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 des systèmes distribués grâce à l'apprentissage fédéré

Thèse présentée et soutenue à Rennes, le XX septembre 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 CNRS au laboratoire LIRIS
	Pierre-François Gimenez	Maître de Conférence à CentraleSupelec
	Vincent Nicomette	Professeur à l'INSA de Toulouse
	Fabien Autrel	Ingénieur de Recherche à IMT Atlantique
	Marc-Oliver Pahl	Directeur d'Études à IMT Atlantique
Dir. de thèse :	Yann Busnel	Directeur de la Recherche et de l'Innovation (DRI) 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 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 distribués 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 distribuées 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 l'impact de la qualité des données dans des contextes hétérogènes, certains types d'attaques par empoisonnement, et proposons des outils et des méthodologies pour améliorer l'évaluation de ce type d'algorithmes distribués.

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, 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 the impact of data quality in heterogeneous contexts, some types of poisoning attacks, and propose tools and methodologies to improve the evaluation of these types of distributed algorithms.

ACKNOWLEDGEMENTS

Table of Contents

Chapter 1

Introduction

Contents

1.1 Context and Motivation

Modern information security is made difficult by the scale, complexity, and heterogeneity of information systems. Because security by design in these conditions is an impossible task, security agencies also recommend complementary measures. For instance, the NIST Cybersecurity Framework [Nat24] suggests a five-stage lifecycle for managing risks in information systems: identify, protect, detect, respond, and recover.

The *detection* and *response* stages immensely benefit from the recent advances in Artificial Intelligence (AI) and Machine Learning (ML), enabling the analysis of more complex behaviors. Yet, because organizations usually face similar threats, including large-scale campaigns such as Mirai in 2016 or NotPetya in 2017, they would greatly benefit from sharing insights on the intrusions they have encountered, or any knowledge that might help others to identify the incident before the damages are too important. Collaboration is further encouraged by regulation, for instance with the NIS [16] and NIS2 [22] European directives. Sharing data is made even more important for training ML and Deep Learning (DL) models, which require large amounts of data to be effective. Yet, stakeholders are often reluctant to involve their organization in data-sharing practices, fearing confidentiality and privacy breaches, reputation loss, or regulation non-compliance.

Federated Learning (FL) [McM+17] has emerged as a promising paradigm for collaborative ML, enabling model training across distributed data sources while preserving privacy. Deployed in intrusion detection contexts, FL can help organizations to virtually extend the size of their training sets, thus producing more accurate models. This architecture could also be used to disseminate information about esoteric attacks or devices behavior owned locally, that would benefit to other organizations. FL also promises to solve other drawbacks of ML-based Intrusion Detection Systems (IDSs), such as the need for continuous retraining[**?**], the lack of adaptability to new threats[**?**], or the risk of local biases due to a lack of heterogeneity in the training set[**?**].

Consequently, applying FL to IDS seems like a promising approach to collaboratively improve the local detection of cyber threats. This is supported by the amount of recent literature on the topic, which has grown exponentially since 2018 [Lav+22b; Ism+24]. Yet, novel challenges arise in this context, such as how to handle the heterogeneity of data sources or how to deal with untrusted participants. But more importantly, *what makes applying FL to IDS different from other applications? And is FL even a suitable framework for collaborative IDS?*

This dissertation aims to investigate the potential of Federated Learning as a collaborative framework for Intrusion Detection System, which we will refer to as Federated Intrusion Detection System (FIDS). The remaining of this manuscript will discuss the state of the art in FL and IDS, some of the challenges that arise in this context, and the potential solutions to address them.

1.1.1 Use case boundaries

While applying FL to IDS can already be considered as a restricted scope, the IDS literature contains a wide variety of use cases, each coming with its own set of specificities and constraints. For instance, IDSs can be deployed at the network level, the host level, or the application level. Likewise, objectives and constraints may vary depending on the context and the type of devices involved: Internet of Things (IoT), Industrial Control System (ICS), or traditional information systems. Among the most common combinations, Network-based Intrusion Detection System (NIDS) on Information Technology (IT) network data stands out, notably in terms of implemented algorithms and available datasets. This is particularly important for evaluation purposes, as it makes it easier to compare the performance of different approaches.

Additionally, this use case provides a realistic application for FIDSs, where the actors are organizations that own or oversee an information system, and that are interested in improving their local detection. This is typically referred to as Collaborative Intrusion Detection System (CIDS). For instance, Security Operations Centers (SOCs) monitor the network traffic of their customers for security purposes, and cannot afford to share this data with other organizations. Two SOCs could, however, share insights on the threats they have encountered, or the behaviors they have observed, without sharing the raw data. Existing structures, such as Information Sharing and Analysis Centers (ISACs) or inter-SOCs could benefit from such a framework, as they already have a trust relationship with their members.

Consequently, this dissertation will focus on the use case of building collaborative NIDSs by leveraging FL on IT network data. Note however, that the results presented in this manuscript could theoretically be extended to other applications. Figure 1.1 illustrates

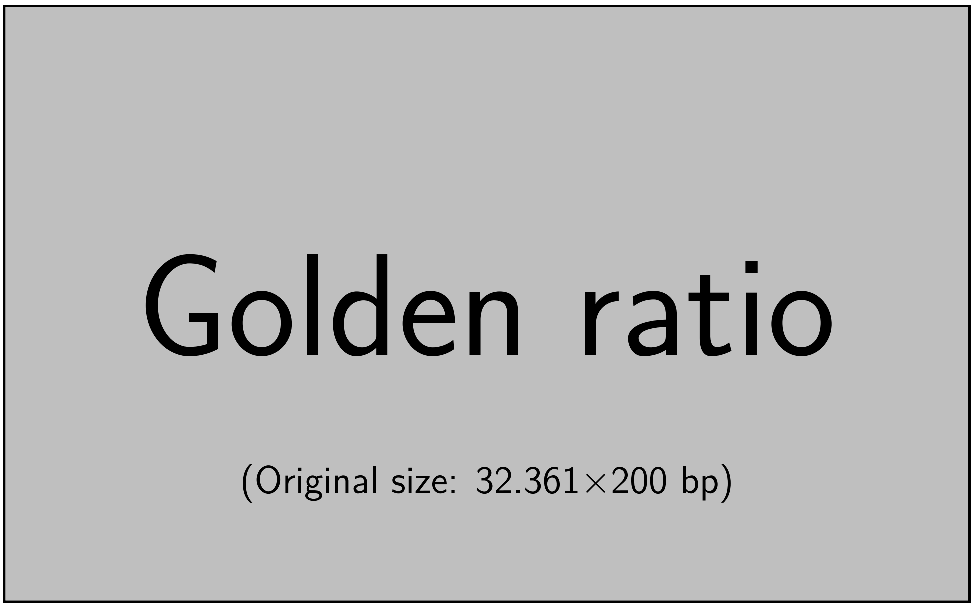


Figure 1.1 – Illustration of FL in a CIDS use case.

this use case.

1.1.2 Research Objectives

Based on the context and motivation laid out in the previous section, we formulize the general objectives of this dissertation as a set of research questions. The questions stated hereafter are intended to be completed and extended in the following chapters, some of which introduce their own research questions. Overall, this work aims to answer the following: *Can FL serve as a trustable knowledge-sharing framework for collaboratively improving intrusion detection mechanisms?*

Specifically, we focus on the following research questions:

RQ1. *What makes applying FL to IDSs specific?*

RQ2. *Can FL be used to federate IDSs across heterogeneous data sources?*

RQ3. *How does FL handle malicious contributions in a federated IDS?*

RQ4. *How can we address the main challenges in applying FL to IDS?*

1.2 Contributions

We summarize the contributions of this dissertation as follows:

1. The first Systematic Literature Review (SLR) on applying FL to IDS, with quantitative and qualitative analyses of the existing works, as well as the proposal of a reference architecture and a taxonomy for structuring the domain.

2. A demonstration highlighting the challenges of heterogeneity and malicious contributions in FIDS.
3. An extensible evaluation framework for FIDSs called Eiffel, leveraging popular open source libraries like Flower [Beu+20] and Hydra [Yad19], and a set of malicious clients simulators.
4. A systematic analysis of the impact of label-flipping attacks on an FL-based collaborative IDS, leveraging the aforementioned evaluation framework.
5. The first FL architecture for collaborative IDS that handles malicious contributions in heterogeneous environments, leveraging a cross-evaluation mechanism and a reputation system.
6. A methodology allowing to generate network topologies with heterogeneity constraints, and laying down the foundations toward a more realistic evaluation of FIDS and distributed networking telemetry experiments in general.

1.3 Outline

Outside of the introduction and conclusion, the manuscript is organized in three parts: defining FIDSs, quantifying their limitations, and providing solutions to address them.

Part I: The first part delves into the application of FL to IDS. After layout out the necessary background in Chapter 2, we present the state of the art in FIDS in Chapter 3. This chapter notably presents the results of our SLR on the topic, and focus on the related challenges and research opportunities. **??** then closes this first part by highlighting the main challenges in FIDS using toy examples.

Part II: The second part presents our contributions to quantifying the limitations of FIDS. Chapter 5 introduces a practical method to generate network topologies based on the composition of sub-topologies, and lays down the foundations for further studies on distributed networking analyses. Finally, Chapter 6 introduces our evaluation framework, and systematically analyses the impact of label-flipping attacks on FIDS.

Part III: The last part focuses on providing solutions to the challenges studied in Part II. Notably, Chapter 7 introduces a novel FL architecture for FIDS that handles malicious contributions in heterogeneous environments. Chapter 8 then makes a statement on the future of FIDS, with discussions about open issues and potential research directions.

1.4 Publications

Journal articles

[Lav+22b] Léo Lavaur, Marc-Oliver Pahl, *et al.*, « The Evolution of Federated Learning-based Intrusion Detection and Mitigation: A Survey », *in*: *IEEE Transactions on Network and Service Management*, Special Issue on Network Security Management (June 2022).

International conference papers

[Lav+24] Léo Lavaur, Pierre-Marie Lechevalier, Yann Busnel, Romaric Ludinard, *et al.*, « RADAR: Model Quality Assessment for Reputation-aware Collaborative Federated Learning », *in*: *Proceedings of the 43rd International Symposium on Reliable Distributed Systems (SRDS)*, Sept. 2024.

[LBA24a] Léo Lavaur, Yann Busnel, and Fabien Autrel, « Systematic Analysis of Label-flipping Attacks against Federated Learning in Collaborative Intrusion Detection Systems », *in*: *Proceedings of the 19th International Conference on Availability, Reliability and Security (ARES), Workshop on Behavioral Authentication for System Security*, Aug. 2024.

[LBA24b] 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)*, July 2024.

National conference papers

[Lav+23] Léo Lavaur, Pierre-Marie Lechevalier, Yann Busnel, Marc-Oliver Pahl, *et al.*, « Metrics and Strategies for Adversarial Mitigation in Federated Learning-based Intrusion Detection », *in*: *Rendez-vous de la Recherche et de l'Enseignement de la Sécurité des Systèmes d'Information (RESSI)*, May 2023.

[Lav+22a] Leo Lavaur, Benjamin Coste, *et al.*, « Federated Learning as Enabler for Collaborative Security between Not Fully-Trusting Distributed Parties », *in*: *Proceedings of the 29th Computer & Electronics Security Application Rendezvous (C&ESAR), Ensuring Trust in a Decentralized World*, Oct. 2022.

