# ROBUST AND EFFICIENT AGGREGATION FOR DISTRIBUTED LEARNING

*Stefan Vlaski*[⋆]*, Christian Schroth*[†]*, Michael Muma*[†] *and Abdelhak M. Zoubir*[†]

[⋆]Department of Electrical and Electronic Engineering, Imperial College London, UK
[†]Signal Processing Group, Technische Universität Darmstadt, Germany

## ABSTRACT

Distributed learning paradigms, such as federated and decentralized learning, allow for the coordination of models across a collection of agents, and without the need to exchange raw data. Instead, agents compute model updates locally based on their available data, and subsequently share the update model with a parameter server or their peers. This is followed by an aggregation step, which traditionally takes the form of a (weighted) average. Distributed learning schemes based on averaging are known to be susceptible to outliers. A single malicious agent is able to drive an averaging-based distributed learning algorithm to an arbitrarily poor model. This has motivated the development of robust aggregation schemes, which are based on variations of the median and trimmed mean. While such procedures ensure robustness to outliers and malicious behavior, they come at the cost of significantly reduced sample efficiency. This means that current robust aggregation schemes require significantly higher agent participation rates to achieve a given level of performance than their mean-based counterparts in non-contaminated settings. In this work we remedy this drawback by developing statistically efficient and robust aggregation schemes for distributed learning.

***Index Terms*—** Distributed learning, robust aggregation, sample efficiency, malicious agents.

## 1. INTRODUCTION AND RELATED WORKS

We consider a general distributed learning problem, where a collection of $K$ agents aim to collaboratively solve a stochastic optimization problem defined through:

$$w^o \triangleq \arg\min_{w} \frac{1}{K}\sum_{k=1}^{K} \mathbb{E} Q(w; \boldsymbol{x}_k) \tag{1}$$

Here, $\boldsymbol{x}_k$ denotes a random variable describing the privately available data at agent $k$, and $Q(w; \boldsymbol{x}_k)$ denotes the associated loss. It will be convenient to define $J_k(w) \triangleq \mathbb{E} Q(w; \boldsymbol{x}_k)$ and $J(w) \triangleq \sum_{k=1}^{K} p_k J_k(w)$, so that:

$$J(w) = \frac{1}{K}\sum_{k=1}^{K} J_k(w) = \frac{1}{K}\sum_{k=1}^{K} \mathbb{E} Q(w; \boldsymbol{x}_k) \tag{2}$$

This formulation is general enough to cover a wide range of learning problems, from distributed least mean-squares and logistic regression [1] to distributed deep learning [2, 3].

Solutions to consensus optimization problems of the form (1) can be pursued through a number of distributed strategies, depending on resource and communication constraints. Broadly, algorithms for distributed learning can be classified into (a) fusion-center based strategies, and (b) fully-decentralized strategies. Fusion-center based strategies involve communication with a central parameter server, which performs aggregation of intermediate model estimates, and subsequently broadcasts them back to participating agents. Fully-decentralized approaches on the other hand rely purely on peer-to-peer exchanges over some (potentially sparse) graph topology.

**Example 1 – Federated learning:** Federated architectures rely on a central processor to coordinate computations, but avoid exchanges of raw data by allowing agents to locally compute updates of a common model in a highly asynchronous manner. A representative example is the federated averaging algorithm [4], where at each iteration $i$, a subset $\mathcal{N}$ of $N$ agents is chosen, and each agent is provided with the current version of the model $\boldsymbol{w}_{i-1}$ stored at the central parameter server. Each agent then initializes $\boldsymbol{\phi}_{k,0} = \boldsymbol{w}_{i-1}$ and performs $L_k$ steps of (stochastic) gradient descent by iterating over $j$:

$$\boldsymbol{\phi}_{k,j} = \boldsymbol{\phi}_{k,j-1} - \mu \widehat{\nabla J}_k(\boldsymbol{\phi}_{k,j-1}) \tag{3}$$

Here, $\mu > 0$ denotes the step-size and $\widehat{\nabla J}_k(\boldsymbol{\phi}_{k,j-1})$ corresponds to a stochastic gradient approximation of $J_k(w)$ based on the locally available data. Upon completion, each agent returns $\boldsymbol{\phi}_{k,L_k}$ to the parameter server, where the aggregate model is updated according to:

$$\boldsymbol{w}_i = \frac{1}{N}\sum_{k\in\mathcal{N}} \boldsymbol{\phi}_{k,L_k} \tag{4}$$

