Towards Robust and Bias-free Federated Learning

Ousmane Touat, Sara Bouchenak

INSA Lyon – LIRIS
Lyon, France
ousmane.touat@insa-lyon.fr, sara.bouchenak@insa-lyon.fr

Résumé
Federated learning (FL) is an exciting machine learning approach where multiple devices collaboratively train a model without sharing their raw data. The FL system is vulnerable to the action of Byzantine clients sending arbitrary model updates, and the trained model may exhibit prediction bias towards specific groups. However, FL mechanisms tackling robustness and bias mitigation have contradicting objectives, motivating the question of building a FL system that comprehensively combines both objectives. In this paper, we first survey state-of-the-art approaches to robustness to Byzantine behavior and bias mitigation and analyze their respective objectives. Then, we conduct an empirical evaluation to illustrate the interplay between state-of-the-art FL robustness mechanisms and FL bias mitigation mechanisms. Specifically, we show that classical robust FL methods may inadvertently filter out benign FL clients that have statistically rare data, particularly for minority groups. Finally, we derive research directions for building more robust and bias-free FL systems.

Mots-clés : Federated Learning, Bias Mitigation, Robustness, Byzantine behavior

1. Introduction

Federated Learning (FL) is a distributed learning paradigm that allows multiple entities, denoted FL clients, to collaborate to learn a model with the help of a central node called the server, with privacy in mind, as the entities do not share their training data [9]. The deployment of FL in high-risk sectors such as healthcare has sparked concerns as the models may exhibit bias due to underrepresented minorities in data, leading to poorer performance for specific groups [4]. Furthermore, some FL clients denoted Byzantine client, can significantly impact the FL model training by sending arbitrary updates to the server [11]. Addressing model bias and Byzantine clients in FL systems is critical, and several existing works are addressing such issues separately. For instance, FL bias mitigation methods formulate an optimization approach with bias constraints, requiring FL clients to send additional statistics on their local data [3, 16, 4]. On the other hand, robust FL systems usually rely on passive outlier detection mechanisms using robust aggregation or gradient clipping [11, 12, 18]. However, incorporating both goals into a single FL system is challenging, as the current objectives are contradictory. FL bias mitigation requires considering outliers clients for adding representativity in their local data, while robust FL systems may eliminate outlier clients, which can affect model bias for worse [15, 6]. This paper investigates the challenges of building a FL system that combines bias constraints and robustness guarantees. We empirically showcase where classical robust aggregation degrades model bias and hinders bias mitigation. We then discuss some research directions to build a FL system combining robustness and bias mitigation goals.

2. Background and System Model

In this section, we recall the principles of FL, demographic bias and the standard Byzantine attack model.

2.1. Federated Learning

In Federated Learning (FL), we have n clients aiming to optimize a shared objective function collectively managed by a central server. The training process progresses through communication rounds, in which the central server sends the current global ML model with parameters θ to the FL clients. In this setup, the i-th client has its local data denoted as $\mathcal{D}_i$ and trains a local model denoted as θ_i by minimizing its local loss function $l_i(\theta_i, \mathcal{D}_i)$. We denote the union of clients' local data $\mathcal{D}$. At the end of the round, the clients send their updated model to the server, which update the new global model by aggregating the clients' updates. The classical approach for FL aggregation is called FedAvg [9] and consists in averaging the clients' model parameters as follows :

$$\theta = \sum_{k=1}^{k=N} \frac{|\mathcal{D}_k|}{|\mathcal{D}|}.\theta_k \tag{1}$$

2.2. Bias in Federated Learning

Bias refers to the property of an classifier model to disparately treat different groups based on sensitive attributes, denoted S. A sensitive attribute is binary as it partition the dataset into two groups. One widely used metric for measuring bias with respect to sensitive attributes is statistical parity difference (SPD). With y the prediction value from the FL model and S the value of the sensitive attribute we define SPD_S as follows :

$$SPD_S = |Pr(y = 1|S = 1) - Pr(y = 1|S = 0)| \tag{2}$$

In this formulation, a value of zero for SPD_S indicates that the algorithm is statistically fair concerning the sensitive attribute, as it predicts positive outcomes with the same probability for both groups.