[Lav+21] Leo Lavaur, Marc-Oliver Pahl, *et al.*, « Federated Security Approaches for IT and OT », *in*: *Journée thématique du GT sur la Sécurité des Systèmes, Logiciels et Réseaux (GT-SSLR)*, May 2021.

Tutorials

[BL24] Yann Busnel and Léo Lavaur, « Tutorial: Federated Learning × Security for Network Monitoring », *in*: *Proceedings of the 44th International Conference on Distributed Computing Systems (ICDCS)*, July 2024.

[BL23] Yann Busnel and Léo Lavaur, « Federated Learning × Security for Network Management », 15th International Conference on Network of the Future (NoF), Sept. 2023.

Part I

Federated Learning to build CIDSs

Chapter 2

Background and Preliminaries

Contents

2.1 Introduction

This chapter provides the necessary background and preliminaries to understand the rest of the thesis. We start by laying out the basics of Machine Learning (ML) for intrusion detection in Section 2.2, followed by the implications of scaling up to Collaborative Intrusion Detection Systems (CIDSs) in Section 2.3, enumerating along the way the challenges that motivate the use of Federated Learning (FL). We then introduce the fundamentals of FL in Section 2.4, focusing on the FedAvg algorithm and the notations used throughout the thesis. Finally, we discuss the threats against FL in Section 2.4.4, with a particular focus on data poisoning attacks.

2.2 Intrusion Detection

Organizations long relied on signature-based Intrusion Detection Systems (IDSs) to detect intrusions. These systems leverage a database of known attack patterns (*i.e.*, *signatures*) to identify malicious activities. Listing 2.1 displays an example of a signature for detecting the Heartbleed vulnerability using Suricata, a popular open-source IDS. This signature relies on dedicated code inside Suricata's engine that required extensive human intervention to develop. It is consequently specific to Suricata. Such limitations motivated the study of ML for automatically extracting patterns from data, enabling the development of more flexible and adaptive IDSs.

IDSs can be broadly classified into two categories: *misuse detection* and *anomaly detection*. Misuse detection refers to the identification of known attack patterns. Signature-based IDSs fall into this category. On the other hand, anomaly detection compares a

1. https://github.com/OISF/suricata/blob/master/rules/tls-events.rules

```
alert tls any any -> any any (msg:"SURICATA TLS overflow heartbeat
    encountered, possible exploit attempt (heartbleed)"; flow:established;
    app-layer-event:tls.overflow_heartbeat_message; flowint:tls.anomaly.
    count,+,1; classtype:protocol-command-decode; reference:cve,2014-0160;
    sid:2230012; rev:1;)
```

Listing 2.1 – Example of a Suricata signature for detecting the Heartbleed vulnerability [1].

normal profile (trained on nominal traffic) with observed events to determine if they are malicious [Gar+09]. This approach is *de facto* more efficient for detecting novel attacks, but it is also more prone to false positives.

An analogy can be drawn between this classification and the two main paradigms of ML: supervised and unsupervised learning. In supervised learning, the model is trained on labeled data, where each sample is associated with a label. Labels can be classes (binary or multi-class classification) or continuous values (regression). Either way, the model's objective is to predict the label of unseen samples. In unsupervised learning, no labels are provided. The model's goal is to find patterns in the data, such as clusters or outliers. In anomaly detection, the model is trained on normal data only, and its objective is to detect deviations from this normal profile.

Multiple ML algorithms have been applied to intrusion detection, including Support Vector Machines (SVMs), Random Forests (RFs), and Artificial Neural Networks (ANNs). However, the rise of Deep Learning (DL) has led to a significant improvement in the performance of IDSs. DL is a subfield of ML that focuses on learning representations of data through the use of neural networks. In the following sections, we introduce the basics of DL for intrusion detection, review the existing paradigms, and discuss the metrics and datasets used to evaluate these systems.

2.2.1 Deep Learning for Intrusion Detection

DL present several advantages over traditional ML algorithms. Most notably, they automatically learn features from the data, reducing the need for manual feature engineering. This is particularly useful in the context of intrusion detection, where the features are often complex, interdependent, and of unequal relevance. The training data can range from network traffic to system logs depending on the type of mechanism used: network-based, host-based, or hybrid. Yet, Network-based Intrusion Detection Systems (NIDSs) greatly outnumber other approaches in the literature, due the availability of network traffic datasets and the ease of deployment.

Most of the research on NIDS use a representation known as *unidirectional network flows* or *netflows*, where a flow is defined as a sequence of packets sharing the same source and destination addresses, ports, and protocol. Various features can be extracted from

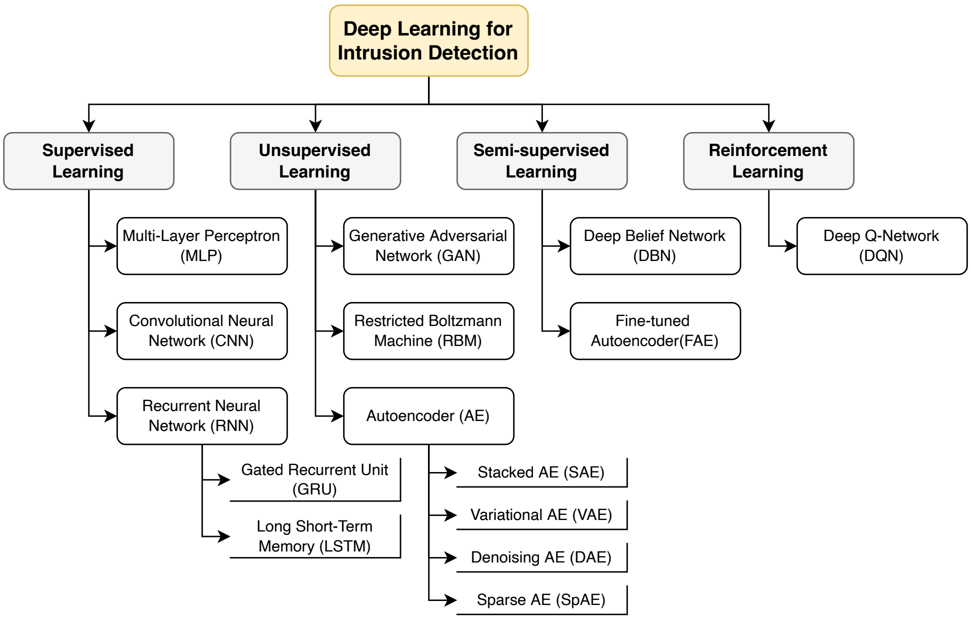


Figure 2.1 – Taxonomy of the main DL paradigms.

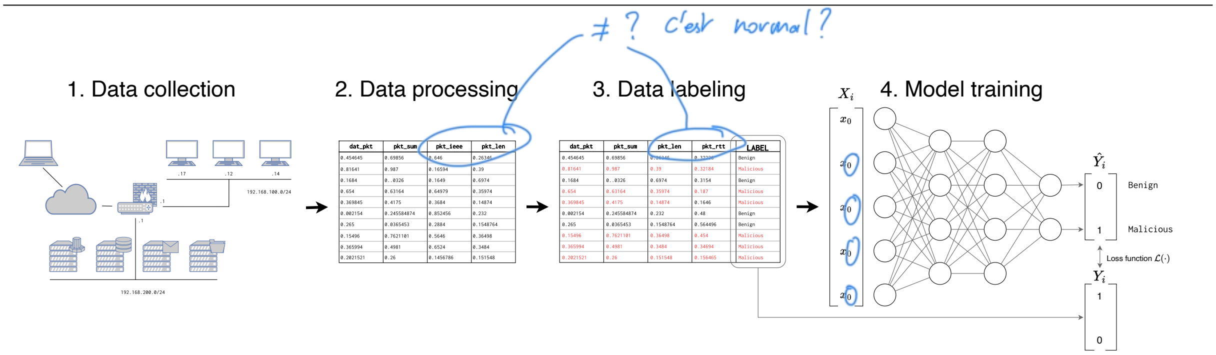
these flows, such as the number of packets, bytes, and the duration of the flow. More details on the features used in NIDS can be found in Section 2.2.2. More generally, the dataset D of size n is represented as a set of variables $X_i = \langle x_1, x_2, \ldots, x_m \rangle, i \in [\![1, n]\!]$, where x_j corresponds to the j-th feature, and m to the number of features.

Main DL Paradigms

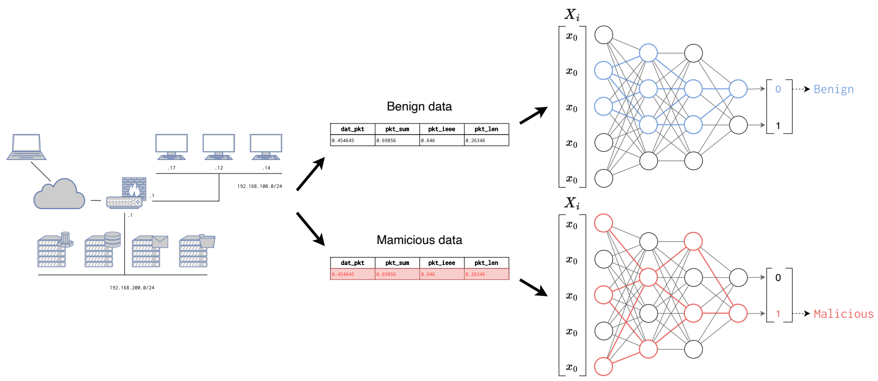
Because of their layered architecture, Deep Neural Networks (DNNs) can adopt different forms depending on the type of input data and the task at hand. Figure 2.1 presents the major families of DL algorithms: supervised-, unsupervised-, and semi-supervised learning, and finally reinforcement learning. While works exist on the application of reinforcement learning to intrusion detection [?], we focus on supervised and unsupervised learning in this thesis. This section provides an overview of these paradigms, and define for each the learning problem in the context of intrusion detection.

Supervised Learning Supervised learning is the most common approach in ML, and refers to the training of a model on labeled data. In the context of IDSs, practitioners usually seek to classify network flows into two classes (*benign* and *malicious*), which is a binary classification task. Consequently, the dataset D of size n associates each sample X_i with a label $y_i \in \{0, 1\}$. The model is trained to predict the label $\hat{y}$ of unseen samples. To do so, we generally use a Stochastic Gradient Descent (SGD)-based optimizer to minimize a loss function

$$\mathcal{L}(w, X_i, y_i), i \in [\![1, n]\!]. \tag{2.1}$$



(a) Training phase.



(b) Prediction phase.

Figure 2.2 – Workflow of a Multilayer Perceptron (MLP) for intrusion detection.