**Example 2 – Decentralized learning:** In contrast to federated approaches, decentralized learning algorithms rely

Emails: s.vlaski@imperial.ac.uk, {cschroth, mmuma, zoubir}@spg.tu-darmstadt.de

solely on peer-to-peer interactions between pairs of agents connected by some (potentially sparse) graph topology, and avoid the need for a central aggregator or coordinator. Similar to federated structures, these algorithms perform combinations of local updates steps, based on locally available data, and aggregation steps, with the difference being that instead of aggregating at a central processor, aggregation occurs locally over neighborhoods of agents based on peer-to-peer exchanges. Here, the neighborhood $\mathcal{N}_k$ of agent $k$ defines the set of agents, with which agent $k$ is willing and able to exchange information. An example is the ATC-diffusion algorithm, which takes the form [1]:

$$\boldsymbol{\phi}_{k,i} = \boldsymbol{w}_{k,i-1} - \mu \widehat{\nabla J}_k(\boldsymbol{w}_{k,i-1}) \tag{5}$$

$$\boldsymbol{w}_{k,i} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k} \boldsymbol{\phi}_{\ell,i} \tag{6}$$

Examining relations (4) and (6), we note that both federated and decentralized learning approaches rely on an averaging step of the form:

$$\boldsymbol{w}_{k,i} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k} \boldsymbol{\phi}_{\ell,i} = \arg\min_{\boldsymbol{w}} \sum_{\ell \in \mathcal{N}_k} a_{\ell k} \| \boldsymbol{\phi}_{\ell,i} - \boldsymbol{w} \|^2 \tag{7}$$

for some non-negative weights $a_{\ell k}$ that add up to one. This immediately makes clear the limited robustness of averaging-based schemes for distributed learning. Manipulating the value of a single $\boldsymbol{\phi}_{\ell,i}$, either for benign or malicious reasons, has the potential to influence the aggregate model $\boldsymbol{w}_{k,i}$ arbitrarily. This has motivated increased interest over recent years on robust alternatives to the aggregation scheme (7). An example is the secure aggregation protocol of [5] based on the geometric median (also known as spatial median), which takes the form:

$$\boldsymbol{w}_{k,i} = \arg\min_{\boldsymbol{w}} \sum_{\ell \in \mathcal{N}_k} a_{\ell k} \| \boldsymbol{\phi}_{\ell,i} - \boldsymbol{w} \| \tag{8}$$

Variations based on element-wise median/trimmed-mean have also been considered [6]. The authors of [7] consider a more elaborate procedure termed "Krum", which nevertheless discards a majority of (potentially) benign samples. While these approaches yield increased robustness to perturbations in $\boldsymbol{\phi}_{\ell,i}(m)$ up to a contamination rate of 50%, employing the median in place of the mean results in reduced sample efficiency, resulting in a drop in performance relative to averaging-based approaches in the absence of adversaries. While this fact is acknowledged in the literature [7], it is generally accepted as a necessary price to pay for the guarantee of robustness in the presence of adversaries. An alternative based on $\ell_p$-norm penalization of deviation from consensus is presented in [8]; a similar formulation in the context of multi-task learning appears in [9].

The aforementioned works [5–8] focus on centralized or federated learning in the presence of a fusion center. Generalizations to the decentralized setting of trimmed-mean, median and Krum based approaches have been provided in [10, 11], and of the penalty based RSA-approach in [12]. We note that other works, such as [13], have considered the problem of distributed robust estimation by networked agents. Here, a collection of benign agents, all following a prescribed learning protocol, aim to learn collaboratively from contaminated data. Robustness in this context is achieved by adjusting the update (5), rather than the aggregation scheme (6).

