1, x_2, \cdots, x_{n-f}$, if $f \leq \lceil \frac{n}{2} \rceil - 1$ (the correct values dominates the sequence), then the median value m of this sequence satisfies $m \in [x_{min}, x_{max}]$.*

3.2 Distribution of extreme normal values

The maximum or the minimum values observed when drawing normal samples changes when n takes different values. The extreme value theory [15] shows that the extreme values of a normal distribution follows a Gumbel distribution, depending on the number n of samples drawn. Thanks to the symmetry of our problem, we only discuss the maximum value. Formulas for the minimum are derived in a similar way. Kotz and Nadarajah [19] show that the distribution of the maximum of n samples drawn from a standard normal random variable $\mathcal{N}(0,1)$ with a standard normal quantile function $\Phi^{-1}(x)$ has the following statistics. Let $\mu_m(n), \sigma_m(n), q_m^p(n)$ be the mean, the standard deviation and the p^{th} quantile of the maximum distribution when n samples are drawn from a standard normal distribution. Then:

$$\begin{aligned} \mu_m(n) = & \quad \Phi^{-1}(1 - \frac{1}{n}) \\ \sigma_m(n) = & \quad \Phi^{-1}(1 - \frac{1}{n \quad e}) - \mu_m(n) \\ q_m^p(n) = & \quad \mu_m(n) - \sigma_m(n) \log(-\log(p)) \end{aligned} \tag{1}$$

We use these results to compute a probabilistic bound of the gap between the mean and the maximum value of n normal samples.

▶ **Lemma 7.** *Let $\mathbf{X} = \{\mathbf{x_1}, \cdots, \mathbf{x}_{n-f}\}$ be a set of column vectors drawn from a multivariate normal random variable $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$ with $\boldsymbol{\mu} = (\mu_1, \cdots, \mu_d)^T$ and the covariance matrix $\boldsymbol{\sigma} = diag_{n\times d}(\sigma)$. Let $\mathbf{B} = \{\mathbf{b_1}, \cdots, \mathbf{b_f}\}$ be a set of arbitrary column vectors and $(n, f) \in \mathbb{N}^2$. Let $\mathbf{S} = \mathbf{X} \cup \mathbf{B}$ and $\mathbf{M} = Median(\mathbf{S})$. Let $\mathcal{E}_k$ be the following event for the k^{th} coordinate: $|\mathbf{M}[k] - \boldsymbol{\mu}[k]| \leq \lambda(n,p)$. We then have, $\forall p \in [0,1)$ and $\forall n \in \mathbb{N}$: $P\left[\bigwedge_{k=1}^{d} \mathcal{E}_k\right] = p$, where*

$$\lambda(n,p) = \Phi^{-1}(1 - \frac{1}{n}) \left(1 + \log\left[-\log(p^{\frac{1}{d}})\right]\right) - \Phi^{-1}(1 - \frac{1}{ne}) \left(\log\left[-\log(p^{\frac{1}{d}})\right]\right)$$

3.3 Measuring the Byzantine resilience of GARs

We make use of the now classical metric to evaluate the Byzantine resilience of gradient aggregation rules [4, 12, 9, 28]. This metric encompasses two conditions. First, as long as a proposed vector lies inside a cone around the true gradient, with an angle less than $\frac{\pi}{2}$ (first condition), and as long as its statistical moments are controlled by the moments of the (correct) gradient estimator $\mathbf{G}$ (second condition), this vector can be considered correct and will make a step toward the minimum of the function being optimized using SGD. The second condition allows to transfer the control (classically expressed as bounds on the moments of the gradient estimator $\mathbf{G}$ [5]) of the discrete nature of the SGD dynamics to the choice function $\mathcal{X}$. Below, we recall the definition of (α, f)-Byzantine resilience (introduced in [4]):

▶ **Definition 8.** *Let $0 < \alpha < \frac{\pi}{2}$ be any angular value and $f \in \{0, \cdots, n\}$. Let $\mathbf{V_1}, \cdots, \mathbf{V_n}$ be any independent identically distributed random vectors in $\mathbb{R}^d$ with $\mathbb{E}\,\mathbf{V_i} = \mathbf{G}, \forall i \in \{1, \cdots, n\}$. Let $\mathbf{B_1}, \cdots, \mathbf{B_f}$ be any random vectors in $\mathbb{R}^d$, possibly dependent on the $\mathbf{V_i}$'s. A choice function $\mathcal{X}$ is said to be (α, f)-Byzantine resilient if, for any $1 \leq j_1 < \cdots < j_f \leq n$, the vector $\mathcal{X} = \mathcal{X}(\mathbf{V_1}, \cdots, \underbrace{\mathbf{B_1}}_{j_1}, \cdots, \underbrace{\mathbf{B_f}}_{j_f}, \cdots, \mathbf{V_n})$ satisfies the following two conditions:*