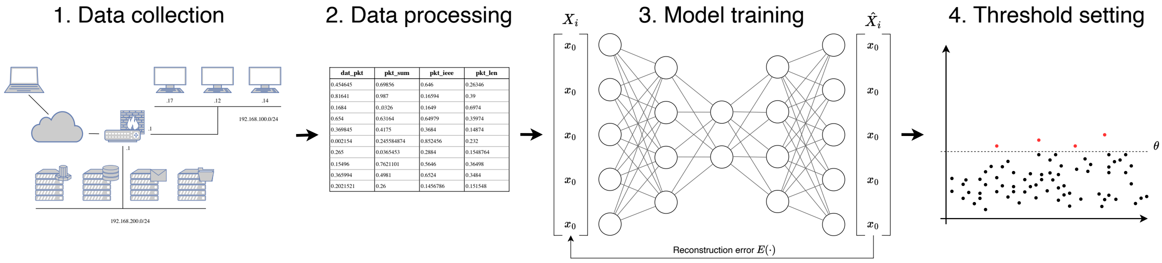
After computing the gradients $\nabla\mathcal{L}(w, X_i, y_i)$, they can update their model as

$$w^{t+1} \leftarrow w^t - \eta\nabla\mathcal{L}(w, X_i, y_i), \tag{2.2}$$

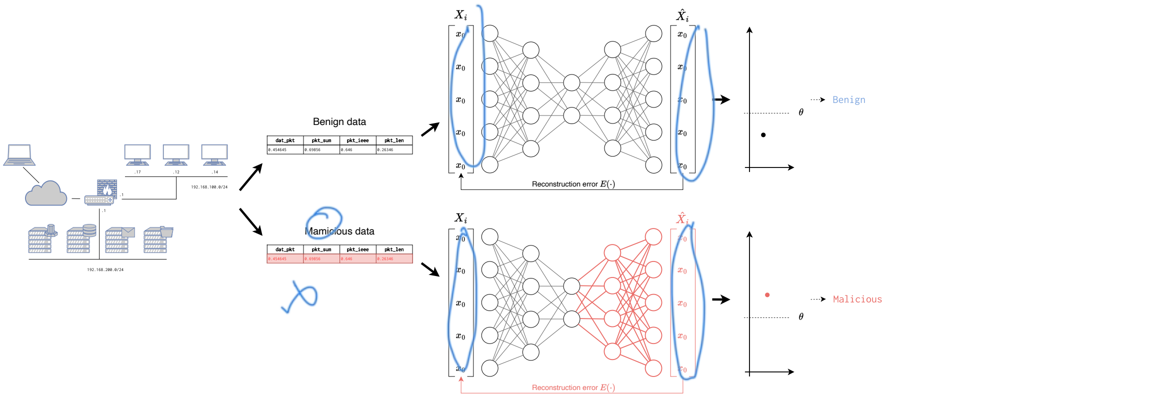
where η is the learning rate, and w the model's parameters, and t the iteration. The last layer usually uses `softmax` or `sigmoid` activation functions to output a probability of being in a class (normal or abnormal). In the case of multi-class classification, the label is one-hot encoded [2] into a vector Y of size c (the number of classes), and the `softmax` function is used. Depending on the available features and the learning objective, various architectures can be used, such as Convolutional Neural Networks (CNNs) for high-dimensional data, or Recurrent Neural Networks (RNNs) for sequential data. Multilayer Perceptrons (MLPs) are the simplest and most common architecture used in IDS, and the one that we focus on in this thesis, although most concepts can be extended to other architectures.

One of the main challenges in supervised learning is the availability of labeled data and its quality. In the context of IDS, obtaining enough labeled data is particularly challenging, as labeling requires expert knowledge and is time-consuming. Moreover, the class distribution is often unbalanced, with benign traffic being much more frequent than anomalies in the testing set [CBK09]. This issue is aggravated in siloed configurations, *i.e.*, in which

2. One-hot encoding is a binary representation of categorical variables, where each category is mapped to a binary vector. It is typically used in ML to represent categorical data, such as the protocol type in netflows.



(a) Training phase.



(b) Prediction phase.

Figure 2.3 – Workflow of a Stacked Autoencoder (SAE) for intrusion detection.

models can only be trained on locally-collected data. This can lead to models that are skewed by the unbalanced class distribution [Cam+22].

Challenge 1. Locally collected data is often unbalanced, leading to models that are skewed by the class distribution.

Redite d'une ligne sur l'autre.
Paraphrase la précédente ou retire-la.

Unsupervised Learning To circumvent the need for labeled data, unsupervised learning can be used. Unsupervised ML algorithms are typically used for clustering or outlier detection. The DL variants are rather used for feature extraction and dimensionality reduction, or anomaly detection. To detect anomalies, Autoencoders (AEs) can be trained on normal data only, and then used to see whether the reconstruction error of a new sample is above a certain threshold. This builds on the assumption that (i) benign traffic is much more frequent that anomalies in the testing set [CBK09]; and (ii) abnormal packets are statistically different from normal ones. In this scenario, the training dataset D is composed of benign (*i.e.*, normal) samples only and no associated label. Given $X = \{X_i | i \in [\![1, n]\!]\}$, the model is trained to minimize the reconstruction error $\mathcal{L}(X, \hat{X})$, where $\hat{X}$ is the output of the AE. A typical error function for this task is the Mean Squared Error (MSE), expressed as:

Si tu veux, tu peux aussi l'écrire $X = \{X_i\}_{i \in [\![1,n]\!]}$

C'est un ensemble ordonné non? Auquel cas, c'est plutôt $[\cdot]$ (voire $(\cdot)$) la notation.

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} \left(X_i - \hat{X}_i \right)^2 . \tag{2.3}$$

Then the model can be updated using the same process described in Equation (2.2). Different architectures of AEs can be used, such as Stacked Autoencoders (SAEs) to improve the quality of the extracted features, or Denoising Autoencoders (DAEs) to improve the robustness of the model. To detect anomalies, the reconstruction error of a new sample is compared to a threshold τ defined during the training phase on validation data. A high reconstruction error indicates that the considered samples is too *far* from the training data, and can indicate an anomaly. The performance of the model and its threshold can then be evaluated using a labelled test set.

Si tu ne les décris pas en détail, donne une réf. pour que le lecteur puisse se renseigner si intérêt / besoin.

Le seuil n'est pas "évalué" mais paramétré.

C'est la valeur du threshold qui est évaluée en fonction du DS spécifique

While unsupervised learning is particularly useful for detecting novel attacks, it is also more prone to misclassification. Local data in the real-world is likely to be collected on devices with little variance, *e.g.* same brand, same protocols, or use cases. This can lead to a normal profile that is too specific to the local environment, and thus would raise alerts as soon as a change occurs [LL19].

Challenge 2. Local data is specific to the environment, increasing the risk of false positives when changes occur.

Semi-supervised Learning Semi-supervised learning is a hybrid approach where only a small part of the training data is labeled. This approach is particularly useful in the context of IDS, where labeled data is scarce. A common strategy is to train an AE on the full dataset to learn the optimal representation of the data, and use the encoder (see Figure 2.3) part with a classifier to predict the label of the samples [APB20]. Other known model architectures for semi-supervised learning include Deep Belief Networks (DBNs), where multiple layers of Restricted Boltzmann Machines (RBMs) are stacked to form a deep network that extracts features from the data. The model is then fine-tuned using the labeled data for classification purposes.

2.2.2 Datasets

Datasets are essential in intrusion detection, as they allow researchers to evaluate and compare their solutions. This is even more critical when leveraging ML and DL techniques, as the performance of these models is highly dependent on the quality and quantity of the training data. Until the mid-2010s, the most common dataset used for intrusion detection was the KDD'99 dataset [Sig99], built for the KDD Cup 1999 competition using the DARPA 1998 dataset. Tavallaee *et al.* [Tav+09] published an updated version of the dataset, called NSL-KDD, which removes duplicates and corrects some errors in the original dataset. However, NSL-KDD is still based on the original DARPA 1998 dataset, and is considered outdated by today's standards.

Since 2015 with the publication of the UNSW-NB15 dataset [MS15], new datasets have been developed to address the limitations of the KDD'99 and NSL-KDD datasets,

Table 2.1 – Most common feature-based datasets for NIDSs.

Dataset	Year	Use case	Feature extraction	Features	Records (train/test) [a]	Attack classes	Reference
KDD Cup 99	1999	Military IT Network	Bro-IDS	41	4,898,431/311,029	4	[Sig99]
NSL-KDD	2009	Military IT Network	See KDD'99	41	125,973/22,544	4	[Tav+09]
UNSW-NB15	2015	Company IT Network	Bro, Argus, & Custom	49	2,540,044	10	[MS15]
CIDDS-001	2017	Small Business	NetFlow v9	10	31,959,175	5	[Rin+17a]
CIDDS-002	2017	Small Business	NetFlow v9	10	16,161,183	5	[Rin+17b]
CICIDS2017	2017	Company IT Network	CICFlowMeter	80	2,830,743	9	[SHG18]
CICIDS2018	2018	Large-scale IT Network	CICFlowMeter	80	8,284,254	7	[SHG18]
Bot-IoT	2019	Botnets and IoT	Argus & Custom	14	72,000,000+	4	[Kor+19]
ToN_IoT	2021	Cross-layer Infrastructure	Zeek & Custom	44	461,043	9	[Mou21]
Edge-IIoTset	2022	Cross-layer Infrastructure	Zeek & TShark	61	181,156/30,440	15	[Fer+22]

?

a. Some datasets do not have a recommended train/test split. In such cases, only the total number of records is provided.

such as the lack of realism[3] of the generated traffic, the lack of attack diversity, and the scale of the experiments. Table 2.1 presents the most common feature-based datasets for NIDSs, along with their characteristics. Two teams have been particularly active in this area: the Intelligent Security Group (ISG) [MS15; Kor+19; Mou21] at the University of New South Wales, Australia, and the Canadian Institute for Cybersecurity (CIC) [SHG18] at the University of New Brunswick, Canada. They brought the most used datasets in the field in recent years, UNWS-NB15 and CICIDS2017, respectively.

Provided features Because most of the datasets presented in Table 2.1 are made to train and evaluate MLs models, they rely on a set of features extracted from the network traffic. Some also include the original network captures (PCAPs) for further analysis, or complementary system logs for correlation purposes. Two non-exclusive approaches can be used to produce these features: feature extraction and feature selection.

Ajouter une tabulation.

Feature extraction refers to the computation of numerical characteristics after the data collection; *e.g.*, Inter-Arrival Time (IAT) or number of packets per device in the context of traffic monitoring. Most modern dataset use existing IDSs to extract these features, such as Zeek[4] or Argus[5]. The resulting data are network flows, aggregating the information of multiple packets into a single record.

Feature selection relates to the selection of the relevant features for a given task. This is particularly useful in the context of ML, where irrelevant or redundant features can degrade the performance of the model. For instance, Edge-IIoTset [Fer+22] contains 61 features, selected from a pool of 1176 based on feature correlation.

The choice of features is critical for the performance of the model, although DL models make this process less relevant due to their ability to filter out irrelevant features.

3. Only in regard to modern networks. Indeed, the DARPA 1998 dataset simulates multiple workstations in a military environment, using the US Air Force Research Laboratory's testbed. The technologies deployed where representative of the state of the art at the time.
4. Formerly known as Bro, available at: https://www.zeek.org/
5. Available at: https://openargus.org/argus-ids

Tu ne le cites pas dans le tableau 2.1 ?

		Predicted	
		Positive	**Negative**
Actual	**Positive**	True Positives (TPs)	False Negatives (FNs)
	Negative	False Positives (FPs)	True Negatives (TNs)

Table 2.2 – Confusion matrix for binary classification.

