Metrics and Strategies for Adversarial Mitigation in Federated Learning-based Intrusion Detection

Léo Lavaur
IMT Atlantique, IRISA
leo.lavaur@imt-atlantique.fr

Yann Busnel
IMT Atlantique, IRISA
yann.busnel@imt-atlantique.fr

Pierre-Marie Lechevalier
IMT Atlantique, IRISA
pierre-marie.lechevalier@imt-atlantique.fr

Marc-Oliver Pahl
IMT Atlantique, IRISA
marc-oliver.pahl@imt-atlantique.fr

Fabien Autrel
IMT Atlantique, IRISA
fabien.autrel@imt-atlantique.fr

***Abstract*—Since its introduction in 2016, federated learning (FL) has been used in multiple domains, such as intrusion detection. However, FL literature shows that the heterogeneity of most real-world FL applications makes it difficult for clients to converge in a suitable global model. Furthermore, as a collaborative system, FL is vulnerable to attacks, such as model poisoning. While strategies have been identified in the literature, they often rely on the assumption that the data distribution among participants is homogeneous. In this paper, we review the current challenges in clustering and adversarial mitigation in heterogeneous FL, and propose different strategies to address them. Namely, we present a cross-evaluation framework for exhaustive gathering, and a set of algorithmic countermeasures based on principal component analysis. We show preliminary results of our clustering mechanism, which validates the effectiveness of the cross-evaluation framework.**

***Index Terms*—intrusion detection, federated learning, adversarial mitigation, model poisoning, clustering, trust, principal component analysis**

I. Introduction

Due to the increasing interconnection of computing environments, the number of cyberattacks have been increasing over the last years. Intrusion detection systems (IDSs) play a critical role in ensuring the security of information systems. In particular, network-based intrusion detection systems (NIDSs) process network traffic to detect malicious activities, such as malware propagation, data exfiltration, and denial of service (DoS) attacks. Traditional machine learning (ML) methods, while effective in many cases, can be limited by the lack of training data, and the difficulties in reusing existing models.

Previous studies have demonstrated that, despite the availability of technologies for data sharing, stakeholders are still reluctant to share sensitive information, even for ML purposes as training data. This is particularly true for IDSs, which are often used to detect attacks on critical systems.

Federated learning (FL) is a distributed ML technique that allows clients to train a global model without sharing their data [1]. In the context of IDS, FL can be used to collaboratively train models that can be used to detect attacks on information systems, while keeping the data of each participant private [2].

This research is part of the chair CyberCNI.fr with support of the FEDER development fund of the Brittany region.

In the context of FL, each participant contributes to the system with the model it produces. Depending on the use case, contributions can be highly heterogeneous, as they are directly related to the data distribution of each participant. In NIDSs, each client can have different network architectures, services, and protocols; therefore training their own model on different data distributions.

A. Problem statement

Due to the heterogeneity of the data distributions, FL algorithms have difficulties to converge to a global model that is suitable for all participants [3]. This is particularly true for FL algorithms that treat participants' contributions evenly, such as `FedAvg` [1]. Other algorithms, such as `FedProx` [4], can be used to mitigate the impact of heterogeneity, but also show limitations as they make assumptions on local data distribution [3]. Other approaches based on clustering have been proposed to mitigate the impact of heterogeneity, such as by Briggs *et al.* [5].

Furthermore, as a collaborative system, FL is vulnerable to attacks, such as model poisoning [6]. Fortunately, malicious contributions can often be detected by comparing them to the other contributions. However, similarity-based detection strategies falter at distinguishing a malicious contribution from a legitimate one that is simply different from the others, especially in the context of heterogeneous FL.

In order to make sound and reproducible experiments, we make the following working assumptions:

(i) *The server is honest but curious.* The server is not malicious, but it is interested in the data of the clients. To facilitate the aggregation of the models, we consider the server to be trusted to perform as expected in the aggregation process.

(ii) *Participants have incentives in sharing their data.* We assume that the participants are willing to share their data in order to benefit from the collaboration.

(iii) *Participants can be honest, negligent, or malicious.* Typical participants are honest, but some of them can train their model on data of poor quality without knowing it. Some of them can also be malicious, train their model

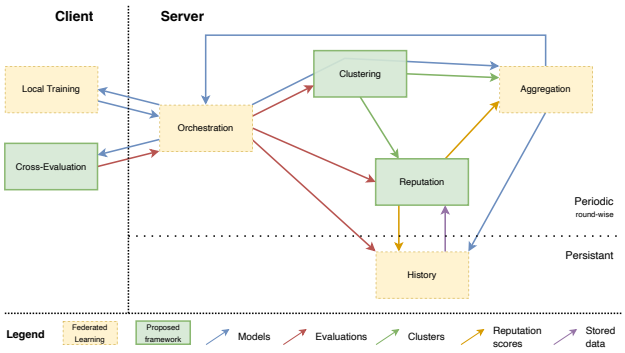

Fig. 1. Architecture of the cross-evaluation framework.

on malicious data, and voluntarily trying to degrade the performance of the global model.