- ***Condition (i):*** $\langle \mathbb{E}\,\mathcal{X}, \mathbf{G}\rangle \geq (1 - sin\alpha)\,\|\mathbf{G}\|^2$
- ***Condition (ii):*** *for $r = 2, 3, 4$, $\mathbb{E}\,\|\mathcal{X}\|^r$ is bounded above by a linear combination of terms of the form $\mathbb{E}\,\|\mathbf{G}\|^{r_1} \cdots \mathbb{E}\,\|\mathbf{G}\|^{r_{n-1}}$ with $r_1 + \cdots + r_{n-1} = r$*

Generally, *condition (i)* can be proved by showing that $\mathbb{E}\,\mathcal{X}$ belongs to the ball centered at $\mathbf{G}$ with radius $r = \eta(.)\sqrt{d}\sigma$ (formally: $\|\mathbb{E}\,\mathcal{X} - \mathbf{G}\| < \eta(.)\sqrt{d}\sigma$), where $\eta(.)$ is a positive function, d is the dimension of the model and σ is the standard deviation of the gradient estimator.

▶ **Corollary 9.** *The function $\eta(.)$ is positively correlated to the slowdown of convergence speed occasioned by the aggregation rule $\mathcal{X}$ compared to averaging.*

4 The AKSEL Algorithm

We present our aggregation protocol AKSEL in 4.1, discuss the rationale behind its design in 4.2 and give its time complexity in 4.3.

4.1 Aksel

Algorithm 1 AKSEL: Scalable gradient aggregation rule.

Input: $\mathbf{V} = (\mathbf{V_1}, \mathbf{V_2}, \cdots, \mathbf{V_n})$: $d \times n$ matrix (received gradients)
Output: Y: $d \times 1$ vector

```
/* Computing the sum of squares of each column vector V_i centered around the
   coordinate-wise median                                                    */
```

Let $\mathbf{S}$ be a row vector $(1 \times n)$ and $\mathbf{M}$ a column vector $(d \times 1)$

$\mathbf{M} = (\mathbf{M}[1], \mathbf{M}[2], \cdots, \mathbf{M}[d])^T$ = coordinate-wise median vector constructed from $\mathbf{V}$

$\mathbf{S} = (\sum_{j=1}^{d}(\mathbf{V_1}[j] - \mathbf{M}[j])^2, \sum_{j=1}^{d}(\mathbf{V_2}[j] - \mathbf{M}[j])^2, \cdots, \sum_{j=1}^{d}(\mathbf{V_n}[j] - \mathbf{M}[j])^2)$

```
/* Constructing a robust interval                                            */
```

Let r be the median of the set $\mathbf{S}$

Let $\mathbf{I} = [0, r]$

```
/* Averaging the new subset of column vectors from V                         */
```

Let $\mathbf{N}$ the subset of vectors $\mathbf{V_i}$'s such that $\|\mathbf{V_i} - \mathbf{M}\|^2 \in \mathbf{I}$ and $|\mathbf{N}| = p$

$\mathbf{Y}[j] = \frac{1}{p} \sum_{\mathbf{V_i} \in \mathbf{N}} \mathbf{V_i}[j], \quad \forall j \in \{1, \cdots, d\}$

4.2 Rationale

The goal of any aggregation rule is to produce a vector as close as possible from the true gradient of the cost function. This puts conditions on the norm as well as on the direction of the aggregated vector. Clearly, any rule that focuses only on the vectors norms comparison will not succeed because of the vulnerabilities of l_p-norms, as pointed in [12]. For example, $V_{Correct} = (2, 2, 2, 2, 5, 5, 5, 3)^T$ and $V_{Byzantine} = (10, 0, 0, 0, 0, 0, 0, 0)^T$ are two vectors with the same norm and very different coordinates. One way to address this issue is to add a constraint on the coordinates of all vectors by centering them around a robust location estimator. We choose the coordinate-wise median in this work.