2.3. Byzantine Attack Model in Federated Learning

The presence of malicious parties, known as Byzantine clients, denoted $\mathcal{B}$, can threaten the integrity of the FL process. The fraction of Byzantine clients is given by ρ, and the number of Byzantine clients denoted by q such that $q \leq \rho n$. Byzantine clients can deviate from the standard protocol and send arbitrary updates to the server, and they may even collude and have knowledge of the states of all other clients. The remaining $n - q$ clients who follow the protocol, denoted as honest clients, are represented by set $\mathcal{H}$. FL systems such as FedAvg are directly vulnerable to Byzantine attacks [11]. For example , we take a FL system of n client with among them 1 Byzantine client. We suppose that each honest client sends an update denoted as θ_i for $i \in \{1, \dots, n-1\}$. The Byzantine client can craft an update to make the server compute any wanted global model. We denote the wanted global model by the Byzantine client θ_{wt}, the Byzantine client can send the update $\theta_{byz} = |\mathcal{D}|.\theta_{wt} - \frac{1}{|\mathcal{D}|}(\mathcal{D}_k.\theta_i + \cdots + \mathcal{D}_{n-1}.\theta_{n-1})$. The server, unaware of the presence of the Byzantine client, will take all updates and aggregate them, producing the following global model as the Byzantine model wanted : $\theta = \frac{1}{|\mathcal{D}|}(\mathcal{D}_k.\theta_i + \cdots + \mathcal{D}_{n-1}.\theta_{n-1} + \theta_{byz}) = \theta_{wt}$. Such unwanted behavior harms the FL system greatly, making it unable to converge reliably, motivating the research on robust FL methods.

3. Related Work

3.1. Robustness to Byzantine behavior in Federated Learning

The Byzantine fault tolerance problem is well-known in distributed systems, where malicious nodes may provide incorrect information, leading to incorrect results. In FL, this problem can arise when some participating clients behave maliciously and manipulate the aggregated model parameters. Several Byzantine resilient FL methods have been proposed in the literature using the statistical properties of the client updates. Krum [11], and by extension Multi-Krum, order the clients' updates using a score stating how close they are to its $n - f - 2$ nearest neighbors using Euclidean distances. RFA [12] computes an approximate value of the geometric median of all client updates using Weiszfeld's algorithm. TrimmedMeans [18] computes the global model by taking the mean value of each coordinate of the client updates, truncating extreme values. However, such robust aggregation techniques fail when used on non-IID data, cannot withstand attacks, and degrade the global model performance [6]. For non-IID data, some defense mechanisms have been proposed to mitigate the impact of Byzantine behavior. ARAgg [6] randomly partitions client updates into "buckets" and aggregates them before using classical robust aggregation techniques. FedInv [20] performs model inversion on the clients' updates to generate dummy datasets and compare clients' Wasserstein distance on the dummy datasets distribution.

3.2. Bias Mitigation in Federated Learning

The issue of model bias is well-known in machine learning [10] and has also received prime attention in FL. In FL, we distinguish between approaches that mitigate model performance discrepancy between clients [8, 14] and ones that address model bias between demographic groups in FL, from where our work is positioned. We encounter two approaches to tackle this topic, namely client-side techniques and server-side techniques. The client-side technique aims to apply a local debiasing mechanism, such as data reweighting [1]. However, such a technique may fail on the global bias objective, especially where the FL setup is highly non-IID [4]. In server-side techniques, some FL system seeks to solve a global bias-constrained objective, requiring local clients to share information about the local statistic of sensitive attributes [16, 19, 4, 3]. AgnosticFair [16] uses local information from the client to optimize the data reweighting coefficient sent back to the client to seek its globally constrained objective problem. FairFL [19] train a client-selection policy function using multi-agent reinforcement learning to maximize a gain function based on how bias mitigation is performed on the global model. FairFed [4] linearly scalarize the inputted local model parameters based on their local bias discrepancy to the global bias metric. FCFL [3] also linearly scalarize the gradient updates from the clients to optimize both a bias constraint and consistency in the client's model performance.