## 2. M- AND MM-BASED AGGREGATION

Both (7) and (8) can be viewed as instances of the more general M-estimation problem [14, 15]:

$$\boldsymbol{w}_{k,i} = \arg\min_{\boldsymbol{w}} \sum_{\ell \in \mathcal{N}_k} a_{\ell k} \rho^{\text{agg}} \left( \boldsymbol{\phi}_{\ell,i} - \boldsymbol{w} \right) \tag{9}$$

The choice $\rho^{\text{agg}}(\cdot) = \| \cdot \|^2$ yields the ordinary average, with high efficiency, but low robustness, while the choice $\rho^{\text{agg}}(\cdot) = \| \cdot \|$ yields the geometric median, with high robustness, but low efficiency. Letting $\rho^{\text{agg}}(\cdot) = \| \cdot \|_1$ on the other hand yields the elementwise median. Different choices of $\rho^{\text{agg}}(\cdot)$ allow for the trade-off of robustness and efficiency. For simplicity, we will be focusing on loss functions $\rho^{\text{agg}}(\cdot)$, which operate elementwise on their argument, which will in turn translate into elementwise aggregation schemes. For such $\rho^{\text{agg}}(\cdot)$, we have:

$$\sum_{\ell \in \mathcal{N}_k} a_{\ell k} \rho^{\text{agg}} \left( \boldsymbol{\phi}_{\ell,i} - \boldsymbol{w} \right) = \sum_{\ell \in \mathcal{N}_k} a_{\ell k} \sum_{m=1}^{M} \rho \left( \boldsymbol{\phi}_{\ell,i}(m) - \boldsymbol{w}(m) \right) \tag{10}$$

Popular choices for the penalty function $\rho(\cdot)$ include monotone choices such as the Huber loss and redescending ones such as the Tukey's bisquare function — for a detailed discussion on robust loss functions for location estimation we refer the reader to [14]. An alternative formulation of (9) follows after differentiating:

$$\sum_{\ell \in \mathcal{N}_k} a_{\ell k} \psi \left( \boldsymbol{\phi}_{\ell,i}(m) - \boldsymbol{w}_{k,i}(m) \right) = 0 \tag{11}$$

where $\psi(\cdot) = \rho'(\cdot)$ is the derivative of the loss. If we define:

$$b(y) \triangleq \begin{cases} \frac{\psi(y)}{y} & \text{if } y \neq 0, \\ \psi'(0) & \text{if } y = 0. \end{cases} \tag{12}$$

it follows that after algebraic manipulation that [14]:

$$\boldsymbol{w}_{k,i}(m) = \frac{\sum_{\ell \in \mathcal{N}_k} a_{\ell k} b \left( \boldsymbol{\phi}_{\ell,i}(m) - \boldsymbol{w}_{k,i}(m) \right) \boldsymbol{\phi}_{\ell,i}(m)}{\sum_{\ell \in \mathcal{N}_k} a_{\ell k} b \left( \boldsymbol{\phi}_{\ell,i}(m) - \boldsymbol{w}_{k,i}(m) \right)} \tag{13}$$

If we define:

$$\overline{\boldsymbol{a}}_{\ell k}(m) \triangleq \frac{a_{\ell k} b \left( \boldsymbol{\phi}_{\ell,i}(m) - \boldsymbol{w}_{k,i}(m) \right)}{\sum_{\ell \in \mathcal{N}_k} a_{\ell k} b \left( \boldsymbol{\phi}_{\ell,i}(m) - \boldsymbol{w}_{k,i}(m) \right)} \tag{14}$$

this gives rise to the representation:

$$\boldsymbol{w}_{k,i}(m) = \sum_{\ell\in\mathcal{N}_k} \overline{\boldsymbol{a}}_{\ell k}(m)\boldsymbol{\phi}_{\ell,i}(m) \tag{15}$$

Relation (15) indicates that robust aggregation via M-estimation can be interpreted as a convex combination of prior estimates $\boldsymbol{\phi}_{\ell,i}(m)$ with weights $\overline{\boldsymbol{a}}_{\ell k}(m)$, which are obtained by modulating $a_{\ell k}$ with $b\left(\boldsymbol{\phi}_{\ell,i}(m) - \boldsymbol{w}_{k,i}(m)\right)$. Here, $b\left(\boldsymbol{\phi}_{\ell,i}(m) - \boldsymbol{w}_{k,i}(m)\right)$ measures the likelihood that the estimate obtained from neighbor $\ell$ is an outlier. It is worth noting that while (15) indicates that $\boldsymbol{w}_{k,i}(m)$ is a convex combination of $\boldsymbol{\phi}_{\ell,i}(m)$, this relationship is not prescriptive, nor does it imply that it is linear. This is because $\overline{\boldsymbol{a}}_{\ell k}(m)$ is an implicit function of the prior estimates $\boldsymbol{\phi}_{\ell,i}(m)$ as well as the resulting estimate $\boldsymbol{w}_{k,i}(m)$. In practice, M-estimates are pursued by fixed-point iterations, which return the weights $\overline{\boldsymbol{a}}_{\ell k}(m)$ as a byproduct – we refer the reader to [14] for details.

