COLLEGES BRETAGNE LOIRE — SCIENCES POUR L'INGENIEUR ET LE NUMERIQUE

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

THÈSE DE DOCTORAT DE

L'ÉCOLE NATIONALE SUPÉRIEURE
MINES-TÉLÉCOM ATLANTIQUE BRETAGNE
PAYS DE LA LOIRE – IMT ATLANTIQUE

ÉCOLE DOCTORALE Nº 648
Sciences pour l'Ingénieur et le Numérique
Spécialité : *Sciences et technologies de l'information et de la communication*

Par
Léo LAVAUR

Améliorer la détection d'intrusions dans les systèmes répartis grâce à l'apprentissage fédéré

Thèse présentée et soutenue à Rennes, le 7 octobre 2024
Unité de recherche : IRISA (UMR 6074), SOTERN

Rapporteurs avant soutenance :

Anne-Marie KERMARREC	Professeure à l'Université Polytechnique Fédérales de Lausanne (EPFL)
Éric TOTEL	Professeur à Télécom SudParis

Composition du Jury :

Attention, en cas d'absence d'un des membres du Jury le jour de la soutenance, la composition du jury doit être revue pour s'assurer quelle est conforme et devra être répercutée sur la couverture de thèse

Président :	*À compléter après la soutenance.*	
Examinateurs :	Sonia BEN MOKHTAR	Directrice de Recherche au CNRS
	Pierre-François GIMENEZ	Maître de Conférence à CentraleSupelec
	Vincent NICOMETTE	Professeur à INSA Toulouse
	Fabien AUTREL	Ingénieur de Recherche à IMT Atlantique
	Marc-Oliver PAHL	Directeur d'Études à IMT Atlantique
Dir. de thèse :	Yann BUSNEL	Professeur à IMT Nord Europe

Invité(s) :

Prénom NOM   Fonction et établissement d'exercice

Abstracts

Résumé

La collaboration entre les différents acteurs de la cybersécurité est essentielle pour lutter contre des attaques de plus en plus sophistiquées et nombreuses. Pourtant, les organisations sont souvent réticentes à partager leurs données, par peur de compromettre leur confidentialité et leur avantage concurrentiel, et ce même si cela pourrait d'améliorer leurs modèles de détection d'intrusions. L'apprentissage fédéré est un paradigme récent en apprentissage automatique qui permet à des clients répartis d'entraîner un modèle commun sans partager leurs données. Ces propriétés de collaboration et de confidentialité en font un candidat idéal pour des applications sensibles comme la détection d'intrusions. Si un certain nombre d'applications ont montré qu'il est, en effet, possible d'entraîner un modèle unique sur des données réparties de détection d'intrusions, peu se sont intéressées à l'aspect collaboratif de ce paradigme. En plus de l'aspect collaboratif, d'autres problématiques apparaissent dans ce contexte, telles que l'hétérogénéité des données des différents participants ou la gestion de participants non fiables. Dans ce manuscrit, nous explorons l'utilisation de l'apprentissage fédéré pour construire des systèmes collaboratifs de détection d'intrusions. En particulier, nous explorons (i) l'impact de la qualité des données dans des contextes hétérogènes, (ii) certains types d'attaques par empoisonnement, et (iii) proposons des outils et des méthodologies pour améliorer l'évaluation de ce type d'algorithmes répartis.

Abstract

Collaboration between different cybersecurity actors is essential to fight against increasingly sophisticated and numerous attacks. However, stakeholders are often reluctant to share their data, fearing confidentiality and privacy issues and the loss of their competitive advantage, although it would improve their intrusion detection models. Federated learning is a recent paradigm in machine learning that allows distributed clients to train a common model without sharing their data. These properties of collaboration and confidentiality make it an ideal candidate for sensitive applications such as intrusion detection. While several applications have shown that it is indeed possible to train a single model on distributed intrusion detection data, few have focused on the collaborative aspect of this paradigm. In addition to the collaborative aspect, other challenges arise in this context, such as the heterogeneity of the data between different participants or the management of untrusted contributions. In this manuscript, we explore the use of federated learning to build collaborative intrusion detection systems. In particular, we explore (i) the impact of data quality in heterogeneous contexts, (ii) some types of poisoning attacks, and (iii) propose tools and methodologies to improve the evaluation of these types of distributed algorithms.

