COLLEGES SCIENCES
BRETAGNE POUR L'INGENIEUR
LOIRE 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 N° 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

Chapter 1

Performance and Limitations of FIDSs

1.1 Introduction

In the previous chapters, we have discussed the perspectives offered by applying Federated Learning (FL) to Intrusion Detection Systems (IDSs), notably in terms of collaboration. Based on the insights gained from the literature, it is now clear that FL can be used to train a global model over the distributed data of a federation of organizations. It even seems that FL could be used to share attack knowledge, still without sharing participants' local data.

In this chapter, we present critical examples showing the challenges that arise when applying FL to Collaborative Intrusion Detection Systems (CIDSs). We start by laying out in Section 1.2 the practical use case that will be used throughout the rest of the manuscript. Then, we highlight some limitations of FL in the context of CIDSs in **??**, based on our demonstration paper published at ICDCS 2024 [LBA24].

Contributions of this chapter

- A practical use case for FL in the context of CIDSs involving multiple organizations.
- A demonstration of the limitations of FL in the context of CIDSs, notably in terms of data heterogeneity and susceptibility to poisoning attacks.

1.2 A Practical Use Case for FIDSs

We consider a typical FL scenario where a central server S is tasked with aggregating the model updates w_k^r of a set of participants $P = \{p_k | k \in [\![1, n]\!]\}$ at each round r. The participants p_k are entities that oversee an organization's network, which makes them highly available and interested. This can be described as a Cross-Silo Federated Learning (CS-FL) scenario, *i.e.*, fewer participants with consequent amounts of data and

significant computing capabilities. Because of the lower scale of the federation and the assumed interest of the different parties, we set the fraction C of participants that are selected at each round to 1.

For the sake of simplicity, we consider that all participants share the same model architecture and extract the same features from the network traffic. This is not unrealistic, as common formats and protocols are used in the industry for this purpose, such as Cisco's NetFlow format [Cla04] for network flows. Further, this description can fit multiple scenarios, such as organizations deploying the same probe in their network as part of a standardization effort, or a service provider offering a gray-box product to multiple organizations. Although the features are assumed to be identical across participants, the distribution of the data can vary considerably, as each organization has its own network configuration and security policies [ZLK10].

We also consider that participants have access to labeled data, which is a common assumption in the literature. Although labeling data can be costly, it is a more reasonable assumption in CS-FL scenarios, where participants are more likely to have the human and financial resources to label data. Therefore, each participant possesses a local dataset $d_k = (\mathcal{X}_k, \mathcal{Y}_k)$ that is not shared with the others. Because of the differences between organizations, the distribution of each local dataset d_k can vary considerably, independently of the associated labels. Indeed, the same network behavior (say Peer-to-Peer (P2P) file sharing) might be considered normal in an organization (*e.g.*, a media company) but flagged as suspicious or outright malicious in another (*e.g.*, a financial institution). However, the CIDS use case implies that similarities can exist between participants, for instance between organizations operating in the same sector or having similar network infrastructure. This particular setting can be described as *practical* Non Independent and Identically Distributed (NIID), as opposed to the *pathological* NIID settings, where all participants have unique and highly different data-distributions [Hua+21]. This is the most common setting in Federated Intrusion Detection Systems (FIDSs), as it serves the goal of improving behavior characterization, and having access to knowledge that cannot be inferred with only local data.

1.2.1 Dataset selection

Since we consider that all organizations share the same model architecture, we need multiple independently-generated datasets that share the same feature set. Fortunately, Sarhan, Layeghy, and Portmann [SLP22] have proposed a standard feature set for IDS datasets, based on NetFlow v9 (see **??**). Namely, we used the modified versions of the following datasets:

— UNSW-NB15 [MS15] is produced using the IXIA PerfectStorm tool on the Cyber Range Lab of UNSW Canberra. The traffic is a hybrid set of real modern normal

activities and synthetic contemporary attack behaviors, grouped in 9 attack classes.

— Bot-IoT [Kor+19] is another dataset generated at USNW, using a realistic smart home environment setup, completed by IoT devices. It focuses on the detection of IoT botnet attacks, the DoS and DDoS classes being the most represented. This dataset is highly unbalanced, as the majority of the traffic is malicious.

— ToN_IoT [Mou+20] is yet another dataset generated by the same team, containing IoT/IIoT telemetry data, network traffic, as well as system logs. The network dataset contains 9 attack classes, including Ransomware, Scanning, and XSS.