Classical M-estimators trade off robustness and statistical efficiency via the choice of the loss function $\rho(\cdot)$. Simultaneous robustness and efficiency can be achieved as well by utilizing a nested procedure where a robust, but not efficient, estimate of location and scale is used to initialize and normalize the fixed-point recursion of a subsequent M-estimator leading to (15). The resulting procedure is known as MM-estimation, and preserves the robustness of the initialization, while inheriting the statistical efficiency of the subsequent M-estimation [14]. In particular, MM-estimators can exhibit tolerance of close to 50% outliers, while having efficiency close to that of the maximum likelihood estimate. We can then integrate the MM-based aggregator into our distributed learning framework to obtain the proposed algorithm, termed REF-Diffusion for "Robust-and -Efficient Diffusion":

```
Algorithm 1: REF-Diffusion Strategy
  Step 1: At each agent $k$, collect $\boldsymbol{x}_{k,i}$ and update:
      $\boldsymbol{\phi}_{k,i} = \boldsymbol{w}_{k,i-1} - \mu\widehat{\nabla J}_k(\boldsymbol{w}_{k,i-1})$      (16)
  Step 2: Collect $\left\{\boldsymbol{\phi}_{\ell,i}\right\}_{\ell\in\mathcal{N}_k}$, and compute $\overline{\boldsymbol{a}}_{\ell k}(m)$
      for $m = 1, \dots, M$ using a robust and efficient
      MM-procedure.
  Step 3: Aggregate via (15) for $m = 1, \dots, M$.
```

## 3. ANALYSIS

### 3.1. Modeling Conditions

The set of agents $\mathcal{N}$ is decomposed into two sets. The collection of benign agents is denoted by $\mathcal{N}^b$, while the set of malicious agents is denoted by $\mathcal{N}^m$. Benign agents in $\mathcal{N}^b$ follow the learning and aggregation procedures in Algorithm 1 faithfully, while agents in $\mathcal{N}^m$ may deviate arbitrarily. For each agent $k$, we similarly denote by $\mathcal{N}_k^b$ the benign agents within the neighborhood $\mathcal{N}_k$ of agent $k$, and by $\mathcal{N}_k^m$ the malicious agents within that same neighborhood.

**Assumption 1** (**Contamination Rate**). *For each benign agent $k \in \mathcal{N}^b$, the majority of agents in its neighborhood are benign. Specifically:*

$$\frac{\left|\mathcal{N}_k^b\right|}{|\mathcal{N}_k|} > 1 - \epsilon \tag{17}$$

*Here, $|\cdot|$ denotes the cardinality of a set, and $0 \leq \epsilon < \frac{1}{2}$ represents an upper bound on the fraction of malicious agents. Furthermore, the collection of benign agents $\mathcal{N}^b$ form a connected subgraph of the full network $\mathcal{N}$.* □

Assumption (1) ensures that the majority of agents within each neighborhood are benign, and that the remaining network after removing malicious agents remains connected. Such conditions are standard in the development of robust decentralized algorithms [12]. Next, we introduce a condition on the MM-estimator:

**Assumption 2** (**Robust Aggregator**). *The MM-estimator yielding the weights $\overline{\boldsymbol{a}}_{\ell k}(m)$ is robust and efficient with breakdown points greater than $\epsilon$.* □

Finally, we impose standard conditions on the loss functions of benign agents as well as the accuracy of the gradient approximation $\widehat{\nabla J}_k(\boldsymbol{w}_{k,i-1})$: [1, 16, 17]:

**Assumption 3** (**Lipschitz Gradients**). *For each $k$, the gradient $\nabla J_k(\cdot)$ is Lipschitz, namely, there exists $\delta \geq 0$ such that for any $x, y \in \mathbb{R}^M$:*

$$\|\nabla J_k(x) - \nabla J_k(y)\| \leq \delta\|x - y\| \tag{18}$$

□

**Assumption 4** (**Strong Convexity**). *For each $k$, the cost $J_k(\cdot)$ is $\nu$-strongly convex, i.e., for every $x, y \in \mathbb{R}^M$:*

$$(x - y)^{\mathsf{T}} \left(\nabla J_k(x) - \nabla J_k(y)\right) \geq \nu\|x - y\|^2 \tag{19}$$