Yet, because each dataset comes with its own set of features, it is difficult to compare the performance of models across datasets. Recently, Sarhan, Layeghy, and Portmann [SLP22] proposed a standardized feature set for intrusion detection based on NetFlow V9 [Cla04] format. They used nProbe [6] to convert four known IDS datasets to this format: *i.e.*, UNSW-NB15 [MS15], Bot-IoT [Kor+19], ToN_IoT [Mou21], and CSE-CIC-IDS2018 [SHG18]. The NF-V2 datasets contain 43 features extracted from flow characteristics, such as duration or packet length, and some others that are protocols-specific. The uniform feature set across datasets allows the evaluation of ML models across independently generated datasets.

Use cases Until 2017 included, most datasets aim at simulating a *typical* network environment, such as deployed in an organization. This is the case for KDD'99, NSL-KDD, UNSW-NB15, and CIDDS 1 and 2, and CICIDS2017. Since, the focus progressively shifts towards more specific use cases, notably with the generalization of Internet of Things (IoT) devices. These datasets include protocols that are not present in traditional IT-oriented networks, such as MQTT or CoAP. This is the case for Bot-IoT [Kor+19], ToN_IoT [Mou21], and Edge-IIoTset [Fer+22].

Tu ne précises pas (encore) lesquels tu utiliseras ?

2.2.3 Metrics

Most research on ML for intrusion detection relies on the same set of metrics to assess, validate, and compare their solutions [Cha+19; Far+20; BG16]. Most of these metrics are derived from the confusion matrix (see Table 2.2), which is a table that summarizes the performance of a classification model along the different classes. To compute the confusion matrix, the model's predictions ($\hat{y}$) are compared to the true labels (y, the ground truth) of the samples. All the metrics presented in this section are defined for binary classification, but can be extended to multi-class classification [BG16].

(1) *Accuracy* represents the proportion of correctly classified items. It is the ability for the system to correctly distinguish abnormal traffic from legitimate one.

$$Accuracy = \frac{TP + TN}{P + N}$$

6. Available at: https://www.ntop.org/products/traffic-analysis/nprobe/

(Pas relu car déjà commenté :)

(2) *Precision*, or Positive Predictive Value (PPV), is the proportion of correct positive cases among all the cases that have been categorized as positive.

$$Precision = \frac{TP}{TP + FP}$$

(3) *Recall*, or True Positives Rate (TPR) represents the proportion of true positive cases that have been correctly categorized.

$$Recall = \frac{TP}{P} = \frac{TP}{TP + FN}$$

(4) *Specificity*, or True Negative Rate (TNR), is the proportion of negative cases that has been correctly categorized.

$$Specificity = \frac{TN}{P} = \frac{TN}{TN + FP}$$

(5) *Fallout*, or False Positives Rate (FPR), represents the proportion of the positive cases that should have been categorized as negative. A high FPR often requires human intervention after the classification task to filter out the false positive.

$$Fallout = \frac{FP}{N} = \frac{FP}{FP + TN}$$

(6) *Miss rate*, or False Negative Rate (FNR), relates to the proportion of positive cases that have not been categorized as such. In the context of IDSs, it represents an attack that has been missed by the system. Thus, it is a critical metric for this use case.

$$Miss\ rate = \frac{FN}{P} = \frac{FN}{FN + TP}$$

(7) *F1-Score* is the harmonic mean of precision and recall. It is often used to measure ML algorithm, but is also criticized because of the equal importance it gives to both precision and recall [HC18].

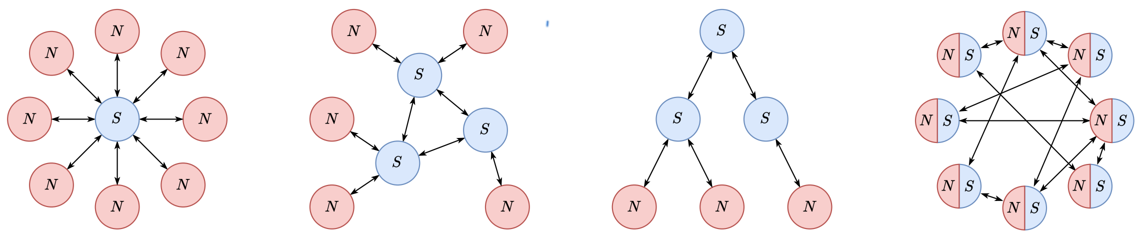
$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

(8) *Mathew Correlation Coefficient (MCC)* is an adaptation of the *Phi* (ϕ) coefficient to confusion matrices. While being mathematically identical, the term is often preferred by the ML community. The MCC has significant advantages over the other metrics, as it covers all four categories of the confusion matrix [CJ20]. Thus, a high score can only be obtained with high TP and TN, and low FP and FN.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Next page.

On a pas la même définition. Pour moi, decentraliser est plus fort que distribuer. Ton (d) représente un système P2P, qui est la decentralisation ultime. Un système cloud est considéré comme distribué alors que c'est clairement une approche type (b). Utilise plutôt "hybride" ?



(a) Centralized **(b)** Decentralized **(c)** Hierarchical **(d)** Distributed

Figure 2.4 – Different topologies for collaborative intrusion detection systems. Nodes are in red and marked as N, while servers are in blue and marked as S. Arrows represent connections between entities.

2.3 Collaboration in Intrusion Detection

The topic of collaboration in intrusion detection is rather old, with several surveys and reviews available in the literature [ZLK10; EO11; VKF15; Men+15; FS16; LMK22], and the oldest references dating back to the early 1990s [Sna+92]. In this section, we present the different types of collaboration in intrusion detection, and discuss some of the challenges that they face. A first distinction can be made between their objectives, ~~also~~ although they are not mutually exclusive: (i) *share results* to correlate alerts and detect attacks at a global scale, or (ii) *share knowledge* to improve the detection capabilities of local systems.

Bof

This distinction also transpires from the scale of the collaboration. The first case mostly refers to different probes, or sensors, that monitor the same infrastructure, and share their results to correlate alerts and detect attacks at the infrastructure level. In the second case, which is sometimes referred to as a Collaborative Intrusion Detection Network (CIDN) [LMK22], collaboration usually happens among different organizations or entities that monitor infrastructures. In this thesis, we will focus on the latter, as it is more relevant to the FL context (see Section 2.4).

2.3.1 The different topologies

The aforementioned literature identify three main types of topologies for CIDSs: centralized, decentralized, and distributed. The definitions of these topologies are not always consistent across the literature, especially between the terms *decentralized*, *hierarchical*, and *distributed*. For instance, Zhou, Leckie, and Karunasekera [ZLK10] consider a decentralized system as a system where each node is autonomous and can make decisions independently, while this definition matches the description of a distributed architecture in the work of Li, Meng, and Kwok [LMK22].

In this manuscript, we will use the definitions illustrated in Figure 2.4. It distinguishes two types of roles: *nodes* (N) and *servers* (S). A node is a device that captures data, Pas seulement, ou plutôt pas nécessairement. Il peut collecter des aggregats (ou des poids comme en FL)

although it can also execute complementary tasks like preprocessing, feature extraction, or traffic analysis. A server is a device that aggregates the data from the nodes and distributes instructions to them, as well as updates for the ~~local~~ detection algorithm. The different topologies are defined as follows:

Centralized In a centralized architecture, a single server centralizes knowledge and distributes instructions to the nodes.

C'est exactement un système distribué hybride

Decentralized In a decentralized architecture, multiple servers coexist. Each server is responsible for a subset of the nodes, and they can share information between them.

Hierarchical A hierarchical architecture is decentralized system where the servers are organized in a tree-like structure. ~~Each~~ Some server ~~is~~ are responsibles for a subset of the nodes, and they can forward information to their parent. Likewise, parents can distribute instructions and updates to their children so that they are disseminated throughout the hierarchy.

Fully decentralized si tu préfères.

Distributed In a ~~distributed~~ architecture, each node is autonomous and can make decisions independently. Both roles of nodes and servers coexist in the same entity. There are no servers anymore, and the node share information over a peer-to-peer network.

Figure 2.4 illustrates these topologies. In the centralized architecture (Figure 2.4a), all nodes are connected to a single server. Figure 2.4b shows a standard example of a decentralized architecture. Figures 2.4c and 2.4d illustrate the specific cases of decentralized architectures: hierarchical and distributed, respectively. The arrows between the different entities represent information exchange, although the nature of these exchanges can vary depending on the direction. An arrow displayed as $N \rightarrow S$ can represent collected data, generated alerts, or requests for updates. An arrow displayed as $S \rightarrow N$ can represent instructions or updates for the local detection algorithm or database.

On comprend que c'est Node et Server mais tu n'as pas vraiment introduit cette notation (sauf dans la légende bien entendu)

2.3.2 Challenges in Collaborative Intrusion Detection

Collaborative intrusion detection faces challenges, including the Single Point-of-Failure (SPoF) in a centralized architecture. If the analysis is performed remotely, like in a Security Operations Center (SOC) monitoring its consumers' infrastructures, a failure on the central server would hinder detection. Fortunately, in knowledge-sharing scenarios, detection is (at least partially) performed locally, reducing the impact of a centralized failure. Nonetheless, collaboration still relies on the availability of the central server.

Challenge 3. CIDSs typically rely on a central server for coordination and updates, which represent a Single Point-of-Failure (SPoF).

Another challenge in collaborative intrusion detection is the latency induced by propagating information over the network, especially under load. The ENISA, the European

Union Agency for Cybersecurity, defines the actionability of Threat Intelligence (TI) as the fulfillment of five criteria: relevance, digestibility, accuracy, completeness, and timeliness [ENI14]. It is the supporting architecture that provides the latter. Because low-latency is crucial for actionable alerts locally, centrally analyzing the data increases the time between the event and its detection.

Challenge 4. Centralized detection increases latency, which makes the shared knowledge less actionable.

Further, sharing data can represent a privacy risk for a company, as the data relevant for intrusion detection is likely to contain sensitive information [ZLK10]. Exposed information might reveal relevant insights to a competitor or an attacker.

Challenge 5. CIDSs can expose sensitive information about the internals of a company.

A lot of other factors can impede collaboration. For instance, stakeholders are often reluctant to share their information, fearing confidentiality and privacy issues (see Challenge 5), but most importantly the reputation loss that could result from a breach [PZ19]. Cultural and language barriers can negatively affect the accuracy of the shared information, even though international collaboration is push by regulation, such as the NIS directives in Europe [16; 22]. Finally, the balance between anonymity and trust must be taken into consideration to protect the participants without sacrificing the quality of the information [ML15].

2.3.3 CIDS with Machine Learning

Before the advent of FL, the literature on CIDSs leveraging ML, or more generally data-mining techniques, was scarce [FS16]. Existing solutions were mostly based sharing alerts for correlation or rules for misuse detection, or rely on a central server to perform learning tasks. Nonetheless, a few works leveraged distributed learning techniques that remind of FL, notably with the constraint of not being able to share data between the nodes. For instance, Folino, Pizzuti, and Spezzano [FPS10] proposed a framework allowing distributed IDS nodes to train and exchange classifiers, before aggregating to build ensemble models. However, most of the aforementioned reviews still identify data-mining and ML techniques as a promising direction for CIDSs.