— CSE-CIC-IDS2018 [SHG18] is a dataset generated by the Canadian Institute for Cybersecurity in collaboration with the Communications Security Establishment (CSE). The traffic is collected on a large-scale infrastructure deployed on AWS. It contains 14 attack labels, grouped in 6 attack classes.

In most of the experiments presented in this manuscript, We use the "sampled" version (1,000,000 data points per dataset) provided by the same team [LP22] We remove the port and IP addresses for both source and destination, as they are rather a representation of the network topology and device configurations than of traffic patterns [dCar+23]. We then use one-hot encoding (see **??**) on the categorical features (both in the sample and labels), and apply min-max normalization to give all features the same importance in model training. This pre-processing step produces 39 features for each sample.

1.3 Exhibiting the Limits of FIDSs

This demonstration spans over four specific scenarios, each highlighting a specific aspect of the considered challenges. The first three (Sections 1.3.2 to 1.3.4) target different heterogeneity scenarios, ranging from homogeneous dataset partitioning to completely independent data sources. The last scenario (Section 1.3.5) focuses on poisoning attacks against FL, where malicious participants try to degrade the performance of the global model.

1.3.1 Setup

To evaluate the performance of FL in the context of CIDSs, and especially evaluate the feasibility of the scenario presented in Section 1.2, we need datasets that are representative of the traffic that can be observed in real-world networks. Consequently, we use the datasets mentioned in Section 1.2 with the NF-V2 format, which allows us to use the same model architecture for all participants.

To generate the different scenarios, we build an evaluation framework for FL called Eiffel[1] [LBA24], which relies on Flower [Beu+20], a modular FL framework. Eiffel is a

1. Available at: https://github.com/phdcybersec/eiffel

Table 1.1 – Parameters used for all scenarios.

Parameter	**Notation**	**Value**
Federated Learning		
Number of rounds	R	10
Local epochs per round	$\mathcal{E}$	10
Number of clients	K	4
Local Training		
Neurons of the (2) hidden layers		128
Activation function (hidden layers)		ReLU
Activation function (output layer)		Softmax
Batch size	β	512
Learning rate	η	0.001
Datasets		
Number of features		39
Number of samples		100,000

Python library that provides a set of tools to automate the evaluation of FL algorithms, such as instantiating various types of data distribution, local models, and aggregation strategies. It further provides multiple label-flipping attacks, and automates metric collection and plotting to quickly evaluate the impact of each parameter.

To assess the impact of a scenario on the federation, we evaluate the global model on each participant's test set and collect different performance metrics. The results are averaged over the different participants to obtain the global model's performance. We select the F1-score as the main metric for its focus on positive samples, but the same methodology can be applied to other metrics. To assess the performance of a model trained only on local data, we define a `FedNoAgg` strategy, where local models are kept by participants at the end of each round. Therefore, models are trained during $\varepsilon \times R$ local epochs, where R is the number of rounds and ε is the number of local epochs per round, instructed by the server. Table 1.1 summarizes the parameters used for all scenarios, with the notations defined in **??**.

1.3.2 Scenario 1: IID Data

The first scenario is the simplest one, where the data is partitioned in Independent and Identically Distributed (IID) settings. Each participant receives $\frac{N}{C}$ samples, after shuffling the dataset. Figure 1.1 presents the results of this scenario based on the global model's F1-score. There are virtually no differences between the `FedNoAgg` and `FedAvg` strategies, since each participant has enough samples of each class to train a suitable local model. Therefore, there are little benefit to using FL in this scenario.

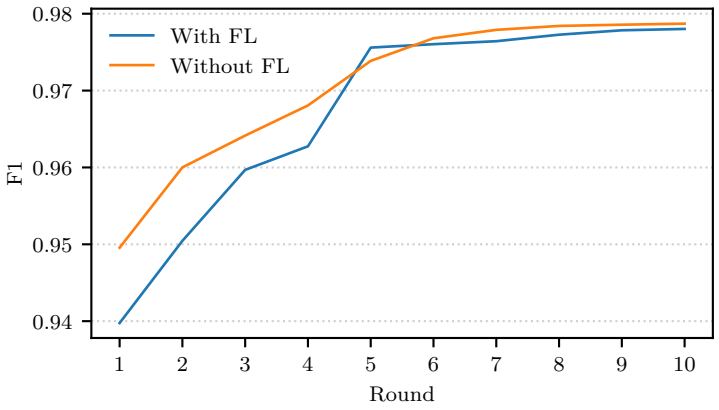

Figure 1.1 – Global model performance in IID.

