Robust Distributed Learning and Robust Learning Machines

El Mahdi El Mhamdi

Research Statement

Whether it occurs in artificial or biological substrates, *learning* is a distributed phenomenon in at least two aspects. First, meaningful data and experiences are rarely found in one location, hence *learners* have a strong incentive to work together, through *distributed learning*. Second, a learner is itself a distributed system, a *learning machine*, made of more basic processes; the change in the connections between these basic processes is what allows learning. This high-level abstraction encompasses a large set of learning situations, from nervous systems, to metabolic networks in an organism, to data centers, where several machines are working together to recommend personalized content for a billion-user social media.

In both levels of distribution, a system's ability to cope with the failure of some of its components is crucial. My research explores the robustness of learning systems from two perspectives. The first aspect is *coarse-grained*, considering the unit of failure as a whole learner. The second is *fine-grained*, considering the unit of failure as the basic component of the learner (e.g. a neuron or a synapse).

(Poorly Designed) AI is Harmful, *Already*

If you ask the general public about AI safety today, you will probably hear about killer robots or rogue self-driving cars. AI safety is seen as a concern that will arise in the **future**. In fact, AI safety is not just an important future concern, the lack of safety in **already deployed** learning systems is already a threat. Technical AI safety research is a pressing concern for the **present**.

Today's arguably most influential learning systems in our society are the recommender systems of social media. Every hour, 30 000 hours worth of videos are uploaded to a platform such as YouTube[1]. YouTube then has to moderate, sort and rank the content in order to make recommendations to users. Reportedly, there are about 5 million views per minute on YouTube, more than there are searches on Google and about five times more than there are scrolls on Facebook. Most importantly, 70% of these views are recommended by the platform and not directly searched by the viewer[2]. Manually programming a recommender system, such as the one used by YouTube, is an intractable task given its complexity. Instead, a learning system is leveraging users data to offer meaningful recommendations. One particular vulnerability of learning systems is that, since they learn from data, they are prone to *data poisoning*. Poisoning describes the act of degrading the quality of a learning system by feeding it maliciously mislabelled data, e.g. a an anti-vaccine video labeled as *health advice*[3].

From fuelling an ongoing ethnic cleansing[4] in Myanmar[5] to enabling interference in the elections of the country with the largest military budget in the world[6], the past three years gave us numerous cases where the amplification effect of social media calls for urgent research on securing learning systems from poisoning. Poisoning is only one among many other pressing questions in AI safety, but it is probably the most urgent motivation for the type of research that the larger part of my thesis tackled.

Robust Distributed Learning

Today's most influential deployments of machine learning systems are distributed for efficiency and scalability. Distribution however induces a higher risk of failures during the training phase. These include crashes and computation errors, stalled processes, biases in the way the data samples are distributed among the processes, but also, in the worst case, attackers trying to compromise the entire system and make it learn meaningless or harmful values.

[1] YouTube by the Numbers: `https://www.youtube.com/about/press/`

[2] According to Neal Mohan, YouTube's chief product officer. `https://www.cnet.com/news/youtube-ces-2018-neal-mohan/`

[3] 2019 was a good year in terms of awareness. In a positive development, most platforms acknowledged their vulnerability to poisoning and role in the aforementioned social issues while initiating research to combat these vulnerabilities. Platforms such as YouTube, Facebook, Twitter or Pinterest explicitly mentioned phony medical advice or anti-vaccine resentment in their different statements.

[4] The United Nations Human Rights Council: Report of Independent International Fact-Finding Mission on Myanmar (08/27/2018).

[5] *"We were too slow to respond to the concerns raised by civil society, academics and others in Myanmar"*. Mia Garlick, Facebooks director of Asia Pacific policy told Reuters. `https://www.reuters.com/investigates/special-report/myanmar-facebook-hate/`

[6] U.S. House of Representatives Permanent Select Committee on Intelligence: The Internet Research Agency and Advertisements. `https://intelligence.house.gov/social-media-content/`

The most robust system is one that tolerates *Byzantine* failures, i.e., completely arbitrary, potentially malicious behaviors of some of the processes.