3.3. Federated Learning with Bias and Robustness Guarantees

While the issues of demographic bias and robustness have been extensively explored in the literature, combining these two notions within a single system needs to be further examined. Ditto [8] confronts the notions of robustness against client-level fairness, showing that these two notions are competing, and implements a FL system that combines these properties using client model personalization. Recently, Singh and al. [13] built a robustness mechanism that minimizes the amount of honest "minority" clients detected as malicious ones. Assuming each client represents only one group, the server employs a decentralized micro aggregation algorithm to cluster the clients based on their published sensitive attribute, identifying malicious clients having updates too far from those cluster centroids.

4. Problem Illustration

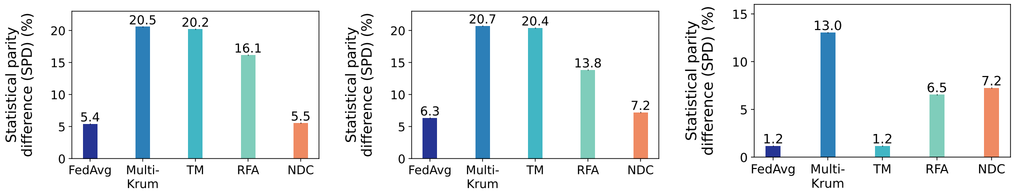
While the robustness of FL systems is essential, using robust aggregators in FL may impair model bias, particularly in non-IID FL settings. The robustness methods may eliminate outlier clients with different data distribution, suppressing their contribution to the global model. In this section, we will further illustrate this problem and explore its implications for model bias in FL.

4.1. Impact of FL Robustness Methods on Bias

We compared the behavior of 4 robust aggregators against a simple FL baseline on two tabular datasets, MEPS [2] and Adult [7], in a non-IID FL setting. On MEPS, an hospital expenditure dataset, the target attribute for the binary classification task is the utilization of medical facilities with the sensitive attributes of race and gender. The MEPS dataset is cut into a 4-client FL setup, showing the opposite trend of the three other clients, with the minority race becoming the majority. In the Adult dataset, the target attribute is income, and the considered sensitive attributes are gender and age. We use a 10-client FL setup generated with a Dirichlet function [5] for this dataset. We report the model bias for the baseline FedAvg method and the robust aggregators in figure 1. We observe that the robust aggregators, including Multi-Krum, RFA, and NDC, increase the model bias compared to the FL baseline on both MEPS (Figure 1(c)) and Adult (Figure 1(a),1(b)). The robust aggregators' behavior in such experiments is concerning as it demonstrates a degradation in the model bias in a setup without any Byzantine attacks. We attribute such behavior to the fact that robust aggregators tend to exclude clients with different data distributions. In our case, it corresponds to clients showing opposite bias dispositions, which are necessary to offset the biases from the other FL clients, thus showing the robust aggregators' difficulty working in a heterogeneous context when considering demographic bias.

4.2. Impact of FL Robustness Methods on FL Bias Mitigation

We study the behavior of two robust aggregators, Multi-Krum and NDC when used with a state-of-the-art bias mitigation method called FCFL [3]. We add the robust method as the first step after getting clients' updates before sending the updates to the FCFL server solver to perform the optimization task. Using NDC with FCFL means the clients' updates are clipped if their norm exceeds a fixed threshold and