However, this configuration is often found in the literature to evaluate CIDSs based on FL, such as in [Aou+22]. While this experiment illustrates the lack of performance gains on IID data, larger-scale setups configurations might benefit from FL. In fact, selecting only a subset of the available participants could obtain similar results while reducing the local computing costs for participants. This setup is thus more akin to a distributed learning approach, where the server is only used to coordinate the training process.

1.3.3 Scenario 2: NIID Data from the Same Source

The second scenario highlights the knowledge-sharing capabilities of FL, as it can transfer characteristics of the data distribution between participants. To illustrate this, after partitioning the data as in Section 1.3.2, we randomly drop two classes from each participant's train set. This results in a NIID data distribution among participant, where each one has a different subset of classes. Figure 1.2 displays the results of this scenario, where `FedAvg` performs significantly better overall than having clients train locally. However, the F1-score hides the fact that some participants can miss entire attack classes in the test set, rather than it being a global model issue.

Specifically, since clients have different subsets of classes, they might be unable to detect some intrusions that are not present in their training data. For example, Table 1.2 displays the Detection Rate (DR) of the first client (`client_0`) in our setup for each attack class, both in local and federated training, along with the number of samples of each class. `client_0` has no samples of the `Infiltration` and `DoS` classes, and therefore cannot detect them, *i.e.* its DR is either 0 or very low. However, the global model is able to detect these classes, as other clients have samples of these classes in their training set. We also see a slight decrease in performance for the other classes (*e.g.*, 99.91 instead of 100 for `DDoS`) due to the aggregation process. Note that the `Infiltration` being only detected at 20.11% by the global model is the expected behavior on this dataset, as it is particularly difficult to detect (see the baseline results in **??** for more details).

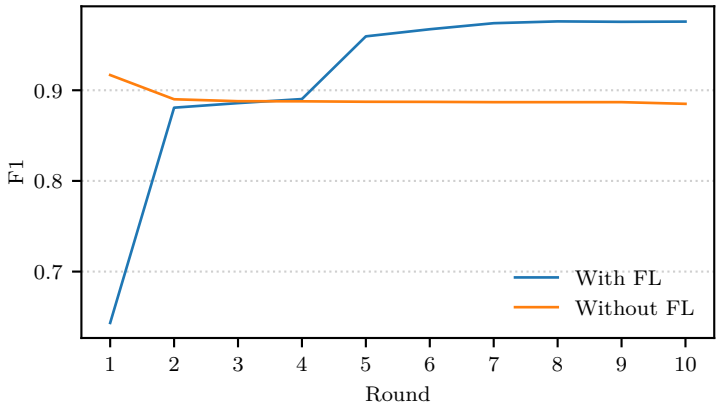

Figure 1.2 – Global model performance in NIID (same source).

Table 1.2 – Detection rate (DR) of `client_0` in NIID settings. Rows where knowledge-sharing is visible are highlighted in gray.

Attack class	Samples	DR (local)	DR (federated)
DDoS	176107	100	99.91
DoS	0	2.43	98.57
Bot	1513	100	99.94
Brute force	1299	99.77	99.55
Infiltration	0	0	20.11
Injection	3	100	100

These results indicate that FL can effectively share knowledge between participants, allowing them to detect attacks that are not present in their local training data. This is a key feature of FL in the context of intrusion detection.

1.3.4 Scenario 3: NIID Data from Different Sources

While we highlight in Section 1.3.3 that FL can benefit from having different datasets per client, to the point where it can share knowledge between participants, the third scenario illustrates the limits of this assumption. CIDS experiments in the literature often evaluate their approach with a scenario close to the ones presented in Sections 1.3.2 and 1.3.3, where one dataset is partitioned among participants. However, in practice, participants will likely collect data from different networks, and therefore have different data distributions.

In this third scenario, we test `FedAvg` in this configuration, with each participant having a different dataset. Thanks to the standardized feature set (see Section 1.3.1), we can use the same model architecture for all participants, which is a requirement for `FedAvg`. The class overlap between datasets is also not an issue in this use case, as we focus on binary-classification, which implies that all participants have benign and malicious samples.