A classical approach to mask failures in distributed systems is to use a state machine replication protocol, which requires however totally ordered state transitions to be applied by all processes. In the case of distributed machine learning, this constraint can be translated in two ways: either *(a)* the processes agree on a sample of data based on which they update their local parameter vectors, or *(b)* they agree on how the parameter vector should be updated. In case *(a)*, the sample of data has to be transmitted to each process, which then has to perform a heavyweight computation to update its local parameter vector. This entails communication and computational costs that defeat the entire purpose of distributing the work. Not mentioning the complete loss of privacy by sharing the data. In case *(b)*, the processes have no way to check if the chosen update for the parameter vector has indeed been computed correctly on real data: a Byzantine process could have proposed the update and may easily prevent the convergence of the learning algorithm. Neither of these solutions is satisfactory in a realistic distributed machine learning setting.

The first (and larger) part of my work so far focused on the *coarse-grained* aspect. I studied the robustness of distributed Stochastic Gradient Descent (SGD is arguably the work-horse algorithm behind many of today's machine learning successes). I began by proving [2, 3] that the standard deployment of SGD today is brittle, as this deployment typically consists of *averaging* the inputs from each learner. This leads to harmful consequences, as the data that is used in machine learning comes from different and potentially unreliable sources. To account for the various types of failures (data poisoning, malicious users, software bugs, communication delays, hacked machines etc.), we adopt the general abstraction of arbitrary failures in distributed systems, namely, *Byzantine failures*. We provide a sufficient condition for SGD to be Byzantine resilient and present three algorithms that satisfy our condition under different constraints.

The key algorithms that were introduced by my work are (1) Krum [2, 3], a gradient aggregation rule (GAR) that we prove to be a robust alternative to averaging in synchronous settings; (2) Bulyan [19], a meta-algorithm that we prove to strengthen any given GAR in very high dimensional situations and (3) Kardam [6], a gradient filtering scheme that we prove to be Byzantine resilient in the more challenging asynchronous setting. Some of these algorithms are now available as system implementations over TensorFlow [5] and PyTorch [7].

Robust Learning Machines

What about the robustness of the *decision making algorithms* themselves, once learning has happened?

The most powerful of these decision making algorithms today turn out to be the ones that contain many simple components and parameters. Among these, (artificial) Neural Networks (NNs) are responsible for the current success and regain of interest in machine learning. These networks are very loosely inspired by the higher vertebrates brain. Comprised of simple computing units, *neurons*, connected with simple links, *synapses*, NNs learn by modifying the *weights* of these links.

Biological plausibility, together with scalability, call for going one step further and considering each neuron as a *single* physical entity (that can fail *independently*), i.e., to consider genuinely distributed neural networks. This approach is considered for example in brain simulation projects, trying to emulate the brain, or the work on hardware-based neural networks, called *neuromorphic* chips. Recently, teams from IBM reported a successful neuromorphic implementation of neural networks that require a running power as low as 25 mW to 275 mW. In those settings, the unit of failure is one single neuron or synapse, not a whole machine as in the previous section.

The second part of my work went down to the *fine-grained* aspect. I focused on the special case of (artificial) neural networks. I view these networks as weighted directed graphs and prove upper bounds [9, 18, 16] on the *forward propagated error* when the basic components (neurons and synapses) are failing. Interestingly, this line of work found application [21] on the robustness of other systems such as biological (metabolic) networks.

Main Results

Synchronous Byzantine Resilient SGD [2, 3, 11]

We studied [2, 3] the resilience to Byzantine failures of distributed implementations of Stochastic Gradient Descent (SGD). So far, distributed machine learning algorithms have largely overlooked the possibility of failures, especially arbitrary (i.e., Byzantine) ones. Causes of failures include software bugs, network asynchrony, biases in local datasets, as well as attackers trying to compromise the entire system. Assuming a set of n machines (workers), up to f being Byzantine, we ask how resilient can SGD be, without limiting the dimension, nor the size of the parameter space. We first show that no gradient aggregation rule based on a linear combination of the vectors proposed by the workers (i.e, current approaches) tolerates even a single Byzantine failure. We then formulate a resilience property of the aggregation rule, capturing the basic requirements to guarantee convergence despite f

Byzantine workers. We propose *Krum*, an aggregation rule that satisfies our resilience property. We also report on experimental evaluations of Krum. Krum was the first provably Byzantine-resilient algorithm for distributed SGD. We also proved [11] improvements on the speed of Krum.