ACKNOWLEDGEMENTS

Table of Contents

Part I

Federated Learning to build CIDSs

Part II

Quantifying the Limitations of FIDSs

Part III

Providing Solutions

Chapter 1

Model Quality Assessment for Reputation-aware Collaborative Federated Learning

Contents

1.1 Introduction

In the previous chapters, we identified and studied two major challenges that currently hinder the adoption and deployment of Federated Intrusion Detection Systems (FIDSs): (1) the heterogeneity of the data sources, notably in Cross-Silo Federated Learning (CS-FL) settings; and (2) the susceptibility of FIDSs to adversarial attacks. More generally, because collaborative systems are inherently sensitive to input quality, any form of Byzantine failure should be considered. While we focus specifically on data-related failures in the context of this thesis, Byzantine faults can also encompass other types of failures, such as crashes, arbitrary behavior, or communication issues. This applies whether the participants are honest but use faulty data, or actively malicious. In this heterogeneous context, it is particularly challenging to distinguish a faulty or malicious contribution from a legitimate one originating from a different type of infrastructure.

Approaches that assess model quality [PB23] or mitigate poisoning [Bla+17; Cao+22] in homogeneous distributions typically compare or evaluate a model using a single source of truth. Building such a single source of truth, however, is inadequate in heterogeneous contexts due to the differences between participants. Assuming that all contributions are therefore different, some approaches detect colluding attackers based on their similarity [ALL21; FYB20]. Nevertheless, these approaches fail to detect isolated attackers.

In this chapter, we present `RADAR`, an architecture for CS-FL guarantying high-quality model aggregation, regardless of the data homogeneity. `RADAR` relies on three main ingredients: *i)* a modified Federated Learning (FL) workflow, where each participant uses its local dataset to evaluate the other participants' models, between the training and aggregation steps; *ii)* a clustering algorithm leveraging the participants' perceived similarity to aggregate group-specific global models; and *iii)* a reputation system that weights the participants' contributions based on their past interactions.

We evaluate the performance of RADAR in a realistic Collaborative Intrusion Detection System (CIDS) use case, using four network flow datasets with standardized features, representing different environments, and model various Byzantine behavior using label-flipping. We also compare our approach to existing strategies [FYB20; McM+17], and conclude that RADAR can detect Byzantines contributions under most scenarios, from noisy labels to colluding poisoning attacks.

The content of this chapter is based on our work published in IEEE International Symposium on Reliable Distributed Systems (SRDS) [Léo+24], which results from a collaboration with Pierre-Marie Lechevalier, another Ph.D. student at IMT Atlantique. It is organized as follows. **??** introduces the reader to the problem of model quality assessment in CS-FL and the necessary background. Section 1.3 reviews the related work, before we dive in RADAR's architecture in Section 1.4. **????** present the experimental setup and results, and we discuss our findings in **??**. Finally, **??** concludes this chapter.

Contributions of this chapter

- RADAR, an architectural framework to protect FL strategies using clustering and reputation-aware aggregation, validated by extensive evaluation against relevant baselines;
- a demonstration that evaluation metrics (such as accuracy, F1-score, or loss) can be used to effectively assess similarity between FL participants, and as an input to clustering and reputation algorithms;
- the confirmation that combining reputation and clustering successfully addresses the problem of contribution quality assessment in heterogeneous settings.

1.2 Problem Statement

In continuity with **????**, we consider once more the use case introduced in **??** and the associated datasets. Specifically, we focus on a heterogeneous declination of this CIDS use case, where we admit that participants share similarities in their data distributions—*e.g.*, between organizations operating in the same sector or having similar network infrastructure. This setting, also mentioned in **??**, is referred to as *practical* Non Independent and Identically Distributed (NIID) [Hua+21]. We also set $C = 1$, as we consider that the participants are highly available and interested in collaborating.

1.2.1 Low-quality Contributions

In FL, the quality of the global model is directly impacted by the quality of the participants' contributions. In a Intrusion Detection System (IDS) context, the poor quality of a Machine Learning (ML) model can be induced by some choices in terms of architecture, hyperparameters, or optimizer—all fixed by the server, but also by the quality of the training data. Multiple factors can affect the quality of local training data [Jai+20], such as: (1) *Label noise*—samples associated with the wrong labels; (2) *Class imbalance*—differences in terms of class representation in the dataset; or (3) *Data heterogeneity*—the variations between samples of the same class.

Similar to existing works on data-quality [Den+21; Den+22], we focus on label noise, which can have significant consequences on the global model's performance, depending on the proportion of mislabeled samples. In a CIDS, label noise can unknowingly be introduced by the participants, either due to misconfigurations or to the presence of compromised devices. We consider two types of label noise: *missed intrusions* and *misclassification*.

a) *Missed intrusions* occur when a malicious sample is mislabeled as benign, leading to a false negative. Participants in CIDSs label the attacks they are aware of, but some might have been unnoticed.

b) A *misclassification* is the random mislabeling of a sample. This can be due to a lack of knowledge or to a misconfiguration.