Since MEDIAN is a *blended-GAR* and provides only one aggregate which is very far from the correct mean, we choose to incorporate more vectors in the aggregation process. Therefore, a better alternative is to choose an interval around the median, and to average the values within this interval. Using this alternative, we are guaranteed to produce, most of the times, an aggregated value that lies between the real mean and the deviated median. Figure 1 illustrates the idea that an interval is better than a single value. Many GARs used this concept on each coordinate to improve the defense mechanism [31, 28, 29]. However, operating on each coordinate has consequences on the overhead cost of the Byzantine resilience. As a matter of fact, coordinate-wise operations lead to a blended vector which is different (in structure) from the full gradients. As a consequence, the top accuracy is never reached even in honest environments ($s = 0$).

AKSEL is unique in the sense that it is a full-gradient GAR using indirectly the power of coordinate-wise operations. It performs the filtering method on the squared norms of the centered vectors, rather than selecting the mean around the median in each coordinate, in order to aggregate full gradients. Since norms are positive, the filter interval will be $[0, r]$, where r is the median of norms in our work.

The idea of centering the vectors around their coordinate-wise median is a very powerful guardrail against the vulnerability of norms, and also a very handy tool for the proof development. In fact, it is hard to come up with a probability density function for the sum of squares of normal variables with nonzero expectation, although a recent work [13] has shown that it is possible to derive a complex cumulative distribution function (but no elementary expression for the density function). Subtracting a scalar from each coordinate makes us close enough to normal variables with zero expectation, whose sum of squares density function is known and expressed through elementary expressions. We derive in Lemma 16 the expectation and the variance of the sum of squares of normal samples centered around a scalar (which, in our case, is equal to the median of the n values) for each coordinate.

4.3 Complexity

Our AKSEL aggregation rule has an optimal time complexity $\mathcal{O}(nd)$. First, AKSEL computes the coordinate-wise median (M) in $\mathcal{O}(nd)$ steps. Next, it subtracts M from all n gradients and computes their euclidean norms, also in $\mathcal{O}(nd)$ steps. Then, AKSEL computes the median (m) of the n norms using a *Quickselect* [17] in $\mathcal{O}(n)$ steps. Finally, it averages the vectors whose norm is less than (m) in $\mathcal{O}(nd)$ steps. The global time complexity is therefore $\mathcal{O}(nd)$.

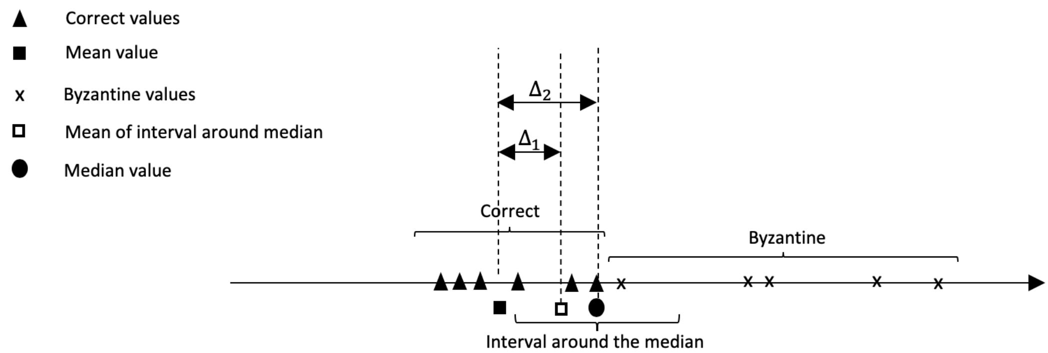

Figure 1 Comparison of (1) the median and (2) the mean of an interval around the median, in terms of distance to the mean. In a setting where the number of Byzantine workers is exactly the number of correct workers minus one, and their values are all positioned in an extremum side, the median is always the farthest correct value from the mean among correct values. However, taking the average of values inside an interval around the median can reduce the distance to the mean value in many situations.

5 Theoretical Guarantees of Aksel

We give an upper bound on the variance of AKSEL and prove its (α, f)-Byzantine resilience as well as its convergence properties for non convex as well as strongly convex losses.

5.1 Bounded variance

The following lemma states an upper bound of the variance of AKSEL.