□

**Assumption 5** (**Gradient Noise Process**). *For each $k$, the gradient noise process is defined as*

$$\boldsymbol{s}_{k,i}(\boldsymbol{w}_{k,i-1}) = \widehat{\nabla J}_k(\boldsymbol{w}_{k,i-1}) - \nabla J_k(\boldsymbol{w}_{k,i-1}) \tag{20}$$

*and satisfies*

$$\mathbb{E}\left[\boldsymbol{s}_{k,i}(\boldsymbol{w}_{k,i-1})|\boldsymbol{\mathcal{F}}_{i-1}\right] = 0 \tag{21}$$

$$\mathbb{E}\left[\|\ \boldsymbol{s}_{k,i}(\boldsymbol{w}_{k,i-1})\|^2|\boldsymbol{\mathcal{F}}_{i-1}\right] \leq \beta^2\|w^o - \boldsymbol{w}_{k,i-1}\ \|^2 + \sigma^2 \tag{22}$$

*for some non-negative constants $\{\beta^2, \sigma^2\}$, and where $\boldsymbol{\mathcal{F}}_{i-1}$ denotes the filtration generated by the random processes $\{\boldsymbol{w}_{\ell,j}\}$ for all $\ell = 1, 2, \dots, K$ and $j \leq i - 1$.* □

### 3.2. Convergence Analysis

Assumptions 1 and 2 ensure that the number of malicious agents within each neighborhood is smaller than the breakdown point of the MM-estimator driving the aggregation procedure. This ensures that the aggregate $\boldsymbol{w}_{k,i}$ obtained from (15) provides a meaningful estimate of the mean of $\{\boldsymbol{\phi}_{\ell,i}\}_{\ell \in \mathcal{N}_k^b}$ over the set of *benign* agents. Specifically, one expects for an efficient estimator that:

$$b\left(\boldsymbol{\phi}_{\ell,i}(m) - \boldsymbol{w}_{k,i}(m)\right) \approx \begin{cases} 1, \text{ if } \ell \in \mathcal{N}_k^b, \\ 0, \text{ if } \ell \in \mathcal{N}_k^m. \end{cases} \tag{23}$$

This translates to:

$$\overline{\boldsymbol{a}}_{\ell k}(m) \approx \overline{a}_{\ell k} \triangleq \begin{cases} \frac{a_{\ell k}}{\sum_{\ell \in \mathcal{N}_k^b} a_{\ell k}}, \text{ if } \ell \in \mathcal{N}_k^b, \\ 0, \text{ if } \ell \in \mathcal{N}_k^m. \end{cases} \tag{24}$$

In other words, the effective weights $\overline{a}_{\ell k}$ of benign agents are obtained by scaling the original weights $a_{\ell k}$, to account for the fact that the effective weights $\overline{a}_{\ell k}$ of malicious agents are set to zero. This ensures that effective weights continue to add up to one. Under this approximation, we can write Algorithm 1 as:

$$\boldsymbol{\phi}_{k,i} = \boldsymbol{w}_{k,i-1} - \mu \widehat{\nabla J}_k(\boldsymbol{w}_{k,i-1}) \tag{25}$$

$$\boldsymbol{w}_{k,i} \approx \sum_{\ell \in \mathcal{N}_k^b} \overline{a}_{\ell k} \boldsymbol{\phi}_{k,i-1} \tag{26}$$

Comparing (25)–(26) with the classical diffusion strategy (5)–(6), we note two differences. First, the aggregation step (26) involves averaging only over the set of benign agents $\mathcal{N}_k^b$ within $\mathcal{N}_k$, and second the weights $\overline{a}_{\ell k}$ are adjusted from $a_{\ell k}$. The adjacency matrix $[\overline{A}]_{\ell k} \triangleq \overline{a}_{\ell k}$ can be decomposed as:

$$\overline{A} = \begin{pmatrix} \overline{A}^b & 0 \\ 0 & 0 \end{pmatrix} \tag{27}$$

where $\overline{A}^b$ contains the weights $\overline{a}_{\ell k}$ of benign agents $\ell \in \mathcal{N}^b$. Assumption 1, in light of the Perron-Frobenius theorem [18], then ensures that $\overline{A}^b$ is a *primitive* matrix with a single eigenvalue at one and corresponding eigenvector $\overline{p}^b$, which can be normalized to satisfy:

$$\overline{A}^b \overline{p}^b = \overline{p}^b, \quad \overline{p}^b(k) > 0 \, \forall \, k, \quad \sum_{k \in \mathcal{N}^b} \overline{p}^b(k) = 1 \tag{28}$$