High Dimensional Vulnerabilities in Distributed Non-Convex Optimization [19]

Some of the approaches to secure SGD (including ours, mentioned in th previous section) have been proven *Byzantine–resilient*: they ensure the *convergence* of SGD despite the presence of a minority of adversarial workers. We show in this part of the work that *convergence is not enough*. In high dimension $d \gg 1$, an adversary can build on the loss function's non–convexity to make SGD converge to *ineffective* models. More precisely, we bring to light [19, 11] that existing Byzantine–resilient schemes leave a *margin of poisoning* (leeway) of $\Omega\left(f(d)\right)$, where $f(d)$ increases at least like $\sqrt{d}$. Based on this leeway, we build a simple attack, and experimentally show its strong to utmost effectivity. We introduce *Bulyan*, and prove it significantly reduces the attacker's leeway to a narrow $\mathcal{O}(\frac{1}{\sqrt{d}})$ bound. We empirically show that Bulyan does not suffer the fragility of existing aggregation rules and, at a reasonable cost in terms of required batch size, achieves convergence *as if* only non–Byzantine gradients had been used to update the model.

Asynchronous Byzantine Resilient SGD [6]

Asynchronous distributed machine learning solutions have proven very effective so far, but always assuming perfectly functioning workers. In this part of my work [6], we introduce *Kardam*, the first distributed asynchronous stochastic gradient descent (SGD) algorithm that copes with Byzantine workers. Kardam consists of two complementary components: a filtering and a dampening component. The first is scalar-based and ensures resilience against $\frac{1}{3}$ Byzantine workers. Essentially, this filter leverages the Lipschitzness of cost functions and acts as a self-stabilizer against Byzantine workers that would attempt to corrupt the progress of SGD. The dampening component bounds the convergence rate by adjusting to stale information through a generic gradient weighting scheme. We prove that Kardam guarantees almost sure convergence in the presence of asynchrony and Byzantine behavior, and we derive its convergence rate. We evaluate Kardam on the CIFAR-100 and EMNIST datasets and measure its overhead with respect to non Byzantine-resilient solutions. We empirically show that Kardam does not introduce additional noise to the learning procedure but does induce a slowdown (the cost of Byzantine resilience) that we both theoretically and empirically show to be less than f/n. Interestingly, we also empirically observe that the dampening component is interesting in its own right for it enables to build an SGD algorithm that outperforms alternative staleness-aware asynchronous competitors in environments with honest workers.

Systems Deployment and Practical Consequences [5, 7].

With the help of talented practitioner colleagues, my algorithmic and theoretical works, (which were described in the previous sections) have led to the deployment of the first Byzantine resilient gradient descent on top of well established ML frameworks. Our deployment on TensorFlow (Google's framework), described in [5] is now open-sourced and has received the ACM badges for re-usability/reproducibility. We also worked [7] on a multi-framework deployment (to include Facebook's PyTorch) that also supports failing servers. It is worth noting that one of the many practical consequences of Byzantine resilience is that our deployment [5] (thanks to its Byzantine resilience) is also the first deployment of TensorFlow that supports communication over UDP instead of TCP, hence being not only more secure, but also **6 times faster** than standard deployments on saturated (lossy) networks.