2.4 Fundamentals of Federated Learning

Introduced in 2016 by McMahan *et al.* [McM+17], Federated Learning (FL) changes the usual ML paradigm where distributed data is centrally collected, curated, and processed on a dedicated server. Instead, FL respects the decentralized nature of the data and rather brings model training to the data sources. By alternating between training on local data and aggregating model updates, FL enables the training of a shared model without the need to share the data itself. This approach reveals itself as a promising solution to multiple challenges faced by traditional ML systems, the two main ones ~~being~~: are

1. Training models over massively distributed data sources, such as smartphones, wearables, or IoT devices; and
2. Training models on sensitive data, such as medical records or financial transactions, while preserving privacy and confidentiality.

Although the term *Federated Learning* was introduced by McMahan *et al.* [McM+17] to describe their approach focusing on distributed mobile devices, the literature has significantly broadened the definition of FL to encompass a wide range of privacy-preserving distributed learning techniques. Therefore, we prefer the definition introduced in 2021 by Kairouz *et al.* [Kai+21] and reiterated in Definition 2.1. The following sections introduce the fundamentals of FL, with a focus on the `FedAvg` algorithm, the different types of FL, and the question of data distribution. Finally, we succinctly discuss the threats against FL, with a focus on data poisoning attacks. Table 2.3 summarizes the notations used in this section, and throughout the manuscript.

Definition 2.1: Federated Learning

Federated Learning is a machine learning setting where multiple entities (clients) collaborate in solving a machine learning problem, under the coordination of a central server or service provider. Each clients raw data is stored locally and not exchanged or transferred; instead, focused updates intended for immediate aggregation are used to achieve the learning objective. – Kairouz *et al.* [Kai+21]

Formally, FL seeks to minimize a objective function $f(\cdot)$ [7], as:

$$\min_{w} f(w), \quad \text{where} \quad f(w) = \sum_{k=1}^{K} \rho_k f_k(w), \tag{2.4}$$

where w is the model parameters, $f_k(w)$ is the local objective function of client k, and ρ_k is the weight of client k.

2.4.1 The `FedAvg` Algorithm

`FedAvg` is the first and most popular algorithm for FL. The algorithm operates in rounds, noted r. At each round r, an orchestrating server S randomly selects $C \cdot K$ clients

7. Note that in the context of intrusion detection using ML, the objective function $f(\cdot)$ is often a loss function $\mathcal{L}(\cdot)$ to minimize, such as mentioned in Section 2.2.1.

Table 2.3 – Summary of Notations.

Pas possible d'avoir des doublons ici. Mais en général ?

Notation	**Description**
K	Number of participants
$P = \{p_k \mid k \in [\![1, K]\!]\}$	Set of all participants
d_k	Local dataset of participant p_k
$D = \bigcup_{k=1}^{K} d_k$	Union of all local datasets
$X_i = \langle x_1, x_2, \ldots, x_m \rangle$	Features of sample i
Y_i	Label enconding for sample i
w_k^r	Local model of the k-ith participant at round r
$W = (w_k^r \mid k \in [\![1, K]\!])$	Local models from participants at round r
$\bar{w}$	Aggregated model at round r

C Fraction of clients selected.

from a pool of participants P, with K being the total number of participants and C the fraction of clients selected with $0 < C \leq 1$. The server then tasks each selected participant $p_k, k \in [\![1, K]\!]$ to train a model w_k^r. The round ends by the aggregation of the collected models into a new global model $\bar{w}^r$, which is redistributed to the clients as a starting point for the next round $(r + 1)$.

In essence, `FedAvg` is a distributed 2-level SGD algorithm, where C controls the *global* batch size, and then each client p_k uses a *local* batch of size β to compute a local update. To train their model, the participants use a SGD-based optimizer to minimize a objective function $f_k(w)$, which is the local loss function of client p_k (see Definition 2.1). They compute the gradients of the loss function with respect to the model parameters, as:

$$g_k^r = \nabla f_k(w_k^r), \tag{2.5}$$

with is the results of the local optimization process.

In their original publication, McMahan *et al.* [McM+17] introduce two algorithms for FL: `FedSGD` and `FedAvg`. `FedSGD` is a straightforward implementation of SGD in a federated setting, where each client computes the gradients after one epoch and sends them to the server. The server then aggregates the gradients and updates the global model. With $D = \bigcup_{k=1}^{K} d_k$ being the union of all local datasets d_k, the server computes the new global model as:

$$w^{r+1} = w^r - \eta \sum_{k=1}^{K} \frac{|d_k|}{|D|} g_k^r, \tag{2.6}$$

where η is the learning rate, and $|d_k|$ and $|D|$ are the sizes of the local dataset and the global dataset, respectively.

Based on the observation that there is no difference between averaging the gradients g_k^r and updating the global model, or updating the model locally and then averaging

Algorithm 2.1 `FedAvg` [McM+17]. The participants of $|P|$ are indexed by p, C is the fraction of participants to be selected at each round, β the local batch size, η the learning rate, $\mathcal{E}$ the number of epochs, and $\nabla\mathcal{L}$ the gradients of the loss function $\mathcal{L}$.

Initialize w_0
for each round $r = 1, 2, \ldots$ **do**
    $m \leftarrow \text{MAX}(C \cdot K, 1)$
    $P^r \leftarrow (\text{SELECTRANDOM}(m, P))$
    for all $p_k \in P^r$ **do**
        $w_k^r \leftarrow \text{CLIENTFIT}(p_k, w^r)$
    $w^{r+1} \leftarrow \sum_{k=1}^{K} \frac{|d_k|}{|D|} w_k^r$
▷ *On client p.* ◁
function $\text{CLIENTFIT}(p, \omega)$
    for $i \leftarrow 1, \ldots, \mathcal{E}$ **do**
        for all $b \in \text{SPLIT}(d_k, \beta)$ **do**
            $\omega \leftarrow \omega - \eta\nabla\mathcal{L}(\omega, b)$
    return ω

Non introduit (est pas self-contained à mon avis)

the results, McMahan *et al.* [McM+17] introduce `FedAvg`. Indeed, the two operations are equivalent, as the following equation shows:

$$w^{r+1} = w^r - \eta \sum_{k=1}^{K} \frac{|d_k|}{|D|} g_k^r = \sum_{k=1}^{K} \frac{|d_k|}{|D|} w_k^r, \tag{2.7}$$

where w_k^r is the model trained by client p_k at round r. This equivalence allows clients to train their models for multiple epochs before sending the results to the server, which reduces the communication overhead. Algorithm 2.1 summarizes the `FedAvg` algorithm.

This core idea, that *averaging locally trained models iteratively converges towards an optimal model trained over distributed data*, is the foundation of FL.

2.4.2 Types of Federated Learning

As stated in the introduction of this section, the literature has broadened the scope of FL, leading to different types of FL depending on the context and the objectives of the federation.

Cross-device *vs.* Cross-silo Federated Learning The first notable distinction is between Cross-Device Federated Learning (CD-FL), the context in which FL was introduced, and Cross-Silo Federated Learning (CS-FL). The *cross-device* settings concerns massively distributed devices which are typically low-power and resource-constrained, such as smart-

Je me demande si je préfère pas l'image "extraite" de Wikipedia:

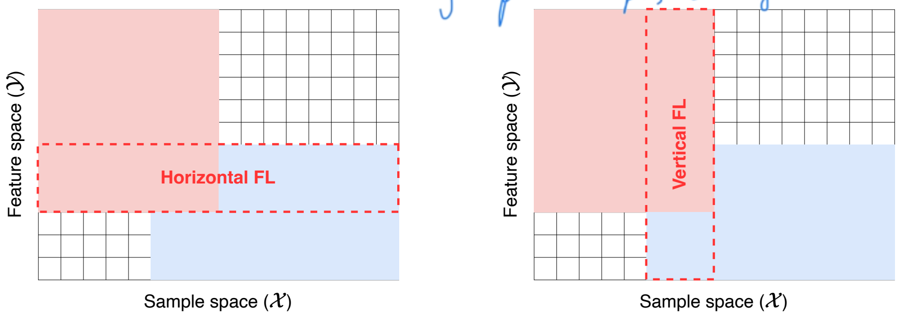


Figure 2.5 – Horizontal *vs.* Vertical Federated Learning. In horizontal FL, clients share the same features but not the same samples. In vertical FL, clients share the same samples but not the same features.

heterogenous and

phones, wearables, or IoT devices. Their number can range from thousands to billions, and they are often owned by different users. Consequently, CD-FL often encounters challenges related to limited availability, reliability, and communication overhead, but offers scalability and adaptability. In contrast, in *cross-silo* settings, FL operates within organizational boundaries or distinct data silos, where each silo represents a separate entity or institution. Silos could correspond to different departments within a company, independent organizations, or even geographical regions. CS-FL typically implies organizations with more homogeneous capabilities and more data to train on. Parties in cross-silo FL are more likely to be reliable and consistently available for participation, as they are usually institutional entities with dedicated infrastructure and resources. Yet, entities involved in CS-FL also tend to have considerably greater discrepancies in terms of objectives and data-distributions, and sometimes even model architectures.

Horizontal *vs.* Vertical FL Another major distinction in FL is on along which axis the data is distributed. In most application (notably in CD-FL), participants share the same features, but possess different samples. This is referred to as Horizontal Federated Learning (HFL) by Yang, Liu, *et al.* [Yan+19], and is illustrated in Figure 2.5. HFL particularly copes with *ground-truth* issues (Challenge 2) by providing more data for the global model to be trained on. Conversely, in Vertical Federated Learning (VFL), participants might have different views over the same data, *i.e.* they share the same samples but not the same features. This is particularly relevant in cross-silo applications, where different organizations might have access to different data sources, but observe the same events. Finally, Yang, Liu, *et al.* [Yan+19] also consider Federated Transfer Learning (FTL), where participants share only a subset of both, features and samples.

Bof.

Architecture Discrepancies Although FL's definition implies a client-server architecture, the literature has also explored other settings. In the *server-orchestrated* setting, the central server plays a pivotal role in distributing model parameters, orchestrating training

(Challenge 1)

rounds, and aggregating updates from individual devices. This approach requires global coordination and synchronization, as all communication and aggregation activities are orchestrated by the server. While server-orchestrated FL offers centralized control and streamlined management, it also introduces potential single points of failure and scalability limitations due to the server's central role. This is reminiscent of the centralized architecture mentioned in Section 2.3 and fig. 2.4a and comes with the same caveats. Researchers have explored multiple alternatives to the server-orchestrated setting, such as hierarchical FL [Liu+20] or fully decentralized FL using gossip algorithms [Tan+23]. This decentralized approach eliminates single points of failure and allows for greater scalability, as communication and computation can be distributed across numerous devices. However, fully decentralized FL may face challenges related to coordination, consistency, and synchronization, especially in scenarios with a vast number of participating devices.

2.4.3 The Question of Data Distribution

C'est marrant, ces paragraphes me rappelent quelque chose ;-)

The performance of FL algorithms is highly dependent on the distribution of the data across the participants. Almost by definition, data in FL settings is Non Independent and Identically Distributed (NIID), as it is distributed across different devices or organizations, with no guarantee of homogeneity. However, the performance of FL algorithms of the literature is often evaluated under the assumption that the data is Independent and Identically Distributed (IID), which is rarely the case in practice. This discrepancy between the theoretical assumptions and the practical reality poses a significant challenge for the FL community.