We can then appeal to known results on the convergence of the non-robust diffusion strategy [1, Theorem 9.1] to conclude:

**Theorem 1** (**Limiting Behavior**). *Suppose Assumptions 1–5 hold, and the approximation* (23) *is accurate. Then, the limiting point of Algorithm 1 is determined by the data* $\boldsymbol{x}_k$ *of* benign *agents* $\mathcal{N}^b$ *through:*

$$\overline{w}^o \triangleq \arg\min_w \sum_{k \in \mathcal{N}^b} \overline{p}^b(k) \mathbb{E} Q(w; \boldsymbol{x}_k) \tag{29}$$

*We have for all* $k \in \mathcal{N}^b$*:*

$$\limsup_{i \to \infty} \mathbb{E} \|\overline{w}^o - \boldsymbol{w}_{k,i}\|^2 = O(\mu) \tag{30}$$

*for sufficiently small step-size* $\mu$*.*

## 4. NUMERICAL RESULTS

We consider a collection of $K = 32$ agents, connected through a fully connected graph. Each agent observes data following a linear model:

$$\boldsymbol{d}_k = \boldsymbol{u}_k^{\mathsf{T}} w^o + \boldsymbol{v}_k \tag{31}$$

where the regressors $\boldsymbol{u}_k \in \mathbb{R}^{10}$ are identically normally distributed with $\boldsymbol{u}_k \sim \mathcal{N}(0, I_{10})$. The noise term $\boldsymbol{v}_k$ is also normally distributed with $\boldsymbol{v}_k \sim \mathcal{N}(0, \sigma_v^2)$ and $\sigma_v^2 = 0.01$. Each agent is equipped with the mean square error cost:

$$J_k(w) = \frac{1}{2} \mathbb{E} \|\boldsymbol{d}_k - \boldsymbol{u}_k^{\mathsf{T}} w\|^2 \tag{32}$$

and constructs the gradient approximation:

$$\widehat{\nabla J}_k(w) \triangleq \boldsymbol{u}_k \left(\boldsymbol{d}_k - \boldsymbol{u}_k^{\mathsf{T}} w\right) \tag{33}$$

It can be readily verified that this formulation satisfies Assumption 3 through 5. Benign agents follow the prescribed learning and aggregation schemes. The proposed scheme of Algorithm 1 is implemented through an M-estimator with Tukey's biweight loss function [14], initialized and normalized with robust location and scale estimates through the median and median absolute deviation respectively. The implementation is taken from the repository of [15], available publicly on Github. Performance is compared to the baseline averaging-based approach [1] and elementwise median aggregation [6]. A variable number of malicious agents deviate from the prescribed learning protocol by additively perturbing their local update via:

$$\boldsymbol{\phi}_{k,i} = \boldsymbol{w}_{k,i-1} - \mu \widehat{\nabla J}_k(\boldsymbol{w}_{k,i-1}) + \boldsymbol{\Delta} \tag{34}$$

where $\boldsymbol{\Delta} = \delta \mathbb{1}$.

We show in in the left column of Fig. 1 the mean-square deviation from $w^o$ for a single malicious agent, as a function of both iteration and contamination strength $\delta$. In the right column of Fig. 1 we show mean-square deviation for a fixed contamination strength $\delta = 1000$ as a function of both iteration and rate of contamination.