▶ **Lemma 10.** *Let $\mathbf{V_1}, \cdots, \mathbf{V_n}$ be any random d-dimensional vectors in $\mathbb{R}^d$, f among them being possibly Byzantine. Under Assumptions 1 and 4, the variance of AKSEL is upper bounded, and we have:*

$$\mathbb{E}\left\|\mathbf{A} - \nabla F\right\|^2 \leq \left(4 + \frac{12\lceil \frac{n}{2} \rceil (n-f)}{(n - \lceil \frac{n}{2} \rceil - f + 1)^2}\right) d\sigma^2 \quad \sim \mathcal{O}(d)$$

5.2 Byzantine resilience

Following Definition 8, the (α, f)-Byzantine resilience of AKSEL can be proved by showing first that the aggregated vector $\mathbb{E}\,\mathbf{A}$ is pointing in the same direction and has a close norm to the true gradient ∇F (*condition i*) and its statistical moments are controlled by a linear combination of the statistical moments of the correct gradient estimator (*condition ii*). We prove the two conditions through the following lemmas.

▶ **Lemma 11** (Expected angular error)**.** *If Assumptions 1 and 4 hold, the angular error of* AKSEL *is upper bounded as follow:* $\|\mathbb{E}\,\mathbf{A} - \nabla F\|^2 \leq \eta^2(n, f) d\sigma^2$ *where:*

$$\eta^2(n, f) = 4 + \frac{12\lceil\frac{n}{2}\rceil(n - f)}{(n - \lceil\frac{n}{2}\rceil - f + 1)^2}$$

▶ **Lemma 12** (Controlled statistical moments)**.** *If Assumptions 1 and 4 hold, the statistical moments of* AKSEL *are upper bounded by a linear combination of the statistical moments of the correct gradient estimator:*

$$\mathbb{E}\,\|\mathbf{A}\|^r \leq C \sum_{r_1+\cdots+r_{n-f}=r} \|G\|^{r_1} \cdots \|G\|^{r_{n-f}}$$

We now present the (α, f)-Byzantine resilience result in the following theorem:

▶ **Theorem 13.** *Let* $\mathbf{V_1}, \cdots, \mathbf{V_n}$ *be a set of gradient estimates in* $\mathbb{R}^d$*. Under Assumptions 1 and 4, if* $\eta(n, f)\sqrt{d}\sigma < \|\nabla F\|$*, then* AKSEL *is* (α, f)*-Byzantine resilient where* $\alpha \in [0, \frac{\pi}{2}]$ *is defined by:* $sin\alpha = \frac{\eta(n,f)\sqrt{d}\sigma}{\|\nabla F\|}$

5.3 Convergence for non convex losses

When analyzing optimization algorithms under the non convexity assumption, the objective function can have several local minima instead of one global minimum. A simple solution would be to partition the parameter space into many convex pools and proceed as in the convex case. Bottou [5] proposes however to study the convergence of the objective function and its gradient instead of the parameter vector itself. When some conditions are met regarding the cost function being minimized and the learning rate, SGD converges almost surely to a flat region, where the gradient is very small. Blanchard et al. [4] combine this result with the (α, f)-Byzantine resilience framework to derive a second result on the almost sure convergence of SGD using an (α, f)-Byzantine resilient aggregation rule. Since AKSEL is Byzantine resilient, as proven in Theorem 13, we only restate the convergence result without proof in Theorem 14. The reader is kindly referred to [4] and [5] for more details on the convergence analysis.

▶ **Theorem 14.** *Let* $\mathbf{A_t}$ *be the output of the* AKSEL *aggregation rule over the* n *received gradients* $\mathbf{V_i} \sim \mathbf{G}$*. We assume that (i) the cost function* F *is three times differentiable with continuous derivatives and is non negative (*$F(\mathbf{w}) \geq 0$*); (ii) the learning rate satisfies* $\sum_t \gamma_t = \infty$ *and* $\sum_t \gamma_t^2 < \infty$*; (iii) the gradient estimator satisfies* $\mathbb{E}\,\mathbf{G}(\mathbf{w}) = \nabla F(\mathbf{w})$ *and* $\forall r \in \{2, 3, 4\}, \mathbb{E}\,\|\mathbf{G}(\mathbf{w})\|^r \leq A_r + B_r \|\mathbf{w}\|^r$*; (iv) there exists a constant* $0 \leq \alpha \leq \frac{\pi}{2}$ *such that* $\forall \mathbf{w}$*,* $\eta(n, f)\sqrt{d}\sigma \leq \|\nabla F(\mathbf{w})\|\, sin\alpha$*; (v) finally, beyond a certain horizon* $\|\mathbf{w}\|^2 \geq D$*, there exist* $\epsilon > 0$ *and* $0 \leq \beta \leq \frac{\pi}{2} - \alpha$ *such that:*

$$\|\nabla F(\mathbf{w})\| \geq \epsilon$$

$$\frac{\langle \mathbf{w}, \nabla F(\mathbf{w})\rangle}{\|\mathbf{w}\|\,\|\nabla F(\mathbf{w})\|} \geq \cos\beta$$