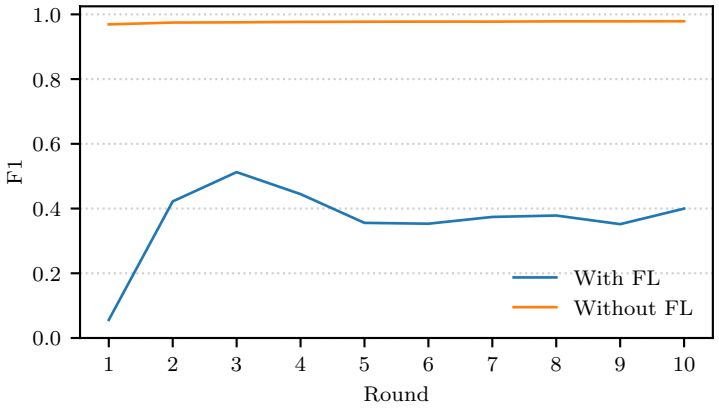

Figure 1.3 – Global model performance in NIID (different sources).

The results presented in Figure 1.3 confirm great performances overall when participants are trained locally. However, the global model's performance is highly impacted by the heterogeneity of the data distributions. This is likely due to the fact that all participants converge to local minima that are too different from each other, and therefore the aggregation do not result in a suitable model for all participants. Other approaches than `FedAvg` have been proposed to address this issue in IDS context, as the one by Popoola *et al.* [Pop+21] for instance, who use `Fed+` [Kun+22] as the aggregation strategy and present promising results in a similar scenario.

1.3.5 Scenario 4: Poisoning Attacks

With the first three scenarios, we have highlighted how the heterogeneity between participants can impact the performance of FL. However, these scenarios assume that participants are honest and respect the protocol. In this last scenario, we demonstrate how FL can be vulnerable to malicious participants, whose goal is to degrade the performance of the global model. To do so, we use poisoning attacks (see **??**), where attackers flip the labels of samples in their training data to degrade the performance of the global model.

In order to observe the impact in an extreme scenario, two of the four clients are instructed to perform a label flipping attack on their entire training set. We can observe in local training (Figure 1.4) that participants identified as "Attackers" have a very low DR on their test set, as they literally misclassify all of their testing samples. The two benign participants, on the contrary, reproduce the results of Section 1.3.2, with a high DR on their test set.

In FL however, the global model is impacted by the malicious participants, as illustrated in Figure 1.4. The participants cannot converge towards a stable global model, as the malicious participants' updates are too different from the others. Due to the misclassification introduced by the malicious participants, the global model's performance is degraded, and the F1-score oscillates between 0.1 and 0.2. This is critically low, as it

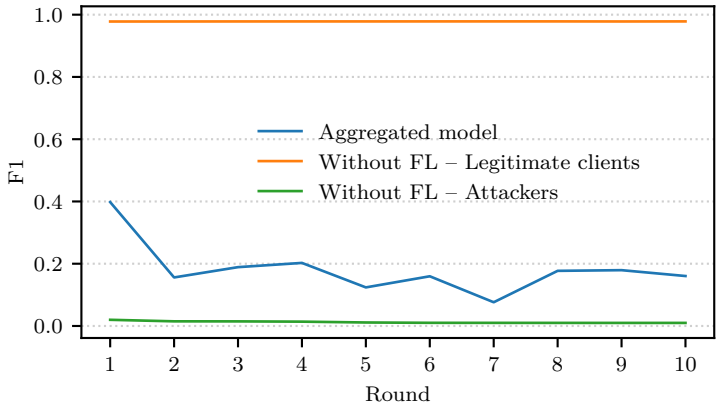

Figure 1.4 – Global model performance in poisoning attacks.

means that the aggregated model either misses a lot of attacks and misclassifies a lot of benign samples. A more in-depth analysis of the impact of poisoning attacks on FL is presented in **??**.

1.4 Conclusion and Takeways

In this chapter, we have presented a practical use case for FL in the context of CIDSs. This use case will be used throughout the rest of the manuscript to illustrate the different contributions and results. Based on this use case, we have also exposed some limitations of FIDSs, notably in terms of data heterogeneity and susceptibility to poisoning attacks. We will explore these limitations further in the next chapters: the impact of data heterogeneity in **??**, and the impact of poisoning attacks in **??**. Finally, we will present some solutions to these limitations in **??** and **??**.

Part II

Quantifying the Limitations of FIDSs

Part III

Providing Solutions

BIBLIOGRAPHY

[Aou+22] Ons Aouedi *et al.*, « Intrusion Detection for Softwarized Networks with Semi-supervised Federated Learning », *in*: *ICC 2022 - IEEE International Conference on Communications*, ICC 2022 - IEEE International Conference on Communications, May 2022, pp. 5244–5249, DOI: 10.1109/ICC45855.2022.9839042, URL: https://ieeexplore.ieee.org/document/9839042 (visited on 04/25/2024).