(iv) *Participants are not faulty.* We assume that participants are always available for training, and thus provide a contribution at each round of FL. While this is a strong assumption, asynchronous communication and client fault tolerance are out of the scope of this work.

B. Contribution

In this paper, we present strategies to mitigate the impact of heterogeneity and malicious contributions in heterogeneous FL systems. Namely, we propose a novel cross-evaluation framework that provide metrics for assessing the similarity between contributions, and provide client clustering and adversarial detection. We show preliminary results of our clustering mechanism, which validates the effectiveness of our framework. Finally, we present a set of algorithmic countermeasures based on principal component analysis (PCA) to mitigate the impact of malicious contributions in large-scale FL systems, where cross-evaluation is not feasible.

II. Cross-evaluation framework

Both clustering and poisoning detection require defining metrics that measure the distance between FL contributions. Therefore, in this section, we present a cross-evaluation framework, which maps out a way to assess the similarity between client distributions. Figure 1 depicts the architecture of our framework, which is composed of three atomic components: cross-evaluation, clustering, and reputation system (RS).

A. Framework operation

At each round of standard FL, the server collects the model updates of all participants, and then disseminate them to all participants. Each participant then evaluates the contributions of the other participants, and provides a score for each of them, *e.g.* detection accuracy or fitting loss.

The resulting scores, so-called evaluation vectors $\vec{e_i}, i \in N$, are then concatenated in a cross-evaluation matrix M. M is a squared matrix, with $M_{i,j}$ being the evaluation of client i on the model provided by j, using i's data. When looking at the rows, we see the point of view of each client. We assume that clients with similar data distributions will produce similar evaluation vectors, which is confirmed by our preliminary results (see Section II-B). Therefore, we use hierarchical clustering [5] to group clients together, starting with 1-client clusters. The resulting clusters are then used to aggregate model updates locally in each client, with more homogeneous data distribution.

To decide how to weight the contributions of each participant in each cluster, and therefore mitigate the impact of identified negative contributions, we propose to rely on reputation system (RS). RSs have the advantage of considering the evolution of a participant's behavior over time and across multiple interactions, while traditional methods only consider the current state of the system at a round r. The columns of M represent the evaluations received by one client's contribution from its peers, that we note $\vec{e_j}, j \in N$. The vectors $\vec{f_j}$ of are analogous to the notion of *feedback* in RSs [7]. Therefore, we can use the results of the cross-evaluation to compute reputation scores for each client, and then use them to weight client contributions in the aggregation.

B. Preliminary results

While this approach is still a work in progress, we have already conducted preliminary experiments to assess the relevance of the cross-evaluation approach. Unfortunately, it is difficult to review atomically the effectiveness of the cross-evaluation framework, as it is tightly coupled with the clustering and RS components. Therefore, we implement a state-of-the-art clustering algorithm [5] based on the evaluation vectors, *i.e.* $\vec{e_i}$.

We instantiate ten clients with the same model architecture—an auto-encoder, and distribute them across four different datasets [8]. All dataset share the same features, but differ in the number of samples and the number of classes. We then train the clients for 10 rounds, and collect the local models at each round. Using the approach described in Section II-A, we compute the cross-evaluation matrix M, and use hierarchical clustering to group clients in an unknown number of clusters.

Finally, we use the Rand index to assess the quality of the clustering. This metric measures the similarity between two distributions. In our case, we compare the clustering obtained by the algorithm with the ground truth, which is the dataset used by each client. The results in Figure 2 show that the clustering algorithm is able to accurately group clients with similar data distributions. However, we observe fluctuations in the Rand index, which are due to the fact that one of the cluster is still heterogeneous without local aggregation. The clustering is not currently integrated in the framework. We expect that aggregating specialized models per cluster will improve the performance of the clustering algorithm, as clients will have converged locally.

III. PCA-based countermeasures

While the first results of the cross-evaluation framework are promising, this approach will not scale. In fact, intrusion detection deployments can also be done on-device, such as in endpoint detection and response (EDR) solutions, where

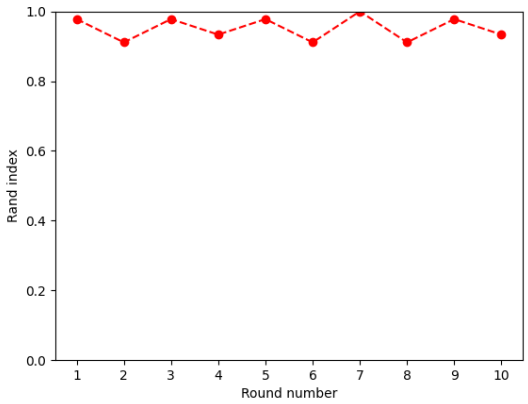

Fig. 2. Rand index for the clustering algorithm—round 1 to 10.

the number of clients is much larger. Therefore, we also consider the use of existing statistical methods to mitigate the impact of malicious contributions. In this section, we present strategies inspired by PCA, to cluster clients and optimize their contributions.