Neural Networks as a Distributed System [9, 16, 18]

In this part, we view a multilayer neural network as a distributed system of which neurons can fail independently, and we evaluate its robustness in the absence of any (recovery) learning phase. We give tight bounds on the number of neurons that can fail without harming the result of a computation. To determine our bounds, we leverage the fact that neural activation functions are (often) Lipschitz-continuous. Our bound applies on a quantity [9, 18], we call the *Forward Error Propagation* (FEP). FEP captures how much error is propagated by a neural network when a given number of components is failing. Computing FEP only requires looking at the topology of the network, while experimentally assessing the robustness of a network requires the costly experiment of looking at all the possible inputs and testing all the possible configurations of the network corresponding to different failure situations. The latter is prevented by a discouraging combinatorial explosion [16].

Other Works

I also explored two other categories of questions that fall broadly into the robustness of distributed and/or learning systems.

(1) Computational questions with a biological motivation. In this front, we looked on how metabolic networks could be approximated as a weighted directed graph, and how we could be inspired by the mathematical models of error propagation developed in [9, 18, 16]. This has led to a collaboration with colleagues from the Molecular Biology Department of the Johns Hopkins School of Medicine, that shed light on long-standing biological questions on gene essentiality [21].

In an other project with my colleagues at EPFL, we asked [17] whether the collective pattern formation in, e.g. gathering of fish schools, can be better explained by learning mechanism, instead of the traditionally accepted imperative algorithms, while showing robust behavior to the removal of individual processes.

(2) Technical questions in AI safety, ethics and social issues. In this front, we covered questions such as the safe interruptibility of reinforcement learning (RL). More precisely, we were the first to formalize and provide solutions for safe interruptibility in a multi-agent setting [14]. We also showed that this key property for safe RL needs trade-offs in the presence of unreliable perception [1]. We also proposed post-learning algorithms for problems such as algorithmic fairness [15].

Beside my regular research, I initiated a small working group around technical AI safety and ethics during my PhD. In this context, my colleagues and I became more and more aware that questions under "ethics" or "social implications", however non-technical they might sound, could in fact entail challenging technical problems. This has led me, for example, to explore how ideas in computational complexity could be used in social sciences, especially given that the latter are increasingly leveraging computing, not only as a practical toolbox (software, libraries, etc.), but also as a conceptual toolbox. To illustrate our theoretical argument, we reviewed practical cases from the work of data-scientists in the platform Kaggle. This project [4] was in collaboration with Professor Dominique Boullier from EPFL Digital Humanities lab and Sciences Po Paris. Another fruitful outcome of this working group is a book [22] with game theorist Dr Lê Nguyên Hoang, where we review the pressing questions in AI safety, ethics and social consequences. We show how they translate into technical challenges (such as the ones addressed in my PhD on robustness, or in Dr Hoang's PhD thesis on game theory and social choice) and propose a roadmap for the value alignment problem. (The value alignment problem is a fundamental multi-disciplinary question where the challenge is to align the objective function being optimized with what human values could be).

Future Research

Whilst a lot of work has been devoted to devising *efficient* distributed training schemes, time has come to also focus on their *robustness*, including their resilience to the most severe failures, namely *Byzantine* ones.

On the one hand, machine learning has now moved from domains such as music or gaming to critical applications such as flight control and autonomous driving where robustness is crucial. On the other hand, machine learning solutions are now distributed over hundreds of nodes. Poisoning attacks against some nodes are also now threatening the reliability of the primary source of information for most of the world's population: the Internet.

Bridging the Gap between Distributed Computing and Robust Statistics. For more than five decades, the field of statistics has been concerned with the question of inference from partially reliable data. Since the mid 1980s, the results of optimal breakdown were proven for the geometric median and the minimum volume ellipsoid. Coming from mathematics, it is understandable that many of these results did not have computational practicality in mind. Even when they had, the complexity was optimized for the number of points (n), not on the dimensionality of these points/vectors (d). We find this pattern even in important papers up to the early 2010s. *Big data* most often meant big n, not d.