[Beu+20] Daniel J Beutel *et al.*, « Flower: A Friendly Federated Learning Research Framework », 2020, arXiv: 2007.14390.

[Cla04] Benoît Claise, *Cisco Systems NetFlow Services Export Version 9*, RFC 3954, RFC Editor, Oct. 2004, DOI: 10.17487/RFC3954, URL: https://www.rfc-editor.org/info/rfc3954.

[dCar+23] Gustavo de Carvalho Bertoli *et al.*, « Generalizing Intrusion Detection for Heterogeneous Networks: A Stacked-Unsupervised Federated Learning Approach », *in*: *Computers & Security* 127 (Apr. 1, 2023), p. 103106, ISSN: 0167-4048, DOI: 10.1016/j.cose.2023.103106, URL: https://www.sciencedirect.com/science/article/pii/S0167404823000160 (visited on 03/14/2023).

[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).

[Kor+19] Nickolaos Koroniotis *et al.*, « Towards the Development of Realistic Botnet Dataset in the Internet of Things for Network Forensic Analytics: Bot-IoT Dataset », *in*: *Future Generation Computer Systems* 100 (Nov. 2019), pp. 779–796, ISSN: 0167739X, DOI: 10.1016/j.future.2019.05.041, URL: https://linkinghub.elsevier.com/retrieve/pii/S0167739X18327687 (visited on 10/23/2021).

[Kun+22] Achintya Kundu *et al.*, « Robustness and Personalization in Federated Learning: A Unified Approach via Regularization », *in*: *2022 IEEE International Conference on Edge Computing and Communications (EDGE)*, 2022 IEEE International Conference on Edge Computing and Communications (EDGE),

July 2022, pp. 1–11, DOI: 10.1109/EDGE55608.2022.00014, URL: https://ieeexplore.ieee.org/document/9860349 (visited on 07/01/2024).

[LBA24] **Léo Lavaur**, Yann Busnel, and Fabien Autrel, « Demo: Highlighting the Limits of Federated Learning in Intrusion Detection », *in*: *Proceedings of the 44th International Conference on Distributed Computing Systems (ICDCS)*, Jersey City, NJ, USA, July 2024.

[LP22] Siamak Layeghy and Marius Portmann, *On Generalisability of Machine Learning-based Network Intrusion Detection Systems*, May 9, 2022, arXiv: 2205.04112 [cs], URL: http://arxiv.org/abs/2205.04112 (visited on 03/23/2023), pre-published.

[Mou+20] Nour Moustafa, Marwa Keshky, *et al.*, « Federated TON_IoT Windows Datasets for Evaluating AI-Based Security Applications », *in*: *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Dec. 2020, pp. 848–855, DOI: 10.1109/TrustCom50675.2020.00114.

[MS15] Nour Moustafa and Jill Slay, « UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems (UNSW-NB15 Network Data Set) », *in*: *2015 Military Communications and Information Systems Conference (MilCIS)*, 2015 Military Communications and Information Systems Conference (MilCIS), Nov. 2015, pp. 1–6, DOI: 10.1109/MilCIS.2015.7348942, URL: https://ieeexplore.ieee.org/abstract/document/7348942 (visited on 10/09/2023).

[Pop+21] Segun I. Popoola *et al.*, « Federated Deep Learning for Collaborative Intrusion Detection in Heterogeneous Networks », *in*: *2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall)*, 2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall), Sept. 2021, pp. 1–6, DOI: 10.1109/VTC2021-Fall52928.2021.9625505.

[SHG18] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani, « Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization », *in*: *Proceedings of the 4th International Conference on Information Systems Security and Privacy*, 4th International Conference on Information Systems Security and Privacy, Funchal, Madeira, Portugal: SCITEPRESS - Science and Technology Publications, 2018, pp. 108–116, ISBN: 978-989-758-282-0, DOI: 10.5220/0006639801080116, URL: http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0006639801080116 (visited on 10/14/2021).

[SLP22] Mohanad Sarhan, Siamak Layeghy, and Marius Portmann, « Towards a Standard Feature Set for Network Intrusion Detection System Datasets », *in*: *Mobile Networks and Applications* 27.*1* (Feb. 1, 2022), pp. 357–370, ISSN: 1572-8153, DOI: 10.1007/s11036-021-01843-0, URL: https://doi.org/10.1007/s11036-021-01843-0 (visited on 04/23/2024).