## 5. CONCLUSION

We have presented REF-Diffusion, an algorithm for robust and efficient learning over networks. The strategy is derived by replacing traditional averaging- or median-based aggregation procedures by an MM-estimate of location, which can be

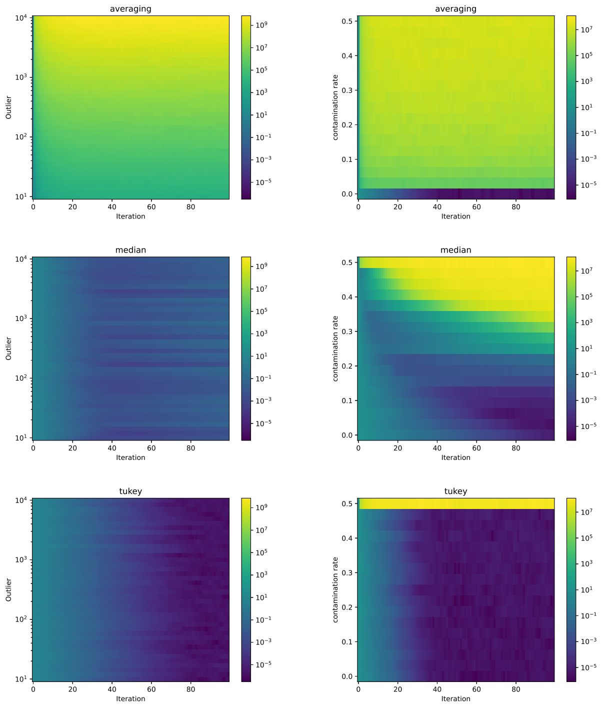

**Fig. 1**: Performance over time and contamination strength $\delta$ for a single malicious agent (left) and performance over time and contamination rate for a fixed strength (right).

designed to be simultaneously robust and efficient. The result is a strategy which performs on par with averaging-based approaches in the absence of deviating agents, while preserving robustness in the presence of perturbations. Numerical results corroborate the claims.

## 6. REFERENCES

[1] A. H. Sayed, "Adaptation, learning, and optimization over networks," *Foundations and Trends in Machine Learning*, vol. 7, no. 4-5, pp. 311–801, July 2014.

[2] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu, "Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent," in *Advances in Neural Information Processing Systems 30*, 2017, pp. 5330–5340.

[3] S. Vlaski and A. H. Sayed, "Distributed learning in non-convex environments – Part II: Polynomial escape from saddle-points," *IEEE Transactions on Signal Processing*, vol. 69, pp. 1257–1270, 2021.

[4] J. Konecny, H. B. McMahan, F. X. Yu, P. Richtarik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," in *NIPS Workshop on Private Multi-Party Machine Learning*, 2016.

[5] S. M. Kakade K. Pillutla and Z. Harchaoui, "Robust aggregation for federated learning," *available as arXiv:1912.13445*, Dec 2019.

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

[7] P. Blanchard, E. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Advances in Neural Information Processing Systems*, 2017, vol. 30.

[8] L. Li, W. Xu, T. Chen, G. B. Giannakis, and Q. Ling, "RSA: byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets," in *The Thirty-Third AAAI Conference on Artificial Intelligence*, 2019, pp. 1544–1551.

[9] Y. SarcheshmehPour, Y. Tian, L. Zhang, and A. Jung, "Networked federated multi-task learning," *available as arXiv:2105.12769*, May 2021.

[10] C. Fang, Z. Yang, and W. U. Bajwa, "BRIDGE: Byzantine-resilient decentralized gradient descent," *available as arXiv:1908.08098*, Aug 2022.

[11] Z. Yang, A. Gang, and 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 Processing Magazine*, vol. 37, no. 3, pp. 146–159, 2020.

[12] J. Peng, W. Li, and Q. Ling, "Byzantine-robust decentralized stochastic optimization over static and time-varying networks," *Signal Processing*, vol. 183, pp. 108020, 2021.

[13] S. Al-Sayed, A. M. Zoubir, and A. H. Sayed, "Robust distributed estimation by networked agents," *IEEE Transactions on Signal Processing*, vol. 65, no. 15, pp. 3909–3921, 2017.

[14] R.A. Maronna, D.R. Martin, and V.J. Yohai, *Robust Statistics: Theory and Methods*, Wiley Series in Probability and Statistics. Wiley, 2006.

[15] A. M. Zoubir, V. Koivunen, E. Ollila, and M. Muma, *Robust Statistics for Signal Processing*, Cambridge University Press, 2018.

[16] A. H. Sayed, "Adaptive networks," *Proceedings of the IEEE*, vol. 102, no. 4, pp. 460–497, April 2014.

[17] J. Chen and A. H. Sayed, "On the learning behavior of adaptive networks - Part I: Transient analysis," *IEEE Transactions on Information Theory*, vol. 61, no. 6, pp. 3487–3517, June 2015.

[18] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, 2003.