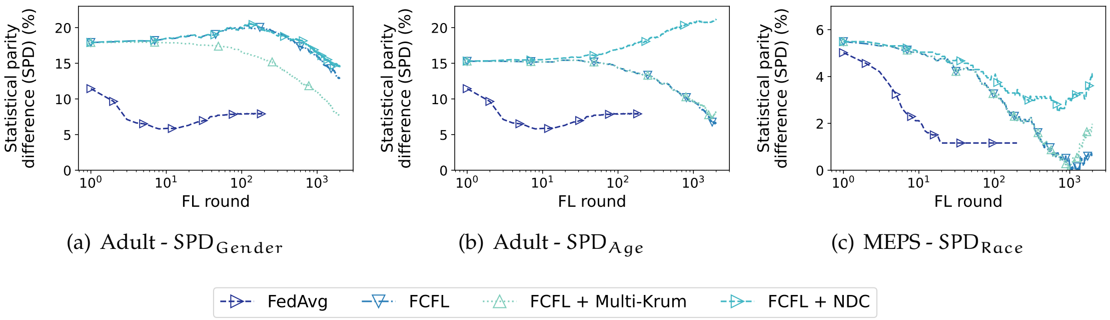
(a) Adult - $\text{SPD}_{\text{Gender}}$

(b) Adult - SPD_{Age}

(c) MEPS - SPD_{Race}

FIGURE 1 – Impact of FL robustness methods on Bias

sent to the server solver while for Multi-Krum it means that clients are selected to be sent to the server solver based on the set of clients with the lowest Krum score. The solver will then work with fewer updates to compute the global model. For all experiments, we used the same hyper-parameters on FCFL, specifically the SPD threshold and the learning rate. As explained earlier, we perform such experiments on the same datasets and sensitive attributes. We report the model bias of both added robust mechanisms against using FCFL alone and the baseline FL method in figure 2. We observe how the robust mechanism influences the behavior of FCFL. Specifically, on both MEPS (Figure 2(c)) and Adult (Figure 2(a),2(b)), we observed that NDC influenced FCFL, getting a higher value of SPD. For Multi-Krum, we can see its influence on FCFL the most on Adult (Figure 2(a)) and MEPS (Figure 2(c)).



(a) Adult - $\text{SPD}_{\text{Gender}}$

(b) Adult - SPD_{Age}

(c) MEPS - SPD_{Race}

FIGURE 2 – Interaction between FL bias mitigation and FL robustness mechanisms

5. Problem Formulation

A FL system must be robust to Byzantine behavior and be able to mitigate model bias, all of which are described as follows. We consider n FL clients, assuming a fraction $\rho \leq 0.5$ of Byzantine clients, and we denote $\mathcal{H}$ the set of honest client. Each client's data is represented as D_i with S the sensitive attribute, and the local model denoted as θ_i.

FL Robustness to Byzantine Behavior. The FL system must be robust against the attack of $f = \rho n$ Byzantine client, which is achieved when for any $\epsilon > 0$, the server can output a global model θ such that we can bound the diameter of the gradient of the loss function $f(\theta)$ by a constant ϵ. Ideally, the system can distinguish the honest FL clients, thus selecting only the latter for aggregation. We call such aggregation $\overline{\theta} = \frac{1}{|\mathcal{H}|}\sum_{j\in\mathcal{H}} \theta_j$ the true average. A robust aggregator must output a model θ close to the true average. Assuming that we can bound the honest client model heterogeneity $\|\theta_j - \theta_i\|^2 < \delta^2$ for all $i, j \in \mathcal{H}$, the output $\hat{\theta}$ from the robust Byzantine aggregator must respect the following constraint [6] :

$$\mathbb{E}\|\hat{\theta} - \overline{\theta}\|^2 < k\rho\delta^2 \tag{3}$$

FL Bias Mitigation. The goal is to learn the optimal value of the global model parameters θ that solve the bias-constrained optimization on the SPD_S value, where ϵ is the SPD_S constraint and $f_k(\theta)$ is the loss of the given model θ over local data of the honest client k :

$$\min_{\theta} f(\theta) = \min_{\theta} \frac{1}{|\mathcal{H}|} \sum_{k \in \mathcal{H}} f_k(\theta) \tag{4a}$$