Such participants are referred to as *honest-but-neglectful*. Because these errors are assumed to be unintentional, the proportion of *misclassified* samples is expected to be low. However, the concept of *missed intrusions* implies that the participants are not aware of an entire attack, which can represent a significant proportion of their dataset.

1.2.2 Data Poisoning Attacks

In addition to accidental low-quality contributions, some participants might deliberately upload model updates that would negatively impact the performance of the global model. Specifically, we consider the same attack model as detailed in **??**, and focus on label-flipping attacks. The model can be summarized as follows:

Attackers' Knowledge. Attackers are *gray-box* adversaries, meaning that they have access to the same information as the other participants; *e.g.*, the last global models, the hyperparameters, or the optimizer.

Attackers' Objective. With targeted poisoning, attackers aim at making a specific type of attack invisible to the Network-based Intrusion Detection System (NIDS). Conversely, with untargeted attacks, they seek to jeopardize the NIDS performance by maximizing the number of misclassifications.

Attackers' Capabilities. Attackers can flip the labels of an arbitrary proportion of their dataset, referred to as the Data Poisoning Rate (DPR) and denoted α. They can act alone or in collusion with other by applying the same strategy. The proportion of attackers in the system is described by the Model Poisoning Rate (MPR) and denoted β.

Because we do not make a priori assumptions on the whether the participants are malicious or not in this contribution, we also refer to the DPR as the *noisiness* of a participant. The MPR, on the other hand, almost exclusively describes attackers, as it is unlikely for the same Byzantine fault to occur in multiple participants simultaneously.

1.2.3 Problem Formalization

Based on the previous assumptions, we consider that participants might upload model updates that would negatively impact the performance of the global model, deliberately or not. Multiple forms of such actors can exist: external actors altering legitimate clients' data (*i.e. compromised*), clients whose local training sets are of poor quality (*i.e. honest-but-neglectful*), or clients modifying their own local data on purpose (*i.e. malicious*). We refer to them as *Byzantine participants* or simply *Byzantines* in the remaining of this paper.

We further consider that the server can be trusted to perform the aggregation faithfully, and that FL guaranties the confidentiality of the local datasets. Attacking the server is out of the scope of this contribution. Consequently, we aim at weighting or discarding the participants' contributions based on their quality to guaranty the performance of the aggregated model.

Problem 1.1: Quality Assessment in Heterogeneous Settings

For n participants p_k and their local datasets d_k of unknown similarity, each participant uploads a model update w_k^r at each round r. Given $P = \{p_1, p_2, \ldots, p_n\}$ and $W = \{w_1^r, w_2^r, \ldots, w_n^r\}$, how can one assess the quality of each participant's contribution w_k^r without making assumptions on the data distribution across the datasets d_k?

1.3 Background and Related Work

The reliability of a submitted local model can be assessed in several ways, whether it is used to detect *honest-but-neglectful* or explicitly *malicious* participants. In this section, we review the existing literature on model quality assessment in FL and the related work on

Byzantine-robust FL. We review the existing approaches to detect and mitigate the impact of Byzantine contributions, and discuss the limitations of these methods in heterogeneous settings. We also review the existing works on reputation systems in FL and the use of clustering to improve the aggregation of local models.

1.3.1 Byzantine-resilient Federated Learning

Some approaches use evaluation to validate submitted models against a centralized dataset [Cao+22], or against randomly selected distributed datasets [PB23] if they are representative of each other—which is the case with Independent and Identically Distributed (IID) data partitioning. Given IID settings, submitted models can also be compared to each other [Bla+17; Cao+22; Ngu+22] or with a reference model [XTL21; Zho+22], using distance metrics. Among these, `FLAME` [Ngu+22] stands out, as it leverages multiple complementary methods to stop malicious participants: clustering to identify *multiple* groups of attackers, norm-clipping to mitigate gradient boosting attacks, and adaptive noising to lessen the impact of outliers. Yet, because it works under the assumption that the biggest cluster represents benign participants and that attackers cannot exceed 50% of the population, `FLAME` *de facto* falters against a majority of malicious clients. Furthermore, while the paper demonstrates that it can resist to low proportions of *NIID* participants, it still aims at delivering one common global model, thus failing to address the more skewed NIID cases, where leveraging multiple sub-federations might be necessary.

The assumption of IID data rarely holds in FL, even though its properties facilitate the detection of Byzantine participants. Indeed, given NIID settings, You *et al.* [You+22] show most of these mitigation strategies are inefficient. These methods rely on a single source of truth that may be known beforehand [Cao+22], or elected among participants [Bla+17]. However, by definition, this single source of truth does not exist in NIID datasets. To circumvent this issue, `FoolsGold` [FYB20] and `CONTRA` [ALL21] assume that sybils share a common goal, and thus produce similar model updates, allowing to distinguish them from benign NIID participants that present dissimilar contributions. Similar participants are classified as sybils using the cosine similarity between gradient updates, and their weight is reduced in the final aggregation. However, while this mitigation strategy works when multiple attackers collaborate, it fails at identifying lone attackers. These approaches are also well suited for *pathological NIID* scenarios, where all participants are significantly different. In *practical NIID* settings, legitimate communities of similar participants can exist. Those legitimate participants would be falsely identified as sybils.