In the FL foundation paper [McM+17], the authors emphasize on NIID data being one of the key attributes of FL, alongside the unbalanced overall distribution. They notably present a *pathological*-NIID situation using MNIST [Lec+98], a digit recognition dataset, where each client is given only two digits, *e.g.* 3 and 7. More recent papers consider alternative NIID use cases, deemed more realistic. For instance, Huang *et al.* [Hua+21] present a *practical*-NIID use case, where participants can share similarities. This is particularly suited for cross-silo use cases, such as CIDSs. Indeed, we can easily expect different organizations to own different architectures, and yet observe similar traffic patterns in their networks.

The literature has addressed the issue of NIID data in FL from multiple angles. First, some algorithms have been specifically designed to handle NIID data, such as `FedProx` [Li+20c] or `SCAFFOLD` [Kar+20], although the former also covers the topic of heterogeneous capabilities. Techniques such as client-side sampling, in which clients sample their data to match the global distribution, have also been proposed [Han+24]. Finally, the literature has also explored clustering approaches to group client with model updates in communities, assuming that similar updates come from clients with similar data distributions [Ye+23].

2.4.4 Threats against Federated Learning

The distributed nature of FL opens the way to various attack vectors, which can be classified into two main categories: attacks against the federated model and attacks targeting the participants' privacy. In the former, adversaries aim to alter the behavior of the global model, either to degrade its performance or to manipulate specific predictions. In the latter, adversaries seek to infer sensitive information about the participants' data, such as inferring the presence of a specific sample in a participant's dataset.

Il n'y a pas de lien de cause à effet.

~~Because~~ the two categories have obviously different objectives, and consequently different threat models; we focus on the former in this thesis. Authors often refer to poisoning attacks in FL as *Byzantine* attacks, as they are analogous to the concept of Byzantine faults in distributed systems. Likewise, the term *Sybil attacks* [Dou02] is frequently used to refer to the problem of *colluding attackers* [FYB20].

Attack vectors Poisoning attacks can be categorized into two main categories depending on the phase in which they are perpetrated: model-poisoning [Bha+19] or data-poisoning [Tol+20]. Model-poisoning attacks aim at manipulating the model's parameters, usually during or after training, to deviate the aggregated model from the global optimum [Fan+20b]. Data-poisoning attacks, on the other hand, happen before the training phase, and manipulate data samples to degrade performance, cause misclassification, or introduce backdoors [Rod+23].

Data poisoning attacks can be categorized into clean-label and label-flipping attacks. Clean-label attacks manipulate the samples to be misclassified, either by adding new samples [Zha+22] or by modifying existing ones [Mer+23]. Label-flipping attacks, on the other hand, change the labels of the samples by flipping them to a different class [Tol+20]: *i.e.*, $y_{\text{src}} \rightarrow y_{\text{target}}$.

↳ tu peux écrire "source" en entier?

Attack target Additionally, most poisoning attacks can be further separated into *untargeted* and *targeted* attacks. Untargeted attacks randomly select samples to be manipulated, and are usually easier to detect as they have a higher impact on the model's performance. Targeted attacks, on the other hand, select samples based on a specific criterion, such as the class to be targeted. In a CIDS context, targeted attacks can be used to introduce backdoors – *i.e.*, making a specific attack class be misclassified as benign – or cause targeted misclassification.

2.5 Conclusion and takeaways

In this chapter, we have introduced the basics of ML for intrusion detection and the challenges of scaling up to CIDSs. We have also introduced the fundamentals of FL its implications. With the challenges listed along the way, the relationship between FL and CIDS becomes straightforward: FL is a natural fit for CIDSs as it allows to leverage the benefits of distributed learning while preserving the locality of the data.

In the next chapter, we will review the state of the art in FL in the context of CIDS, focusing on the different approaches to the problem and the challenges they face. Notably, we will discuss **??** RQ1: *What makes applying FL to IDSs specific?*.

Chapter 3

State of the Art

Contents

3.1 Introduction and Motivation

In the previous chapter, we introduced the concepts of Intrusion Detection System (IDS) and Machine Learning (ML), the challenges of deploying Collaborative Intrusion Detection Systems (CIDSs), and why Federated Learning (FL) is a promising solution to these challenges. This chapter's prime objective is to provide a comprehensive review of how Federated Learning (FL) can be leveraged for intrusion detection purposes, and shed light on the gaps in the literature that are discussed in this thesis.

A recent topic without identity Because of the novelty of FL in the field of Intrusion Detection System (IDS), the literature on the topic is still scarce. Only a handful of reviews [Ala+21; Agr+22; Cam+22] had been published on the topic when we stopped our data collection for this study in late 2021. While these papers provide a good overview of the existing works, they fail to provide synthesis and extract the core characteristics of the field. Notably, *what makes FL for IDS different from FL for other applications, and what challenges are specific to the field of intrusion detection?*

A systematic approach We aim to address this gap as thoroughly and transparently as possible, and leverage the Systematic Literature Review (SLR) methodology to that end. This methodology [KC07] relies on a structured process to identify, select, and analyze the relevant literature on a given topic. With explicitly defined research questions and inclusion/exclusion criteria, the Systematic Literature Review (SLR) methodology ensures that the review is reproducible and unbiased. Therefore, we intend to provide a comprehensive overview of the existing literature, and reproducible, evidence-based conclusions on the specificities of FL for IDS.

Content The content of this chapter is based on our survey published in TNSM in May 2022 [Lav+22b] and its accompanying extension at the C&ESAR conference in November 2022 [Lav+22a]. Because the initial paper was submitted in November 2021, the quantitative analysis has been updated during the writing of this manuscript to include the latest publications on the topic. The qualitative analysis has also completed to a lesser extent.

Contributions of this chapter

- The first SLR on the use of FL for IDS, including qualitative and quantitative analyses of the literature.
- A generalization of the selected works as a reference architecture for Federated Intrusion Detection Systems (FIDSs), providing a starting point for future works on the topic.
- A taxonomy synthesizing the state of the art of FIDS, providing a framework to analyze and compare existing and upcoming literature.
- The identification of the main challenges and opportunities in the field, and a set of research directions to address them.

3.2 Methodology

This section details the methodology applied to review the state of the art of FIDSs. The original article follows the SLR methodology introduced to the engineering field by Kitchenham and Charters [KC07]. SLR uses analytical methods to answer research questions about the literature on a specific topic. The update to the original article is less structured and more focused on the evolution of the field, so the methodology is adapted accordingly.

3.2.1 Research Questions

The SLR methodology recommends defining explicit research questions to structure the review and the selection of papers. This survey aims at evaluating FIDS and their maturity, as well as their core components, and relevant variations. Therefore, using related and selected works, we identify the following Research Questions (RQs) that cover the topic of FIDSs. The questions complete and extend **??** RQ1 which was introduced in Chapter 1.

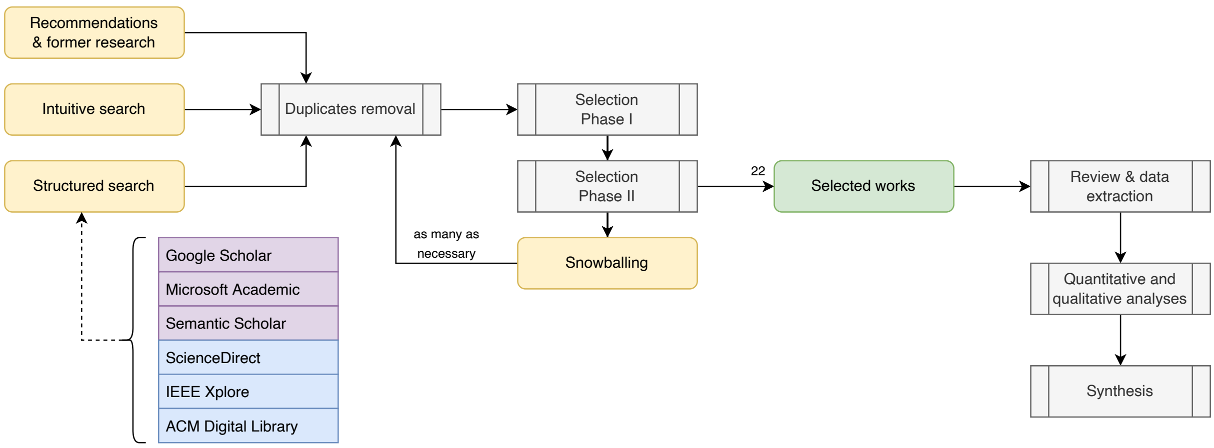


Figure 3.1 – *Search and selection processes.* Sources of papers appear in yellow, the final selection in green, and the processing steps in gray. The tools used in the *Structured search* are presented with search engines in purple, and online databases in blue. Figure from Lavaur, Pahl, *et al.* [Lav+22b] © IEEE 2022.

RQ1-1. *What are FIDS?*

- **1.a.** What challenges do FIDS help to cope with?
- **1.b.** Which techniques exist to federate Machine Learning (ML)–based detection and mitigation mechanisms?

RQ1-2. *What are the differences between FIDS?*

- **2.a.** What are the key components of FIDS? How do they influence the system's performance?
- **2.b.** Which metrics are used to measure and compare FIDS?

RQ1-3. *What is the state of the art of FIDS?*

- **3.a.** What are the topics covered by the academic literature since 2016?
- **3.b.** Where was the literature published? Which research groups and communities are active in this area?
- **3.c.** What are open questions according to existing works?

3.2.2 Search and Selection Process

Figure 3.1 presents the methodology and its search, selection, and synthesis processes. The searching of relevant literature involves four sources: recommendations, intuitive search, structured search, and snowballing.

(1) *Recommendations* were given by supervisors and coworkers throughout the realization of this work. This initial set of relevant papers is also used as a source of snowballing for further searching. Moreover, we included references from an aborted survey on *Collaborative security approaches*, which already yielded a substantial amount of literature by using the same methods.

(2) *Intuitive search* has been performed at the beginning of the survey to get a first grasp on the topic, and to learn about the functioning of FIDSs. At first, mostly Google Scholar has been used.

(3) *Structured search* has been adopted afterward, following the principles of SLR [KC07]. Different search engines and online databases are used for the sake of completeness, as illustrated in Figure 3.1. Databases can provide different results depending on their ownership. Search engine results differ according to the way requests are parsed, and the papers they have indexed. Thus, multiple sources provide more exhaustive results. (a) application of FL to IDSs, and (b) literature addressing the topic of FIDS with unusual keywords.

   (a) `("federated learning" OR "fl" OR "federated")`
   `AND ("intrusion detection systems" OR "ids")`

   (b) `("federated" OR "collaborative")`
   `AND ("detection" OR "defense" OR "mitigation")`

(4) *Snowballing* identifies relevant works that would have been missed otherwise, such as publications cited by articles of our selected corpus, or papers that refer to them. The related surveys identified in this work (Section 3.5) contain a lot of references to technical articles, making them relevant for snowballing. Furthermore, as this survey proceeds with quantitative analysis of the venues and groups (Section 3.3), it provides extended snowballing opportunities by looking at other publications in the most represented venues or research groups in the selected corpus.