$$\text{s.t } |SPD_S(\theta)| \leq \epsilon \tag{4b}$$

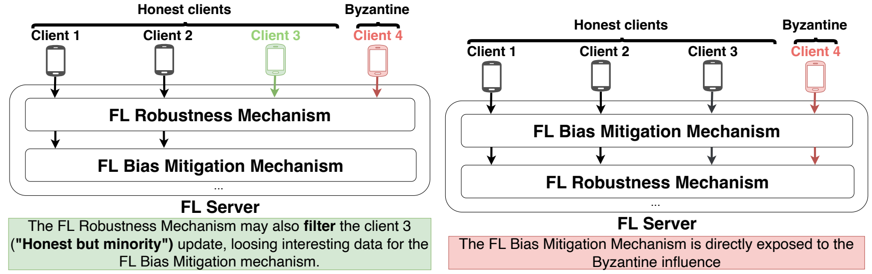
6. Why Handling Bias and Byzantine Behavior in FL is Difficult

This section shows the challenges of building a FL system combining robustness against Byzantine behavior, and model bias mitigation. We cover the following scenarios : Using classical robust aggregators first and then the FL Bias mitigation process separately (in section 6.1). Using the FL Bias mitigation process first and then the robustness mechanism separately (in section 6.2) and finally combine both robustness and bias mitigation objectives at once (in section 6.3). We provide then interesting research direction to address the cited challenges.

6.1. Why Applying a Classical FL Robustness Mechanism Followed by Classical FL Bias Mitigation Does Not Work

Suppose we first apply classical robust aggregation techniques and subsequently perform bias mitigation. The resulting system may fail to operate effectively in a non-IID environment. With a FL system consisting of n clients, with one honest client having a very different data distribution with better representation of the minority group, its contribution to the FL system would then help bring down the global model bias. However, we must expect our FL system to be robust to Byzantine clients by assuming the latter's presence within our system. Let us also suppose the presence of one Byzantine clients in our system. Using classical robust aggregators such as Multi-Krum would mean that one client get eliminated based on how far the model parameters are from all other clients. In section 4 , we illustrated how robust aggregators could eliminate the client with different data representativity. Unfortunately, this can lead to the exclusion of honest clients that offer demographic minority representation (see figure 3), losing crucial data representation that could be used by bias mitigation methods to help reduce the bias between demographic groups.

Observation 1 :
Using classical robust aggregators may eliminate honest client, affecting the normal behavior of FL bias mitigation.



The FL Robustness Mechanism may also **filter** the client 3 (**"Honest but minority")** update, loosing interesting data for the FL Bias Mitigation mechanism.

The FL Bias Mitigation Mechanism is directly exposed to the Byzantine influence

(a) Applying classical robust aggregators hinder FL bias mitigation

(b) Applying bias mitigation first in Byzantine setting does not work

FIGURE 3 – Applying classical robust aggregators hinder FL bias mitigation

6.2. Why Applying Classical FL Bias Mitigation Followed By a Classical FL Robustness Mechanism Does Not Work

In a FL setup where we suppose the presence of Byzantine clients, we must keep the Byzantine client from being considered by the FL Bias mitigation method as we need to trust their behavior. Suppose a FL system with honest clients and Byzantine clients. At the end of the FL round, each honest client sends regular updates, while the Byzantine sends arbitrary information. The classical bias mitigation method happening server-side would then get exposed to the Byzantine clients, which can directly harm the model utility (see figure 3(b)). A tradeoff with model utility is often expected with the bias mitigation methods [16, 3, 4]. Adding Byzantine clients would further increase the tradeoff, making some bias-constrained optimization tasks like AgnosticFair [16] more difficult. Furthermore, as mentioned in section 3.2 , some bias mitigation methods that intervene server-side, such as FairFed [4], require clients to send information on their local bias metrics in order to compute the global model. Byzantines can exploit this mechanism by sending falsified local bias information, which can cause the system to consider their harmful model updates even better.