Finally, Zhao *et al.* [Zha+20] take a different approach and rely on client-side evaluation. Local models are aggregated into multiple sub models, which are then randomly attributed to multiple clients for efficiency validation. To also address NIID datasets, clients self-report the labels on which they have enough data to conduct an evaluation.

While this self-reporting limits the network and client resources consumption, abusive self-reporting is possible. Nevertheless, directly leveraging the participant datasets for evaluation removes the need for a single exhaustive source of truth. Resource consumption is also less of an issue in cross-silo use cases: they often imply fewer participants, with more data and dedicated resources.

1.3.2 Clustered Federated Learning

NIID data can also be regarded as heterogeneous data distributions $\mathcal{P}_k$ that are regrouped together, where $\mathcal{P}_k$ is the distribution of the dataset d_k. Following this idea, some works [BFA20; Ouy+22; Ye+23] try to group participants sharing similarities. The purpose of this approach is twofold. First, from a performance perspective, NIID settings slow down convergence. Even if a global minimum is reached, the model might not be optimal for all participants [Kai+21; Ouy+22]. In addition, considering outliers as poisoned models [Per+20], one can eliminate thier in the aggregation process.

Since the effective number of clusters is unknown, hierarchical clustering is a common way to create appropriate clusters [BFA20; Ye+23]. Specifically, Ye *et al.* [Ye+23] use the cosine similarity of local models to successfully group participants in more homogeneous subgroups. However, as this approach doesn't aim to address Byzantines, it does not consider that some malicious participants might aim to be grouped with benign ones to poison the cluster's model. Another approach for finding the appropriate number of clusters is dynamic *split-and-merge* clustering [Che+21], where the number of clusters is adjusted depending on the distance between the participants' in each cluster. Finally, Ouyang *et al.* [Ouy+22] propose a clustering algorithm relying on K-means and spectral relaxation to group participants without prior knowledge of the number of clusters. Contrary to the most of the existing works, they do not use metrics that rely on vector representations of the models (such as cosine similarity, L2 norm, or scalar products). Rather, they leverage the Kullback-Leibler Divergence (KLD) to compare the models' probability distributions, which do not require the models to rely on a convex loss function.

1.3.3 Reputation systems for Federated Learning

In collaborative applications, reputation systems preemptively assess the ability of participants to perform a task and the quality of its result, based on past interactions. Definition 1.1 provides a formal definition of reputation systems. In the context of FL, they usually have three main applications: (i) client selection; (ii) model weighting and aggregation; and (iii) tracking contribution quality over time.

Definition 1.1: Reputation Systems

A reputation system collects, distributes, and aggregates feedback about participants past behavior. [. . .] To operate effectively, reputation systems require at least three properties:

- Long-lived entities that inspire an expectation of future interaction;
- Capture and distribution of feedback about current interactions (such information must be visible in the future); and
- Use of feedback to guide trust decisions.

– Resnick *et al.* [Res+00]

The first application, client selection, is used to determine which participants should be included in the training process of the next round [ALL21; Kan+20; Son+22; Tan+22]. This is particularly useful in scenarios with constrained resources [Son+22] and in hybrid architectures (see **??**) where servers can exchange reputation information about their users [Kan+20]. `CONTRA` [ALL21] provides an example of such a reputation system for client selection. By progressively penalizing the participants that propose models similar to each others, and that are thus suspected of being *sybils* (see Section 1.2 and **??**), it leaves room for participants issuing dissimilar models to be selected more often. We detail in **??** the limits of these types of approaches in practical NIID settings.

The second main application is to weight local models during the aggregation process [Wan+22; WK21]: the higher the reputation, the heavier the local model contributes to the aggregated model. Some will even go so far as to discard contributions when the author's reputation is too low. Karimireddy, He, and Jaggi [KHJ21] underline the importance of historical record in robust aggregation: malicious incremental changes can be small enough to be undetected in a single round but still eventually add up enough to poison the global model over the course of multiple round. Reputation system's ability to track clients' contributions over time [Kan+20; WK21] can be used as a countermeasure to these attacks.