Then, the sequence of gradients $\nabla F(\mathbf{w}_t)$ *converges almost surely to zero.*

5.4 Convergence for strongly convex losses

Finally, we derive the statistical error rate of SGD using AKSEL as an aggregation rule.

▶ **Theorem 15.** *Let $F(\mathbf{w})$ be the cost function being optimized, $\nabla F(\mathbf{w})$ its actual gradient and $\mathbf{A}$ the output of the AKSEL aggregation rule over the n received gradients. When Assumptions 1, 2, 3 and 4 hold, then after T iterations of SGD updates using the AKSEL GAR with a step size $\alpha_t = \frac{1}{L}$, we have:*

$$\mathbb{E}\,\|\mathbf{w}_T - \mathbf{w}_*\| \leq \left(1 - \frac{K}{L+K}\right)^T \|\mathbf{w}_0 - \mathbf{w}_*\| + \frac{2\sqrt{\mathbf{\Delta}}}{K}$$

$$\mathbb{E}[F(\mathbf{w}_T) - F(\mathbf{w}_*)] \leq \frac{\mathbf{\Delta}}{2L} + \left(1 - \frac{K}{L}\right)^T \left\| F(\mathbf{w}_0) - F(\mathbf{w}_*) - \frac{\mathbf{\Delta}}{2L} \right\|$$

with: $\mathbf{\Delta} = \left(4 + \frac{12\lceil\frac{n}{2}\rceil(n-f)}{(n-\lceil\frac{n}{2}\rceil-f+1)^2}\right) d\sigma^2$

5.5 Probabilistic upper bound on the real angular error of Aksel

In the previous section and in all the related work, results are derived in *expectation*. In fact, recent works only study the expected angular error, the variance (the expected squared absolute error) and the expected statistical error in convergence. Up to our knowledge, [31] is the only work addressing these quantities without expectation. More specifically, they study the two well known GARs MEDIAN and TRMEAN when applied with the gradient descent algorithm, assuming unbiased gradient estimates with bounded variance and skewness. They achieve an upper bound on the variance decreasing like $\mathcal{O}(\frac{1}{\sqrt{n}})$ using normal approximations and Berry-Essen inequalities, but their breakdown point is very far from optimal:

$$\alpha + \sqrt{\frac{d\log(1+nmLD)}{n(1-\alpha)}} + 0.4748\frac{S}{\sqrt{m}} \leq \frac{1}{2} - \epsilon$$

where α is the ratio of Byzantine workers, n is the number of workers, m is the number of data points each worker has, D is the diameter of the parameter space, L is the Lipschitz constant, d is the dimension of the model and S is the skewness upper bound.

We study the optimal robustness ($\alpha < \frac{1}{2}$) of AKSEL applied with stochastic gradient descent when gradients are normally distributed, and we show that the real angular error only has a logarithmic growth ($\mathcal{O}(\sqrt{d}\log n)$) in the number of workers n under this assumption[4].

Expectation and variance of the squared norm of a centered vector

An important step in our algorithm is to sum the squares of all the coordinates centered around their median value. When Assumption 5 holds, it is possible to derive the expectation and the variance of this quantity using the asymptotic approximation of the Gamma distribution and simple bounding properties. We formalize this in the following lemma:

▶ **Lemma 16.** *Let X_i be a normal random variable where μ_i is the mean, σ^2 is the variance and m_i is a value such that $|m_i - \mu_i| \leq \lambda\sigma$. If $Z_i = X_i - m_i$ is the new random variable X_i centered around m_i and $S = \sum_{i=1}^{d} Z_i^2$, then we have:*

$$\mathbb{E}[S] = (1+\lambda^2)d\sigma^2$$

$$var[S] = 2d\sigma^4\left(1 + 2\lambda^2\sigma^2\right)$$

[4] This result is interesting in its own right. Many median based GARs can benefit from this new analysis. In particular, MEDIAN which has been studied under non optimal robustness [31] and MEAMED whose expected angular error was shown to be growing as $\mathcal{O}(\sqrt{nd})$ [28]