This focus on n and not d is reasonable given the situation until the current data deluge. Even in the field of my hosting laboratory, distributed computing, the closest problem to my thesis is *multi-dimensional (approximate) agreement*. The single dimensional version of this problem was solved in the 1980s. Only recently, it was proven by Mendes and Herlihy (STOC 2013) that the d-dimensional version requires a n^d local computation and at least $\Omega(f \cdot d)$ correct processes in the presence of f Byzantine processes, with proofs of optimality. These results are reasonable when $d = 2$ or $d = 3$, like for n robots who need to agree on a position in a bi-dimensional plane or drones meeting in a tri-dimensional space. In modern machine learning, the dimensionality of the problems can reach values as high as $d = 10^9$ and calls for algorithms that are at most linear in d.

Almost at the same time as my thesis, the last four years brought several important results in high dimensional computational robust statistics. These results are not, however the end of the story. While some are linear in d for computation time, they can require an unreasonable d^2 space complexity or vice-versa. Others are linear or even sub-linear on every aspect, but make strong assumptions on data (e.g. Gaussian distribution) and are not

necessarily optimal for the worst-case mindset of Byzantine fault tolerance. With my colleagues, we have empirical evidence [5, 7] that the overheads of our Byzantine resilient solutions are on par with usual robust statistical tools (while also proven linear in time and space). Most importantly, the majority, if not all of the robust statistics toolbox is made for *synchronous* aggregation. Our approach is so far still unique in dealing with *asynchronous* Byzantine resilience. (Beside the worst-case analysis in [6], I am also working on non-malicious asynchronous schedule, viewed as a mild adversary [10]).

In all cases, I expect the interface between high dimensional robust statistics and Byzantine resilience to be a very fruitful topic of investigation for at least the upcoming half-decade.

Unreliable Servers and Decentralization. Besides the parameter-server setting used in most of the work on robust distributed learning, the issue of privacy is calling for peer-to-peer solutions where each peer (modelling a user) keeps their data locally and collaborates with others to boost their personalized learning. Yet, keeping data locally is not sufficient to protect against curious peers: additional measures are necessary, e.g., adding noise to ensure differential privacy or using homomorphic encryption. Nevertheless, combining these with robustness mechanisms against other misbehavior (such as poisoning) is not trivial. For instance, a privacy mechanism can easily leave a malicious actor unnoticed, since its proposed value has been obfuscated! Combining privacy and Byzantine resilience is a another unexplored question that I plan to investigate.

A first challenge, before adding privacy constraints, is to first study the Byzantine resilience of decentralized settings where no reliable single server is assumed. On this front, I spent the last period of my PhD supervising more junior PhD students develop solutions to deal with not only Byzantine workers, but also Byzantine servers [13, 12].

Robust Learning Machines. Another direction where I would like to push further what has been started in my thesis is the robustness of complex systems when viewed as distributed systems. In that *fine grained* view, I have initial results both on (artificial) neural and on metabolic (biological) networks, but they are both only initial steps requiring further investigation. It is worth noting that connectionism (neural networks in machine learning) started with the name *parallel and distributed processing* (the celebrated PDP book). Algorithms that have a priori nothing to do with fault tolerance in distributed systems, such as *Dropout* were in fact invented while optimizing for fault tolerance (Kerlirzin and Vallet 1993). As we argue in [16], complex systems are sometimes better understood by studying the failure of some of their components. This approach has been somehow successful in systems biology (gene knock-out experiments). Before my PhD, I worked in condensed matter physics and used the same approach to investigate solid state interfaces [20]. Understanding a system by studying its robustness to the failure of its components might help solve the mysteries of artificial neural networks.

Notes: except for [20], [21] and [22], authors order follows the theoretical computer science convention and is alphabetical.

I was the lead author in [2, 3, 6, 9, 10, 11, 18, 19, 20] and co-lead in [4, 5, 21].

References

[1] H. Aslund, E. M. El Mhamdi, R. Guerraoui, and A. Maurer. Virtuously safe reinforcement learning. *arXiv preprint arXiv:1805.11447*, 2018.

[2] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer. Brief announcement: Byzantine-tolerant machine learning. In *Proceedings of the ACM Symposium on Principles of Distributed Computing (**PODC**)*, 2017.

[3] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Advances in Neural Information Processing Systems (**NeurIPS**)*. 2017.