Finally, note that the literature on reputation systems sometimes distinguishes between *reputation* and *trust* systems [Che+11; ZY15]. One of the main differences is the use of indirect feedbacks in reputation systems, wheras trust systems rely on direct evaluation an objective metrics. Based on this distinction, the reputation is the global perception of a one's trustworthiness in the system, based on the feedback of others [Che+11]. To the best of our knowledge, no work has yet been published in the context of FL that suit this definition.

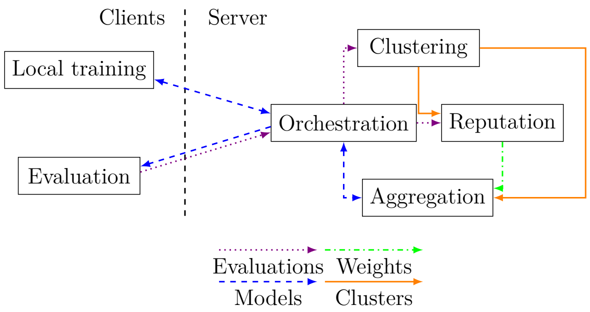

Figure 1.1 – *Architecture overview.*

1.4 Architecture

This section details RADAR's architecture. It is divided into three main components: (*i*) our cross-evaluation scheme that provides local feedbacks on each participant's contributions (Section 1.4.1), (*ii*) a similarity-based clustering algorithm that groups participants based on evaluations (**??**), and (*iii*) a reputation system that assesses participants' trustworthiness based on their past contributions (**??**). Figure 1.1 provides an overview of RADAR.

1.4.1 Assessing Contributions with Cross-Evaluation

As highlighted in Section 1.3, most related works on poisoning mitigation in FL rely on server-side models comparison [ALL21; FYB20]. They measure distance between the parameters (for Deep Neural Networks (DNNs), n-dimensional arrays containing the weights and biases of each neuron) using metrics such as cosine similarity [FYB20] or Euclidean distance [Ma+22]. However, models that are statistically further from others are not automatically of poor quality. To cope with this limitation, as well as the absence of source of truth, we propose to rely on client-side evaluation [Zha+20]. The results of this evaluation can then be used by the server to either discard or weight contributions. RADAR's workflow thus differs from typical approaches by adding an intermediate step for evaluating parameters:

1. *client fitting*—The server sends clients training instructions and initial parameters, *i.e.* randoms values for the first round. For subsequent rounds, the initial parameters of each client are initialized as the aggregated model (denoted $\bar{w}_k^{r-1}$) of the corresponding cluster, using the results of Step 3. at round $r-1$. Each client trains its own model using the provided hyperparameters, and the initial parameters as a starting point before uploading their parameters w_k^r to the server.
2. *cross-evaluation*—The server serializes all client parameters in a single list that is sent to every client. Each client then locally evaluates each received model using its validation set, generating a predefined set of metrics such as loss, accuracy, or

F1-score. The metrics of all clients are then gathered server-side.

3. *parameter aggregation*—The server partitions clients into a set of clusters $\mathscr{C}$ based on the evaluations gathered in Step 2. For each cluster $C_c \in \mathscr{C}$, the server computes the new model $\bar{w}_c^r = \sum_{k|p_k \in C_c^r} \rho_k^r w_k^r$, where the weight ρ_k^r is given by the reputation system for the participant p_k at round r.

Bibliography

[ALL21] Sana Awan, Bo Luo, and Fengjun Li, « CONTRA: Defending Against Poisoning Attacks in Federated Learning », *in*: *Computer Security – ESORICS 2021*, ed. by Elisa Bertino, Haya Shulman, and Michael Waidner, Lecture Notes in Computer Science, Cham: Springer International Publishing, 2021, pp. 455–475, ISBN: 978-3-030-88418-5, DOI: 10.1007/978-3-030-88418-5_22.

[BFA20] Christopher Briggs, Zhong Fan, and Peter Andras, « Federated Learning with Hierarchical Clustering of Local Updates to Improve Training on Non-IID Data », *in*: *2020 International Joint Conference on Neural Networks (IJCNN)*, 2020 International Joint Conference on Neural Networks (IJCNN), July 2020, pp. 1–9, DOI: 10.1109/IJCNN48605.2020.9207469.

[Bla+17] Peva Blanchard *et al.*, « Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent », *in*: *Advances in Neural Information Processing Systems* 30 (2017).

[Cao+22] Xiaoyu Cao *et al.*, *FLTrust: Byzantine-robust Federated Learning via Trust Bootstrapping*, Apr. 11, 2022, DOI: 10.48550/arXiv.2012.13995, arXiv: 2012.13995 [cs], URL: http://arxiv.org/abs/2012.13995 (visited on 08/09/2022), prepublished.