Upper bound on the absolute error of Aksel

Note that the (α, f)-Byzantine resilience and convergence theorems will be exactly the same in our new analysis. It suffices to derive the upper bound on the absolute error $\|\mathbf{A} - \nabla F\|^2$ and use it in every appearance of $\mathbb{E}\,\|\mathbf{A} - \nabla F\|^2$ in the previous results while dropping the expectation sign and introducing the probabilistic statement (with probability p) before each result.

In the following lemma, we upper bound the absolute error between Aksel's output and the true gradient in the squared norm sense.

▶ **Lemma 17.** *Let $\mathbf{V_1}, \cdots, \mathbf{V_n}$ be any random d-dimensional vectors, f among them being possibly Byzantine. Let $\lambda = \Phi^{-1}(1 - \frac{1}{n})\left(1 + \log\left[-\log(p^{\frac{1}{d}})\right]\right) - \Phi^{-1}(1 - \frac{1}{ne})\left(\log\left[-\log(p^{\frac{1}{d}})\right]\right)$, where $p \in [0, 1)$ is an arbitrary probability. When Assumptions 1 and 5 hold, the gap $\|\mathbf{A} - \nabla F\|$ is upper bounded, and we have, with probability p:*

$$\|\mathbf{A} - \nabla F\|^2 \leq 2\left[1 + 2\lambda^2 + \lambda\frac{\sqrt{2(1+2\lambda^2)}}{\sqrt{d}}\right] d\sigma^2 \qquad \sim \mathcal{O}(d\log^2 n)$$

6 Empirical Evaluation

We fully implemented and evaluated Aksel in a distributed setting. Due to space limitations, we only present here a selection of empirical results. A detailed version of the setup, as well as an extensive set of experiments, can be found in the appendix.

We tested Aksel (and its competitors) both in settings with no Byzantine players as well as against two state-of-the-art attacks, namely "A little is enough" [3] and "Fall of empires" [30]. The first attack leverages the normal distribution of data and proposes gradients that lie within a small range containing the mean. The second attack focuses on inner product manipulation: all GARs require their inner product with the true gradient to be positive.

We obtained remarkable results with Aksel, especially on complex datasets (CIFAR10). In fact, Aksel, as any full-GAR, reaches top accuracy when $s << f$ (see Figure 2). It is also able to defend against the extreme case $s \sim f$ while maintaining a descent accuracy, thanks to its low angular error (see Figure 3). In some experiments, Aksel is the only GAR reaching the top accuracy while others never converge (Figure 4).

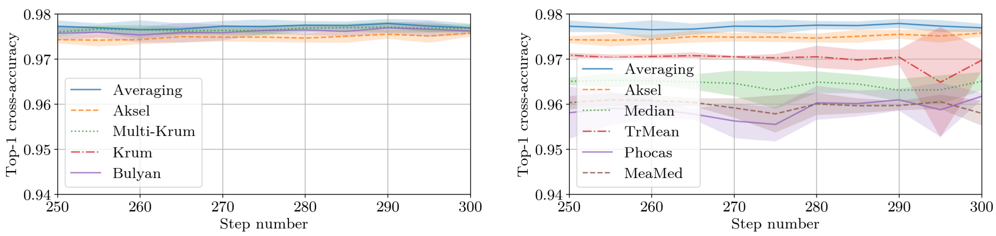

Figure 2 We compare Aksel and averaging ("No Byzantine resilience") to full-GARs (left) and blended-GARs (right) in an environment with no Byzantine worker. Here, Aksel, as well as other full-GARs, perform as well as averaging. (MNIST dataset, using $n = 51$ workers; the GARs are tuned to withstand up to 12 Byzantines workers.)