[4] D. Boullier and E. M. El Mhamdi. Machine learning and social sciences in the face of computational complexity. ***Revue d'Anthropologie des Connaissances**, to appear*, 2020.

[5] G. Damaskinos, E. M. El Mhamdi, R. Guerraoui, A. Guirguis, and S. Rouault. Aggregathor: Byzantine machine learning via robust gradient aggregation. In *the Conference on Machine Learning and Systems (**SysML / MLsys**)*, 2019.

[6] G. Damaskinos, E. M. El Mhamdi, R. Guerraoui, R. Patra, and M. Taziki. Asynchronous byzantine machine learning (the case of sgd). In *International Conference on Machine Learning (**ICML**)*. 2018.

[7] E. M. El Mhamd, R. Guerraoui, A. Guirguis, and S. Rouault. Garfield: System support for robust machine learning. *under submission*, 2019.

[8] E. M. El Mhamdi. *Robust Distributed Learning*. PhD thesis, École Polytechnique Fédérale de Lausanne, 2019.

[9] E. M. El Mhamdi and R. Guerraoui. When neurons fail. In *IEEE International Parallel and Distributed Processing Symposium (**IPDPS**)*. 2017.

[10] E. M. El Mhamdi and R. Guerraoui. Asynchronous learning in the optimal window of staleness. *under submission*, 2019.

[11] E. M. El Mhamdi and R. Guerraoui. Fast and secure distributed learning in high dimension. *arXiv preprint arXiv:1905.04374*, 2019.

[12] E. M. El Mhamdi, R. Guerraoui, and A. Guirguis. Fast Byzantine machine learning with unreliable servers. *arXiv preprint arXiv:1911.07537*, 2019.

[13] E. M. El Mhamdi, R. Guerraoui, A. Guirguis, and S. Rouault. Sgd: Decentralized Byzantine resilience. *arXiv preprint arXiv:1905.03853*, 2019.

[14] E. M. El Mhamdi, R. Guerraoui, H. Hendrikx, and A. Maurer. Dynamic safe interruptibility for decentralized multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems, (**NeurIPS**)*. 2017.

[15] E. M. El Mhamdi, R. Guerraoui, L. N. Hoang, and A. Maurer. Removing algorithmic discrimination (with minimal individual error). *arXiv preprint arXiv:1806.02510*, 2018.

[16] E. M. El Mhamdi, R. Guerraoui, A. Kucharavy, and S. Volodin. The probabilistic fault tolerance of neural networks in the continuous limit. *arXiv preprint arXiv:1902.01686*, 2019.

[17] E. M. El Mhamdi, R. Guerraoui, A. Maurer, and V. Tempez. Exploring the borderlands of the gathering problem. *Bulletin of the European Association of Theoretical Computer Science, (**Bulletin of EATCS**)*, 1(129), 2019.

[18] E. M. El Mhamdi, R. Guerraoui, and S. Rouault. On the robustness of a neural network. In *IEEE Symposium on Reliable Distributed Systems (**SRDS**)*. 2017.

[19] E. M. El Mhamdi, R. Guerraoui, and S. Rouault. The hidden vulnerability of distributed learning in Byzantium. In *International Conference on Machine Learning (**ICML**)*. 2018.

[20] E. M. El Mhamdi, J. Holovsky, B. Demaurex, C. Ballif, and S. De Wolf. Is light-induced degradation of a-si: H/c-si interfaces reversible? ***Applied Physics Letters***, 104(25):252108, 2014.

[21] E. M. El Mhamdi, A. Kucharavy, R. Guerraoui, and R. Li. Predicting complex genetic phenotypes using error propagation in weighted networks. *bioRxiv, 487348 (under review for a biology journal)*, 2018.

[22] L. N. Hoang and E. M. El Mhamdi. *The Fabulous Endeavor: Making Artificial Intelligence Robustly Beneficial*. **EDP Sciences**, published in Nov. 2019 in French under *Le fabuleux chantier: rendre l'intelligence artificielle robustement bénéfique*, English version under revision by the editor, 2020.