[Che+11] Dong Chen, Guiran Chang, *et al.*, « TRM-IoT: A Trust Management Model Based on Fuzzy Reputation for Internet of Things », *in*: *Computer Science and Information Systems* 8.4 (2011), pp. 1207–1228, ISSN: 1820-0214, 2406-1018, DOI: 10.2298/CSIS110303056C, URL: https://doiserbia.nb.rs/Article.aspx?ID=1820-02141100056C (visited on 07/03/2024).

[Che+21] Zheyi Chen, Pu Tian, *et al.*, « Zero Knowledge Clustering Based Adversarial Mitigation in Heterogeneous Federated Learning », *in*: *IEEE Transactions on Network Science and Engineering* 8.2 (Apr. 2021), pp. 1070–1083, ISSN: 2327-4697, DOI: 10.1109/TNSE.2020.3002796.

[Den+21] Yongheng Deng, Feng Lyu, Ju Ren, Yi-Chao Chen, *et al.*, « FAIR: Quality-Aware Federated Learning with Precise User Incentive and Model Aggregation », *in*: *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, IEEE INFOCOM 2021 - IEEE Conference on Computer Communications, Vancouver, BC, Canada: IEEE, May 10, 2021, pp. 1–10, ISBN: 978-1-66540-325-2, DOI: 10.1109/INFOCOM42981.2021.9488743, URL: https://ieeexplore.ieee.org/document/9488743/ (visited on 03/27/2024).

[Den+22] Yongheng Deng, Feng Lyu, Ju Ren, Huaqing Wu, *et al.*, « AUCTION: Automated and Quality-Aware Client Selection Framework for Efficient Federated Learning », *in*: *IEEE Transactions on Parallel and Distributed Systems* 33.*8* (Aug. 2022), pp. 1996–2009, ISSN: 1558-2183, DOI: 10.1109/TPDS.2021.3134647, URL: https://ieeexplore.ieee.org/abstract/document/9647925 (visited on 03/27/2024).

[FYB20] Clement Fung, Chris J.M. M Yoon, and Ivan Beschastnikh, « The Limitations of Federated Learning in Sybil Settings », *in*: *23rd International Symposium on Research in Attacks, Intrusions and Defenses ({RAID} 2020)*, San Sebastian: {USENIX} Association, Oct. 2020, pp. 301–316, ISBN: 978-1-939133-18-2, URL: https://www.usenix.org/conference/raid2020/presentation/fung.

[Hua+21] Yutao Huang *et al.*, « Personalized Cross-Silo Federated Learning on Non-IID Data », *in*: *Proceedings of the AAAI Conference on Artificial Intelligence* 35.*9* (May 18, 2021), pp. 7865–7873, ISSN: 2374-3468, 2159-5399, DOI: 10.1609/aaai.v35i9.16960, URL: https://ojs.aaai.org/index.php/AAAI/article/view/16960 (visited on 09/26/2022).

[Jai+20] Abhinav Jain *et al.*, « Overview and Importance of Data Quality for Machine Learning Tasks », *in*: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, New York, NY, USA: Association for Computing Machinery, Aug. 20, 2020, pp. 3561–3562, ISBN: 978-1-4503-7998-4, DOI: 10.1145/3394486.3406477, URL: https://dl.acm.org/doi/10.1145/3394486.3406477 (visited on 03/28/2024).

[Kai+21] Peter Kairouz *et al.*, « Advances and Open Problems in Federated Learning », Mar. 8, 2021, arXiv: 1912.04977 [cs, stat], URL: http://arxiv.org/abs/1912.04977 (visited on 04/01/2022).

[Kan+20] Jiawen Kang *et al.*, « Reliable Federated Learning for Mobile Networks », *in*: *IEEE Wireless Communications* 27.*2* (Apr. 2020), pp. 72–80, ISSN: 1558-0687, DOI: 10.1109/MWC.001.1900119.

[KHJ21] Sai Praneeth Karimireddy, Lie He, and Martin Jaggi, « Learning from History for Byzantine Robust Optimization », *in*: *Proceedings of the 38th International Conference on Machine Learning*, International Conference on Machine Learning, PMLR, July 1, 2021, pp. 5311–5319, URL: https://proceedings.mlr.press/v139/karimireddy21a.html (visited on 10/21/2022).

[Léo+24] **Léo Lavaur** *et al.*, « RADAR: Model Quality Assessment for Reputation-aware Collaborative Federated Learning », *in*: *Proceedings of the 43rd International Symposium on Reliable Distributed Systems (SRDS)*, Charlotte, NC, USA, Sept. 2024.