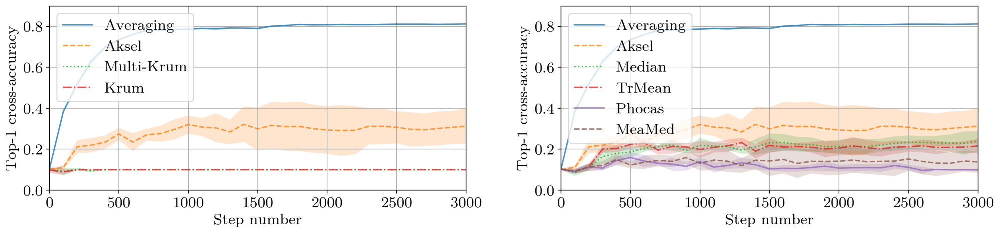

Figure 3 CIFAR-10 using $n = 25$ workers and $s = f = 11$ Byzantine workers implementing attack [30]. The learning rate schedule is 0.01 for the first 1500 training steps, then 0.001 for the remaining of the training.

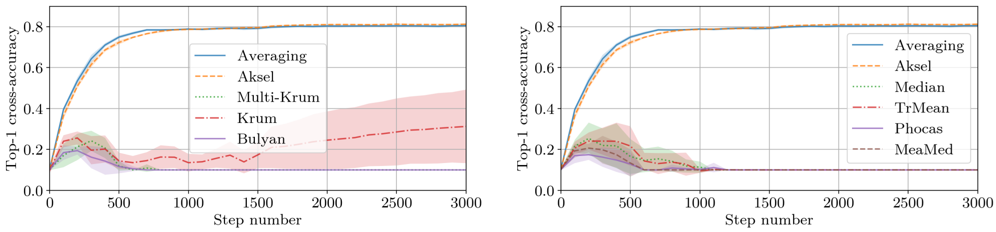

Figure 4 CIFAR-10 using $n = 25$ workers, including $s = f = 5$ Byzantine workers implementing attack [3]. The learning rate schedule is 0.01 for the first 1500 training steps, then 0.001 for the remaining of the training. Aksel is the only GAR which actually converges.

7 Concluding Remarks

Summary. This paper investigates the parameter server architecture of machine learning algorithms when trained in untrusted environments. We address time complexity, breakdown point, angular error and the overhead cost of Byzantine resilience. We propose Aksel, the first full gradient aggregation rule with optimal time complexity and optimal breakdown point with a constant expected angular error in the number of workers. Our empirical evaluation shows that AKSEL achieves top accuracy in frequent situations with none or few Byzantine workers, while maintaining a good defense in the very few cases where the ratio of Byzantine workers approaches 50%. We also provide a new upper bound on the angular error of median based GARs (Aksel included) which grows only in $\mathcal{O}(\sqrt{\frac{d}{n}})$ under optimal robustness.

Discussion. One could also ask whether it is possible to reduce the angular error of Aksel further and obtain that of averaging ($\mathcal{O}(\sqrt{\frac{d}{n}})$), which is not Byzantine resilient. We foresee two ways to improve the angular error: either by reducing the breakdown point, which would result in a interval around the median containing only correct workers (this is the main idea of b-TrMean and b-Phocas [29]), or by sacrificing the time complexity by computing the distance between the median and the closest possible Byzantine value, which should give an idea on how tight the filtering interval should be to average only correct workers. Note that if we replace Median in Aksel with b-TrMean, it is possible to reduce the expected angular error to $\mathcal{O}(\sqrt{\frac{d}{n}})$ when $f = \mathcal{O}(1)$. However, we prefer the current version of Aksel because it does not need the truncation parameter b which, if underestimated, can cause a serious problem in the training.

We see many ways to relax some of the assumptions we make in this paper. We believe for instance that the Byzantine resilience and the convergence analysis could be done using biased estimates, as in [6]. One could also derive an upper bound of the variance of gradients using the smoothness assumption, as discussed in [22], without assuming a constant upper bound σ^2 (as assumed in all previous papers). Another interesting direction is to leverage randomness to improve Byzantine resilience.

References

1 Martin Abadi et al. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 265–283, 2016.

2 Dan Alistarh, Zeyuan Allen-Zhu, and Jerry Li. Byzantine stochastic gradient descent. In *Advances in Neural Information Processing Systems*, pages 4613–4623, 2018.

3 Moran Baruch, Gilad Baruch, and Yoav Goldberg. A little is enough: Circumventing defenses for distributed learning, 2019. `arXiv:1902.06156`.

4 Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Advances in Neural Information Processing Systems 30*, pages 119–129. Curran Associates, Inc., 2017.