Approximately two hundred papers have been identified. Duplicate removal is performed with Zotero which allows identifying and merging redundant items. The selection then happens in two phases. Firstly, the title and abstract are used to discriminate *out-of-scope* papers in Phase I, along with their number of citations given the search engines, and age. However, a paper with few citations, but interesting abstract, probably only lacks visibility. Thus, it is moved to Phase II, which consists of a more thorough analysis of the selected works, using the *three-pass* approach defined by Keshav [Kes07].

After the two selection phases, 22 papers were selected, excluding the 18 initial surveys seen in Section 3.5. All present technical solution for FIDS. The challenges identified in Chapter 2 were also used to either search or select papers, mostly through the *intuitive search* part.

3.2.3 Data Extraction and Analysis

The quantitative section of the original paper was solely based on the 22 selected papers. However, a significant amount of literature has been published since the initial survey. Therefore, we updated the quantitative analysis to include the latest publications on the topic. The qualitative analysis has also been completed to a lesser extent, just

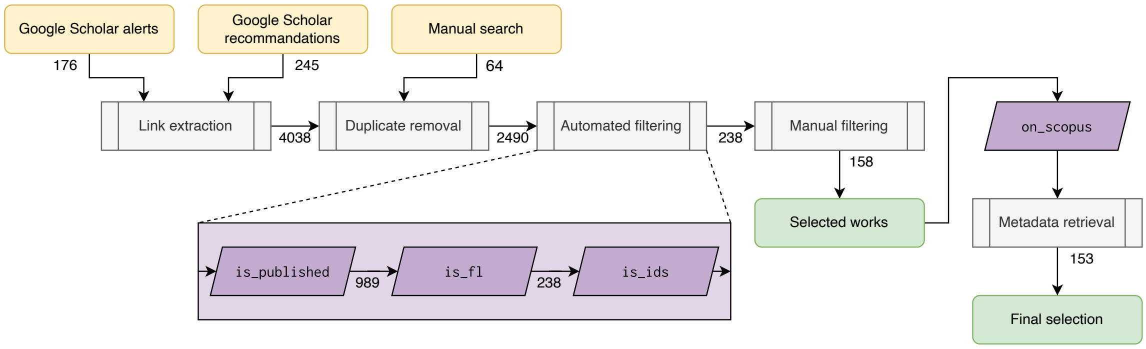


Figure 3.2 – *Updated selection process.* The sources of papers are in yellow, the selected papers in green, and the processing steps in gray. Purple parallelograms the automated filters used to select papers.

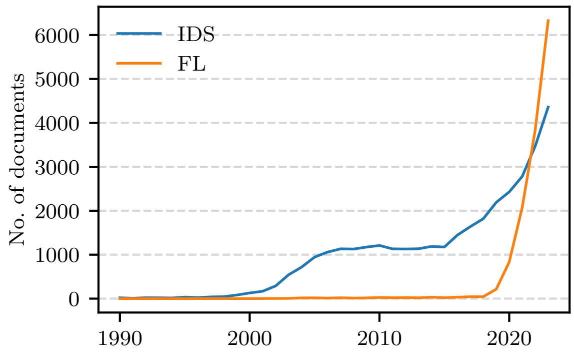
enough to provide a general overview of the field's evolution. Figure 3.2 presents the methodology applied to update the presented results.

We set up an automated collection system on Google Scholar, composed of an alert based on the search queries defined in Section 3.2.2 and automated recommendations. The system was set up in 2021 and ran until the beginning of the writing of this manuscript in April 2024. It brought 423 emails containing 2490 links after duplicate removal. A first selection was performed on the title and abstract, yielding 238 papers. After manual filtering, we select 158 relevant papers, which amount to 136 new publications since the original survey.

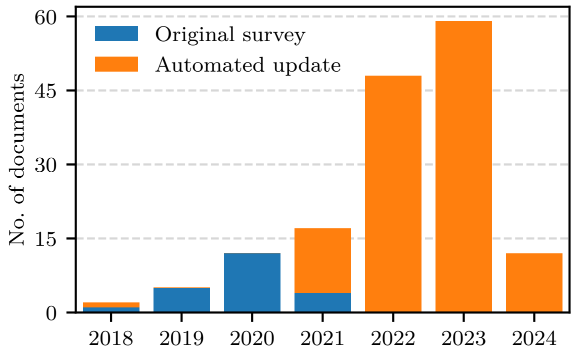
To process this new corpus, we use Litstudy [Hel+22], a Python library providing tools to extract and analyze bibliographic data. On the 158 selected papers, 153 only were available on Scopus, the database available in Litstudy that provides the most complete data. The list of papers is available as appendix of this manuscript.

Literature distribution The distribution of the literature is analyzed in terms of publication year, venue, research group, and authors. We use the properties of the document set generated by Litstudy to filter and group the papers, and use the provided Pandas adn Matplotlib bindings to analyze and plot the data.

Topic modeling Litstudy also provides tools to perform topic modeling on the text data of the papers, mainly title and abstract. We first preprocess the text data by removing stop words, punctuation, and numbers, and then associate each word with its frequency in the corpus. Then, we test the two main approaches of available in the literature, namely Latent Dirichlet Allocation (LDA) and Non-negative Matrix Factorization (NMF), to identify the main topics in the corpus. The presented results have been obtained using the NMF algorithm on 20 topics and after 2000 iterations, as it provided the most interpretable results.



(a) Evolution of the topics using Queries (a) and (b) according to Scopus, up to 2023.



(b) Evolution of the number of publications on FIDSs.

Figure 3.3 – Evolution of the topics and number of publications.

3.3 Quantitative Analysis

This section provides indicators of the representation of FIDSs in the scientific literature: the evolution of the publications, the relevant venues, the active groups, and the topics of interest. Notably, the identification of the most active groups and most relevant venues provides insights on how to keep track of the most recent advances in the field.

3.3.1 Evolution of the Topic

The topic of IDS started to gain traction in the late 1990', as depicted in Figure 3.3a. After a stagnation period, the topic regained interest around 2015, with an increase of the research on Internet of Things (IoT) and Industrial Internet of Things (IIoT) [DAF18; Cha+19], alongside other specific use cases. With the introduction of FL by McMahan *et al.* [McM+17], the community started to explore the application of FL to IDS around the years 2018–2019. Figure 3.3a has been generated using the analytics offered by Scopus and the following queries:

(a) `intrusion AND detection AND system`;

(b) `federated AND learning`.

Recent works on FL focus on its security and privacy-preserving aspects [Ngu+20; LYY20; Mot+21b]. Techniques like homomorphic encryption were introduced as early as 2017 [Har+17], and have been extensively reviewed since. More recently, other privacy-preserving techniques have been applied to FL, such as Multi-Party Computation (MPC) in FLGUARD [Ngu+21] or differential privacy in [KGS21]. FIDSs present a similar tendency with more research towards algorithm security and privacy-preserving techniques. For instance, Li, Wu, *et al.* [Li+20a] use homomorphic encryption to provide a secure and privacy-preserving aggregation of models. Aside from security, variations of Horizontal

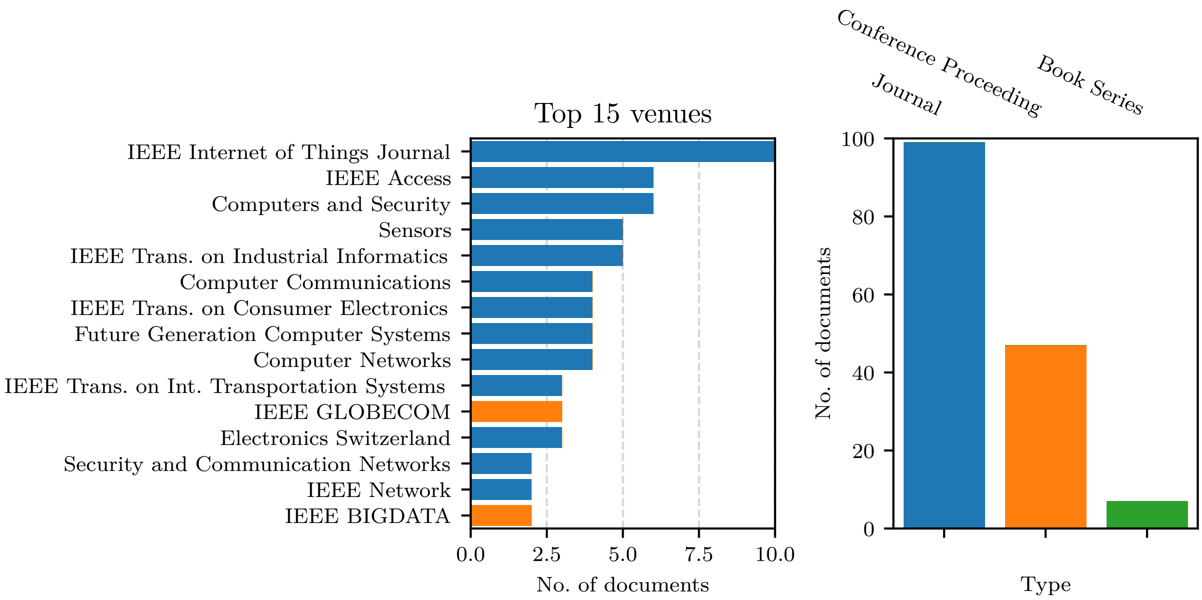


Figure 3.4 – Distribution of the publications in the most recurring venues.

Federated Learning (HFL) started to appear in 2021, such as segmented FL in [Sun+20], as standard HFL has significantly been studied for FIDS.

Finally, the numerous literature reviews published since 2021 [Agr+22; Ala+21; Cam+22; Lav+22b; FNS22; GR22; Ism+24] show the continuous interest of the community for the study of FIDSs. These also show the need for synthesis and structuring of research in this area.

3.3.2 Relevant Venues

The initial study published in 2022 [Lav+22b] observed very few recurring venues for the publication of FIDS research. Indeed, only three venues had more than one publication on the topic: the *IEEE Internet of Things Journal* [Pop+21b; Zha+20a], *IEEE Access* [Che+20; Li+20b], and the *IEEE BigData* conference [Cet+19; Fan+20a]. The original distribution in terms of venue type (11 conferences, 10 journals and 1 book chapter) has significantly changed, since journals represent two thirds of the publications. This is a probable sign of the field gaining maturity, as publishing in conferences first and journals afterward is a common strategy. Figure 3.4 shows the distribution of the publications in the most recurring venues.