[Ma+22] Zhuoran Ma *et al.*, « ShieldFL: Mitigating Model Poisoning Attacks in Privacy-Preserving Federated Learning », *in*: *IEEE Transactions on Information Forensics and Security* 17 (2022), pp. 1639–1654, ISSN: 1556-6013, 1556-6021, DOI: 10.1109/TIFS.2022.3169918, URL: https://ieeexplore.ieee.org/document/9762272/ (visited on 07/05/2022).

[McM+17] Brendan McMahan *et al.*, « Communication-Efficient Learning of Deep Networks from Decentralized Data », *in*: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, ed. by Aarti Singh and Jerry Zhu, vol. 54, Proceedings of Machine Learning Research, PMLR, Apr. 20–22, 2017, pp. 1273–1282, URL: https://proceedings.mlr.press/v54/mcmahan17a.html.

[Ngu+22] Thien Duc Nguyen *et al.*, « FLAME: Taming Backdoors in Federated Learning », *in*: 31st USENIX Security Symposium (USENIX Security 22), 2022, pp. 1415–1432, ISBN: 978-1-939133-31-1, URL: https://www.usenix.org/conference/usenixsecurity22/presentation/nguyen (visited on 03/06/2024).

[Ouy+22] Xiaomin Ouyang *et al.*, « ClusterFL: A Clustering-based Federated Learning System for Human Activity Recognition », *in*: *ACM Transactions on Sensor Networks* 19.*1* (Dec. 8, 2022), 17:1–17:32, ISSN: 1550-4859, DOI: 10.1145/3554980, URL: https://dl.acm.org/doi/10.1145/3554980 (visited on 01/12/2024).

[PB23] Balázs Pejó and Gergely Biczók, « Quality Inference in Federated Learning With Secure Aggregation », *in*: *IEEE Transactions on Big Data* 9.*5* (Oct. 2023), pp. 1430–1437, ISSN: 2332-7790, DOI: 10.1109/TBDATA.2023.3280406, URL: https://ieeexplore.ieee.org/abstract/document/10138056 (visited on 03/27/2024).

[Per+20] Neehar Peri *et al.*, « Deep K-NN Defense Against Clean-Label Data Poisoning Attacks », *in*: *Computer Vision – ECCV 2020 Workshops*, ed. by Adrien Bartoli and Andrea Fusiello, Lecture Notes in Computer Science, Cham: Springer International Publishing, 2020, pp. 55–70, ISBN: 978-3-030-66415-2, DOI: 10.1007/978-3-030-66415-2_4.

[Res+00] Paul Resnick *et al.*, « Reputation Systems », *in*: *Communications of the ACM* 43.*12* (Dec. 1, 2000), pp. 45–48, ISSN: 0001-0782, DOI: 10.1145/355112.355122, URL: https://doi.org/10.1145/355112.355122 (visited on 02/01/2023).

[Son+22] Zhendong Song *et al.*, « Reputation-Based Federated Learning for Secure Wireless Networks », *in*: *IEEE Internet of Things Journal* 9.*2* (Jan. 2022), pp. 1212–1226, ISSN: 2327-4662, DOI: 10.1109/JIOT.2021.3079104.

[Tan+22] Xavier Tan *et al.*, « Reputation-Aware Federated Learning Client Selection Based on Stochastic Integer Programming », *in*: *IEEE Transactions on Big Data* (2022), pp. 1–12, ISSN: 2332-7790, DOI: 10.1109/TBDATA.2022.3191332.

[Wan+22] Ning Wang, Yang Xiao, *et al.*, « FLARE: Defending Federated Learning against Model Poisoning Attacks via Latent Space Representations », *in*: *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*, ASIA CCS '22: ACM Asia Conference on Computer and Communications Security, Nagasaki Japan: ACM, May 30, 2022, pp. 946–958, ISBN: 978-1-4503-9140-5, DOI: 10.1145/3488932.3517395, URL: https://dl.acm.org/doi/10.1145/3488932.3517395 (visited on 07/05/2022).

[WK21] Yuwei Wang and Burak Kantarci, « Reputation-Enabled Federated Learning Model Aggregation in Mobile Platforms », *in*: *ICC 2021 - IEEE International Conference on Communications*, ICC 2021 - IEEE International Conference on Communications, June 2021, pp. 1–6, DOI: 10.1109/ICC42927.2021.9500928.

[XTL21] Qi Xia, Zeyi Tao, and Qun Li, « ToFi: An Algorithm to Defend Against Byzantine Attacks in Federated Learning », *in*: *Security and Privacy in Communication Networks*, ed. by Joaquin Garcia-Alfaro *et al.*, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Cham: Springer International Publishing, 2021, pp. 229–248, ISBN: 978-3-030-90019-9, DOI: 10.1007/978-3-030-90019-9_12.