5 Léon Bottou. *On-Line Learning and Stochastic Approximations*, page 9–42. Cambridge University Press, USA, 1999.

6 Léon Bottou, Frank E. Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2):223–311, 2018.

7 Lingjiao Chen, Hongyi Wang, Zachary Charles, and Dimitris Papailiopoulos. DRACO: Byzantine-resilient distributed training via redundant gradients. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 903–912. PMLR, 2018.

8 Yudong Chen, Lili Su, and Jiaming Xu. Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 1(2):44, 2017.

9 Georgios Damaskinos, El Mahdi El Mhamdi, Rachid Guerraoui, Arsany Guirguis, and Sébastien Rouault. Aggregathor: Byzantine machine learning via robust gradient aggregation. In *SysML*, 2019.

10 Georgios Damaskinos, El Mahdi El Mhamdi, Rachid Guerraoui, Rhicheek Patra, Mahsa Taziki, et al. Asynchronous byzantine machine learning (the case of sgd). In *ICML*, pages 1153–1162, 2018.

11 El-Mahdi El-Mhamdi, Rachid Guerraoui, Arsany Guirguis, and Lê Nguyên Hoang. Geniunely distributed byzantine machine learning. In *PODC*, 2020.

12 El Mahdi El Mhamdi, Rachid Guerraoui, and Sébastien Rouault. The hidden vulnerability of distributed learning in Byzantium. In *Proceedings of the 35th International Conference on Machine Learning*, pages 3521–3530. PMLR, 2018.

13 Yuri Fateev, Vladimir Shaydurov, Evgeny Garin, Dmitry Dmitriev, and Valeriy Tyapkin. Probability distribution functions of the sum of squares of random variables in the non-zero mathematical expectations. *Journal of Siberian Federal University. Mathematics & Physics*, 9:173–179, 2016.

14 Matthias Függer and Thomas Nowak. Fast Multidimensional Asymptotic and Approximate Consensus. In *32nd International Symposium on Distributed Computing (DISC 2018)*, volume 121 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 27:1–27:16. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018.

15 E.J. Gumbel. *Statistics of Extremes*. Dover books on mathematics. Dover Publications, 2004.

16 Robert Hecht-Nielsen. Theory of the backpropagation neural network. In *Neural networks for perception*, pages 65–93. Elsevier, 1992.

17 C. A. R. Hoare. Algorithm 65: Find. *Commun. ACM*, 4(7):321–322, 1961.

18 Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pages 8571–8580, 2018.

19 S. Kotz and S. Nadarajah. *Extreme Value Distributions.* World Scientific Publishing Company, 2000.

20 Hammurabi Mendes, Maurice Herlihy, Nitin Vaidya, and Vijay Garg. Multidimensional agreement in byzantine systems. *Distributed Computing*, 28:1–19, 2015.

21 F. Mosteller and J.W. Tukey. *Data Analysis and Regression: A Second Course in Statistics.* Addison-Wesley Series in Behavioral Science, 1977.

22 Lam M. Nguyen, Phuong Ha Nguyen, Marten van Dijk, Peter Richtárik, Katya Scheinberg, and Martin Takáč. Sgd and hogwild! convergence without the bounded gradients assumption, 2018.

23 B. T. Polyak and A. B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization, 30(4):838–855*, 1992.

24 Peter J Rousseeuw. Multivariate estimation with high breakdown point. *Mathematical statistics and applications*, 8:283–297, 1985.

25 David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

26 Fred B. Schneider. Implementing fault-tolerant services using the state machine approach: A tutorial. *ACM Comput. Surv.*, 22(4):299–319, 1990.

27 V. Vapnik. Principles of risk minimization for learning theory. In J. E. Moody, S. J. Hanson, and R. P. Lippmann, editors, *Advances in Neural Information Processing Systems 4*, pages 831–838. Morgan-Kaufmann, 1992.

28 Cong Xie, Oluwasanmi Koyejo, and Indranil Gupta. Generalized byzantine-tolerant sgd, 2018.

29 Cong Xie, Oluwasanmi Koyejo, and Indranil Gupta. Phocas: dimensional byzantine-resilient stochastic gradient descent, 2018.

30 Cong Xie, Oluwasanmi Koyejo, and Indranil Gupta. Fall of empires: Breaking byzantine-tolerant sgd by inner product manipulation. In *UAI*, volume 115 of *Proceedings of Machine Learning Research*, pages 261–270. PMLR, 2020.

31 Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5650–5659. PMLR, 2018.