[ZLK10] Chenfeng Vincent Zhou, Christopher Leckie, and Shanika Karunasekera, « A Survey of Coordinated Attacks and Collaborative Intrusion Detection », *in*: *Computers & Security* 29.*1* (Feb. 2010), pp. 124–140, ISSN: 01674048, DOI: 10.1016/j.cose.2009.06.008, URL: https://linkinghub.elsevier.com/retrieve/pii/S016740480900073X (visited on 07/21/2021).

List of Figures

List of Tables

APPENDICES

E Additional figures

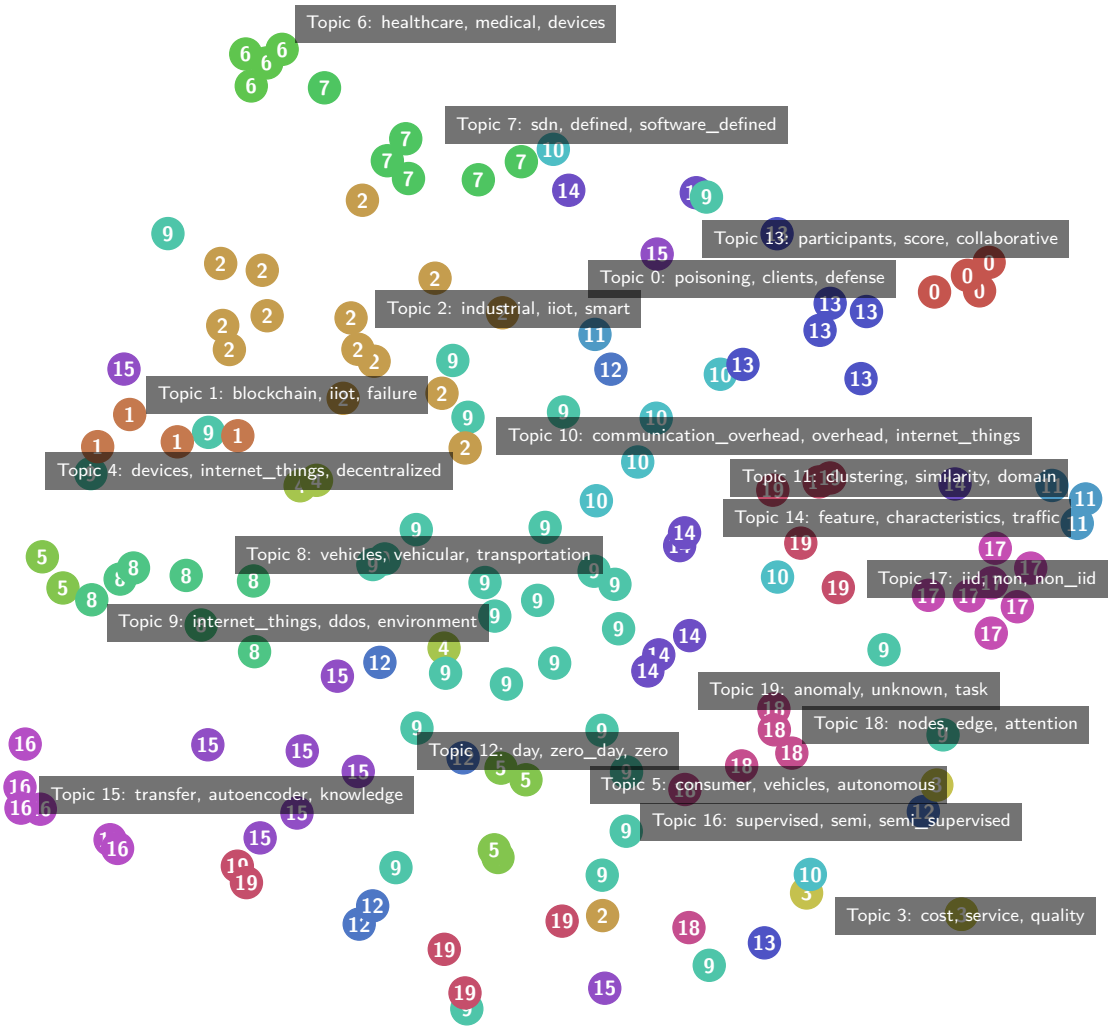

Figure 1.5 – Topic embedding of the Federated Intrusion Detection System (FIDS) literature using a Non-negative Matrix Factorization (NMF) model with 20 topics. Each point represents a paper, and each are labelled with the topic they are the most associated with.

F Résumé en français de la thèse

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.