Observation 2 :
Using the classical FL bias mitigation method before any robustness mechanism expose the bias mitigation method to the influence of the Byzantine clients.

6.3. On the Impossibility to Handle Bias and Byzantine Behavior at Once in Classical Approaches

Each client sends their locally trained model update to the server during the FL process for optimization. In our case, we assume wanting to maximize the global model utility on a data distribution representing the union of honest client data under bias constraint. At the same time, we want our model to have robustness guarantees against the current Byzantine activity in the system. In order to solve the bias-constrained objective problem on the global data distribution, we must optimize our objective based on the set of models updates coming from the honest clients set, and nothing else as we do not trust the updates coming from the Byzantine clients as mentioned in section 6.2 . However, such a set of honest client updates is heterogeneous due to outlier clients, which we assume represents additional information on minorities compared to the one found in most updates. Karimireddy and al. argue that we can construct a set of model updates including all Byzantines that would pass the robust aggregation part [6]. Our system may consider those Byzantine updates for the server-side optimization part, which brings an error component to the global optimization process that we cannot ignore. The output of such a system combining bias mitigation and robustness is not guaranteed to keep the bias constraint under the global data distribution.

Observation 3 :
Using both objectives is impossible in heterogeneous settings using only model updates as information, as the sever cannot distinguish between outlier clients and Byzantine ones.

6.4. Discussion and Research Directions

Despite the challenges of combining bias mitigation and robustness to Byzantine behavior methods in FL, interesting research directions remain to explore. We have seen the necessity of providing robustness before any other bias mitigation operation at the server level in section 6.2 . This requires a new method for detecting Byzantine clients in non-IID environments. First, such a FL system must guarantee the selection of outlier clients when there are no Byzantine attacks on the system. Also, the FL system must reduce the amount of falsely predicted Byzantine clients, as these clients probably contain critical data representativity. The server can try spotting such "honest and minority" clients among the deemed Byzantine clients by the robust aggregators. To distinguish such clients from the Byzantine ones, we could rely on reputation mechanisms to precisely identify Byzantine clients [17]. Furthermore, the server could also ask the clients to send additional information on local data distribution and privately compare such information with their sent model updates. The system could then analyze inconsistencies between the information on the data distribution and what can be extracted from the client model update (i.e., using model inversion [20]), revealing the Byzantine nature of the client.

7. Conclusion

FL model bias and robustness against Byzantines have been extensively studied over the last few years. Addressing both problems within a single FL system, while critically needed, has been elusive. This study investigated the challenges of constructing a FL system with both robustness to Byzantine behavior properties and bias guarantees. By design, a robust FL system degrades inclusiveness, particularly for outlier model behavior, which may have the side effect of setting aside data representing minorities, which is necessary for proper functioning FL bias mitigation techniques. Furthermore, we discuss the limitations of combining classical bias mitigation methods with existing robustness mechanisms. Finally we formulate possible research directions for building robust, bias-free FL.