[Ye+23] Chuyao Ye *et al.*, « PFedSA: Personalized Federated Multi-Task Learning via Similarity Awareness », *in*: *2023 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 2023 IEEE International Parallel and Distributed Processing Symposium (IPDPS), May 2023, pp. 480–488, DOI: 10.1109/IPDPS54959.2023.00055, URL: https://ieeexplore.ieee.org/document/10177489 (visited on 12/05/2023).

[You+22] XinTong You *et al.*, « Poisoning Attack Detection Using Client Historical Similarity in Non-Iid Environments », *in*: *2022 12th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, 2022 12th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Jan. 2022, pp. 439–447, DOI: 10.1109/Confluence52989.2022.9734158.

[Zha+20] Lingchen Zhao *et al.*, *Shielding Collaborative Learning: Mitigating Poisoning Attacks through Client-Side Detection*, Mar. 9, 2020, arXiv: 1910.13111 [cs], URL: http://arxiv.org/abs/1910.13111 (visited on 08/28/2022), pre-published.

[Zho+22] Jun Zhou *et al.*, « A Differentially Private Federated Learning Model against Poisoning Attacks in Edge Computing », *in*: *IEEE Transactions on Dependable and Secure Computing* (2022), pp. 1–1, ISSN: 1941-0018, DOI: 10.1109/TDSC.2022.3168556.

[ZY15] Fatima Zohra Filali and Belabbes Yagoubi, « Global Trust: A Trust Model for Cloud Service Selection », *in*: *International Journal of Computer Network and Information Security* 7.5 (Apr. 8, 2015), pp. 41–50, ISSN: 20749090, 20749104, DOI: 10.5815/ijcnis.2015.05.06, URL: http://www.mecs-press.org/ijcnis/ijcnis-v7-n5/v7n5-6.html (visited on 07/03/2024).

List of Figures

LIST OF TABLES

COLLEGES BRETAGNE LOIRE | SCIENCES POUR L'INGENIEUR ET LE NUMERIQUE

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Titre : Améliorer la détection d'intrusions dans les systèmes répartis grâce à l'apprentissage fédéré

Mot clés : apprentissage automatique, apprentissage fédéré, détection d'intrusions, collaboration, données hétérogènes, confiance

Résumé : La collaboration entre les différents acteurs de la cybersécurité est essentielle pour lutter contre des attaques de plus en plus sophistiquées et nombreuses. Pourtant, les organisations sont souvent réticentes à partager leurs données, par peur de compromettre leur confidentialité et leur avantage concurrentiel, et ce même si cela pourrait d'améliorer leurs modèles de détection d'intrusions. L'apprentissage fédéré est un paradigme récent en apprentissage automatique qui permet à des clients répartis d'entraîner un modèle commun sans partager leurs données. Ces propriétés de collaboration et de confidentialité en font un candidat idéal pour des applications sensibles comme la détection d'intrusions. Si un certain nombre d'applications ont montré qu'il est, en effet, possible d'entraîner un modèle unique sur des données réparties de détection d'intrusions, peu se sont intéressées à l'aspect collaboratif de ce paradigme. En plus de l'aspect collaboratif, d'autres problématiques apparaissent dans ce contexte, telles que l'hétérogénéité des données des différents participants ou la gestion de participants non fiables. Dans ce manuscrit, nous explorons l'utilisation de l'apprentissage fédéré pour construire des systèmes collaboratifs de détection d'intrusions. En particulier, nous explorons (i) l'impact de la qualité des données dans des contextes hétérogènes, (ii) certains types d'attaques par empoisonnement, et (iii) proposons des outils et des méthodologies pour améliorer l'évaluation de ce type d'algorithmes répartis.

Title: Improving Intrusion Detection in Distributed Systems with Federated Learning

Keywords: machine learning, federated learning, intrusion detection, collaboration, heterogeneous data, trust

Abstract: Collaboration between different cybersecurity actors is essential to fight against increasingly sophisticated and numerous attacks. However, stakeholders are often reluctant to share their data, fearing confidentiality and privacy issues and the loss of their competitive advantage, although it would improve their intrusion detection models. Federated learning is a recent paradigm in machine learning that allows distributed clients to train a common model without sharing their data. These properties of collaboration and confidentiality make it an ideal candidate for sensitive applications such as intrusion detection. While several applications have shown that it is indeed possible to train a single model on distributed intrusion detection data, few have focused on the collaborative aspect of this paradigm. In addition to the collaborative aspect, other challenges arise in this context, such as the heterogeneity of the data between different participants or the management of untrusted contributions. In this manuscript, we explore the use of federated learning to build

collaborative intrusion detection systems. In particular, we explore (i) the impact of data quality in heterogeneous contexts, (ii) some types of poisoning attacks, and (iii) propose tools and methodologies to improve the evaluation of these types of distributed algorithms.