PCA is a dimensionality reduction technique that aims at finding a new frame of reference for the data, such that the scattering of the data is maximized. This projection can be used to reduce the dimensionality of the data, by removing scalars corresponding to the least significative eigenvalues. Here, we propose to use the rationale behind PCA, and applying it to a similarity matrix S. The latter is a squared and symmetric matrix, with $S_{i,j}$ being the similarity between the model updates of client i and j. The similarity between two models can be measured using different metrics, depending on the objective, such as cosine similarity. After diagonalizing S to find its eigenvalues, we envision two approaches to protect and optimize FL.

First, the eigenvalues can be used to weight the contributions of each participant. As S represents the similarity between the model updates of each participant, observing the j^{th} column of S is analogous to observing the system from the perspective of client j. Therefore, the j^{th} eigenvalue—if unsorted—represents the importance of client j's point of view to better describe the system. The intuition is that the more important a client's perspective is, the more relevant its contribution will be. Then, we can use the eigenvalues to weight the contributions of each participant in aggregation, thus optimizing model convergence and mitigating the impact of malicious contributions.

Second, we can use the eigenvectors to cluster clients. The eigenvectors of S represent the dilatation of the data along each axis of the new frame of reference. By using the sorted eigenvectors, and selecting the first k axes, we can project the data onto a k-dimensional space, like in traditional PCA. Here, clients will be positioned in the k-dimensional space according to their similarity with the chosen eigenvectors. The k most representative client (according to the k highest eigenvalues) can be considered as the centers of K clusters, and clients can be assigned to the closest cluster when their similarity is above a threshold that could be empirically determined. If clients remain unassigned, we can either consider them as outliers, or consider that there is not enough clusters to properly group clients. In the latter case, we can project the data onto a space of $k+1$ dimensions, iterate the process as necessary.

Implementation and evaluation of these approaches remain to be done. However, the rationale behind them make them promising alternative to the aforementioned strategies based on cross-evaluation, especially for larger-scale systems.

IV. Conclusion

This research paper presents a solution to the challenges faced in applying FL for intrusion detection in heterogeneous environments. After a comprehensive literature review [2], and with the proposal of a novel cross-evaluation framework, we propose methods to assess the similarity between client distributions, and mitigate the impact of heterogeneity and malicious contributions. The preliminary experimental results validate the effectiveness of our framework and contribute to the fields of FL and IDSs. Our solution enables the collaboration of robust models while preserving the privacy of individual participants, the quality of their contributions, and the relevance of the aggregated models.

While this work is based on rigorous positioning and encouraging preliminary results, it is still a work in progress. First, we need to implement the proposed architecture in existing FL frameworks, such as `Flower`, in order to evaluate the effectiveness of the entire stack of components. Second, we need to pursue a thorough evaluation of the proposed framework, in particular by comparing it to existing methods. Finally, we need to implement and evaluate the PCA-based countermeasures.

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, 2017.

[2] L. Lavaur, M.-O. Pahl, Y. Busnel, and F. Autrel, "The Evolution of Federated Learning-based Intrusion Detection and Mitigation: A Survey," *IEEE TNSM*, Special Issue on Network Security Management, 2022.

[3] P. Kairouz, H. B. McMahan, B. Avent, *et al.*, "Advances and Open Problems in Federated Learning," 8, 2021. arXiv: 1912.04977 [cs, stat].

[4] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated Optimization in Heterogeneous Networks," 21, 2020. arXiv: 1812.06127 [cs, stat].

[5] C. Briggs, Z. Fan, and P. Andras, "Federated learning with hierarchical clustering of local updates to improve training on non-IID data," in *2020 IJCNN*, 2020.

[6] Z. Tian, L. Cui, J. Liang, and S. Yu, "A Comprehensive Survey on Poisoning Attacks and Countermeasures in Machine Learning," *ACM Computing Surveys*, 18, 2022.

[7] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman, "Reputation systems," *Communications of the ACM*, 1, 2000.

[8] M. Sarhan, S. Layeghy, and M. Portmann, *Towards a Standard Feature Set for Network Intrusion Detection System Datasets*, 14, 2021. arXiv: 2101.11315 [cs].