Bibliographie

1. Abay (A.), Zhou (Y.), Baracaldo (N.), Rajamoni (S.), Chuba (E.) et Ludwig (H.). – Mitigating Bias in Federated Learning, 2020.
2. Cohen (S.). – Design Strategies and Innovations in the Medical Expenditure Panel Survey. *Medical care*, vol. 41, 08 2003, pp. III5–III12.
3. Cui (S.), Pan (W.), Liang (J.), Zhang (C.) et Wang (F.). – Addressing Algorithmic Disparity and Performance Inconsistency in Federated Learning. *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 26091–26102.
4. Ezzeldin (Y. H.), Yan (S.), He (C.), Ferrara (E.) et Avestimehr (S.). – FairFed : Enabling Group Fairness in Federated Learning, 2021.
5. Hsu (T.-M. H.), Qi (H.) et Brown (M.). – Measuring the Effects of Non-Identical Data Distribution for Federated Visual Classification, 2019.
6. Karimireddy (S. P.), He (L.) et Jaggi (M.). – Byzantine-robust learning on heterogeneous datasets via bucketing, 2020.
7. Kohavi (R.). – Scaling Up the Accuracy of Naive-Bayes Classifiers : A Decision-Tree Hybrid. – In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, KDD'96*, KDD'96, p. 202–207, Portland, Oregon, USA, 1996. AAAI Press.
8. Li (T.), Hu (S.), Beirami (A.) et Smith (V.). – Ditto : Fair and robust federated learning through personalization, 2020.
9. McMahan (B.), Moore (E.), Ramage (D.), Hampson (S.) et Arcas (B. A. Y.). – Communication-Efficient Learning of Deep Networks from Decentralized Data. – In Singh (A.) et Zhu (J.) (édité par), *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, *Proceedings of Machine Learning Research*, volume 54, pp. 1273–1282, Fort Lauderdale Florida, USA, 20–22 Apr 2017. PMLR.
10. Ninareh (M.), Fred (M.), Nripsuta (S.), Kristina (L.) et Aram (G.). – A Survey on Bias and Fairness in Machine Learning. *ACM Computing Surveys*, vol. 54, n6, 2022, pp. 115 :1–115 :35.
11. Peva (B.), El Mahdi (E. M.), Rachid (G.) et Julien (S.). – Machine learning with adversaries : Byzantine tolerant gradient descent. – In Isabelle (G.), Ulrike (v. L.), Samy (B.), Hanna M. (W.), Rob (F.), S. V. N. (V.) et Roman (G.) (édité par), *Advances in Neural Information Processing Systems 30 : Annual Conference on Neural Information Processing Systems 2017*, pp. 119–129, Long Beach, CA, USA, 2017. The MIT Press.
12. Pillutla (K.), Kakade (S. M.) et Harchaoui (Z.). – Robust aggregation for federated learning, 2019.
13. Singh (A.), Blanco-Justicia (A.) et Domingo-Ferrer (J.). – Fair detection of poisoning attacks in federated learning on non-i.i.d. data. *Data Mining and Knowledge Discovery*, vol. 1, 2023, pp. –.
14. Song (T.), Tong (Y.) et Wei (S.). – Profit Allocation for Federated Learning. – In *2019 IEEE International Conference on Big Data (Big Data)*, pp. 2577–2586, Los Angeles, CA, USA, 2019. IEEE.
15. Wang (H.), Sreenivasan (K.), Rajput (S.), Vishwakarma (H.), Agarwal (S.), Sohn (J.-y.), Lee (K.) et Papailiopoulos (D.). – Attack of the tails : Yes, you really can backdoor federated learning, 2020.
16. Wei (D.), Depeng (X.), Xintao (W.) et Hanghang (T.). – *Fairness-Aware Agnostic Federated Learning*, pp. 181–189. – Virtual Event, SIAM, 2021.
17. Xu (X.) et Lyu (L.). – A reputation mechanism is all you need : Collaborative fairness and adversarial robustness in federated learning, 2020.
18. Yin (D.), Chen (Y.), Kannan (R.) et Bartlett (P.). – Byzantine-robust distributed learning : Towards optimal statistical rates. – In Dy (J.) et Krause (A.) (édité par), *Proceedings of the 35th International Conference on Machine Learning*, *Proceedings of Machine Learning Research*, volume 80, pp. 5650–5659, Stockholm, Sweden, 10–15 Jul 2018. PMLR.
19. Zhang (D. Y.), Ziyi (K.) et Dong (W.). – FairFL : A Fair Federated Learning Approach to Reducing Demographic Bias in Privacy-Sensitive Classification Models. – In *Proceedings of the 2020 IEEE International Conference on Big Data, BigData2020*, pp. 1051–1060, United States, 2020. IEEE.
20. Zhao (B.), Sun (P.), Wang (T.) et Jiang (K.). – fedinv : Byzantine-robust federated learning by inversing local model updates, 2022.