Another observation of the initial study was the diversity of the venues, spanning a wide range of topics, from IoT to Industrial Control System (ICS), including transportation systems and extra-terrestrial networks. This diversity is still present in the most recent publications, although a few generic venues now host significantly more publications: the *IEEE Internet of Things Journal*, *IEEE Access* and *Computers & Security*. The latter is the first security-specific venue to appear in the list. Its place in the top

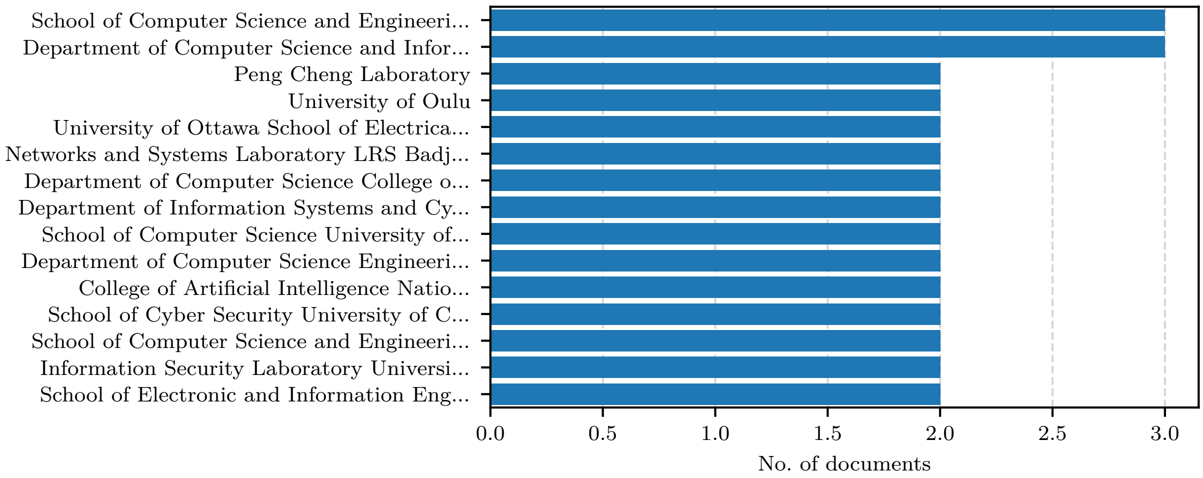


Figure 3.5 – Distribution of the publications by affiliation.

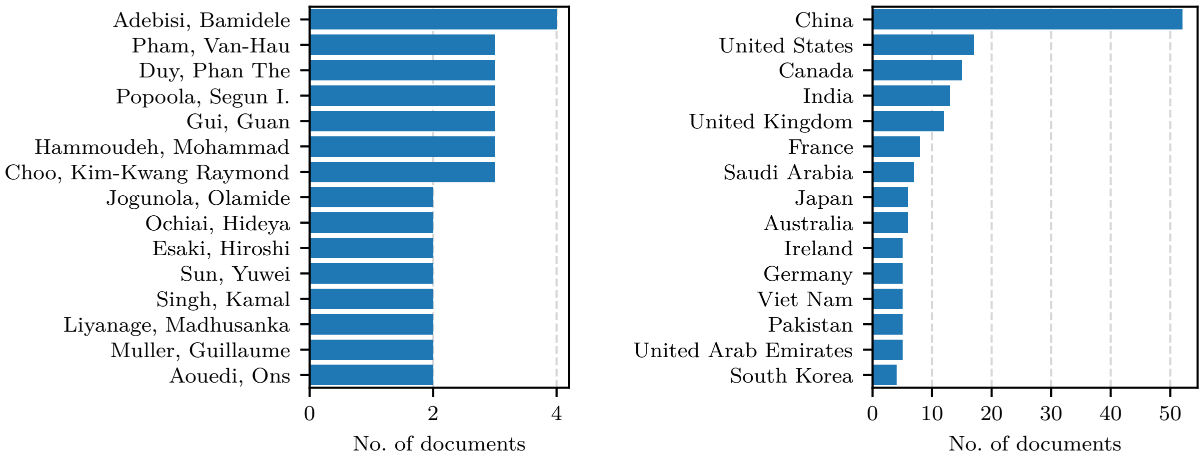
venues is a sign of the increasing interest of the security community for FL and FIDS, as most contributions were previously published in more use-case specific venues. Lastly, while relevant venues have been accepting FL literature since its introduction, they start to host specific tracks or special issues, such as ICDCS's track on "Federated Learning, Analytics, and Deployment", or IEEE BigData's "Special Session on Federated Learning on Big Data".

3.3.3 Active Groups

Since they introduced the topic of FL in 2016, the team at Google Research has been a big influence for the research community [Kon+16b; Kon+16a; McM+17; Bon+17; Bon+19]. They mostly work on the primitives behind FL, such as model aggregation with the `FedAvg` algorithm [McM+17]. The team of TU Darmstadt (Germany) has also been very active in the field, with a focus on IDS with `DÏoT` [Mar+19; Ngu+19] and FL security [Ngu+20]. The two collaborated, bringing `FLGUARD` [Ngu+21] and `FLAME` [Ngu+22], two algorithms focusing on limiting the impact of poisoning attacks in FL. These series of works makes them one of the most impactful groups in the field.

Other noteworthy groups include the Aalto University (Finland) [Ngu+21] and the University of Tokyo (Japan) [Sun+20; SEO21; QK21]. The most active country remains China, with a dozen institutions now amounting to a third of the publications in the field, as illustrated by Figures 3.5 and 3.6b.

Investigating the major authors tells another story, as the most active authors are not necessarily affiliated with the groups mentioned above. In particular, Popoola, Gui, *et al.* co-authored several publications on FIDS [Pop+21b; Pop+21a; Pop+22; Pop+23] as a collaboration between the Nanjing University of Posts and Telecommunications and multiple British universities. Likewise, Duy *et al.*, from the University of Information



(a) Publications by author.

(b) Publications by country.

Figure 3.6 – Distribution of the publications by author and country.

Technology (VNU, Vietnam), are also quite represented in terms of publications [Duy+21; Vy+21; Thi+22; Quy+22].

3.3.4 Topics of Interest

Using topic modeling, we extract the main topics of interest from the 153 publications on FIDSs identified in the updated selection. By construction, the model is unable to differentiate between application domains (such as IoT or ICS) the techniques used (*e.g.*, blockchains) or the addressed challenges in a paper. However, it provides a good overview of the main topics of interest in the field, especially for the consequent amount of literature published since the initial study. Figure 3.7 present the topics identified by the model, with the most recurring keywords for each topic.

First, this analysis highlights the application domains of FIDSs, where the topic of IoT (*i.e.*, `internet_things`, `edge`, `things`) is one of the most recurring (Topics 4, 9, and 10). Other applications stand out, such as ICS (`industrial`, `iiot`), Internet of Medical Things (IoMT) (`medical`, `healthcare`), Vehicle-to-Everything (V2X) (`vehicle`, `vehicular`, `transportation`), and Software-Defined Networking (SDN) (`software`, `defined`, `sdn`). These applications also correlate with the venues identified in Section 3.3.2, as the *IEEE Internet of Things Journal* or the *IEEE Trans. on Industrial Informatics* do focus on IoT and ICS, respectively. Figure 3.8b depicts the distribution of the publications by domain overall.

Likewise, some topics are directly associated with the challenges identified in Section 3.6.2. For instance, Topic 0 (`poisoning`, `defense`, `malicious`) represents works focusing on adversarial attacks against FIDSs and their mitigation. Some techniques can also be extracted from these results. For instance, Topic 0 also contains `similarity` as

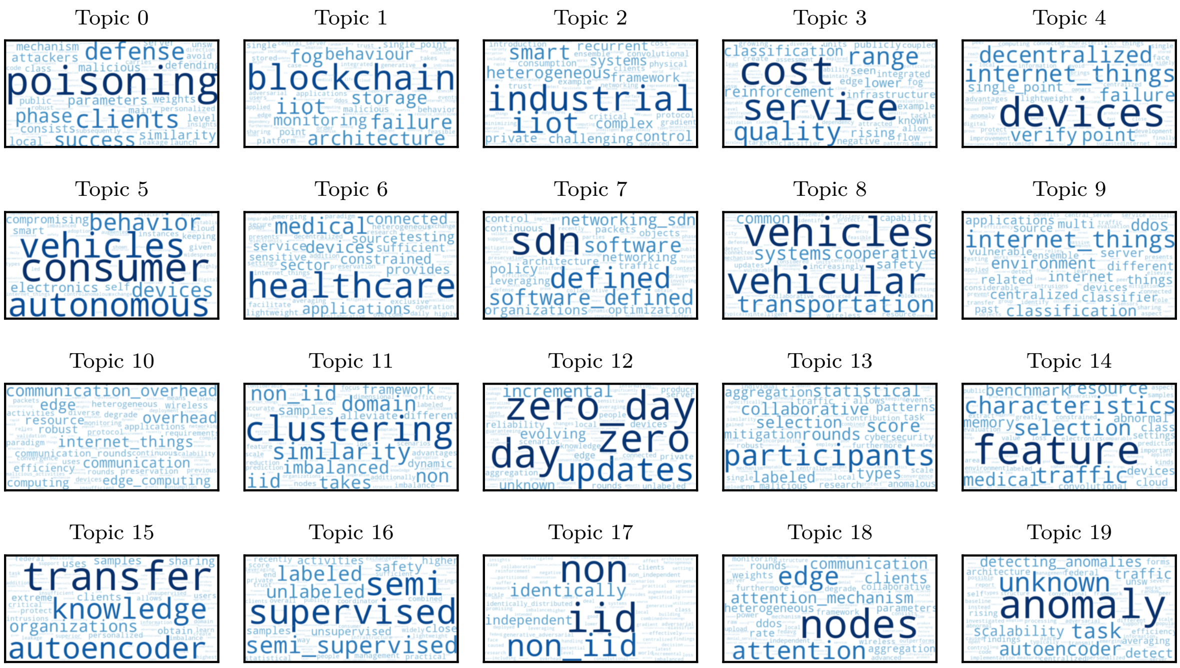
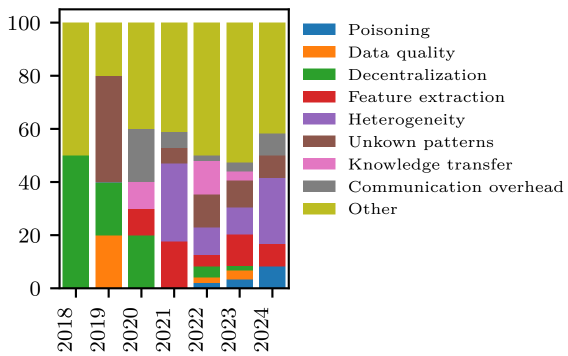
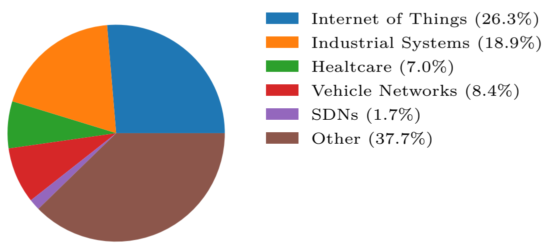


Figure 3.7 – Topics of interest in the field of FIDSs.



(a) Distribution of the addressed challenges over time.



(b) Distribution of the publications by domain.

Figure 3.8 – Exploiting the topics of interest.

a keyword, which is likely to refer to the use of similarity metrics to detect poisoning attacks. This is indeed one of the most represented mitigation techniques in the literature on FIDS [Yan+23] or FL alike [FYB20; Ngu+22]. Figure 3.8a depicts the distribution of the addressed challenges over time. Unlike the distribution of the publications by domain, some challenges are addressed in the literature much later, such as handling the heterogeneity of the data (**??**) or resisting to adversarial attacks (**??**). Both are challenges that have been identified in the initial study as open issues in the field [Lav+22b; Lav+22a].

3.4 Qualitative Analysis

This section contains the results of our literature review. First, it synthesizes the analyses into a reference architecture and a taxonomy for FIDSs, which help structure the field. Then, it goes over a comparison of the selected works to answer Questions RQ1-2.a to RQ1-1.b on the components of FIDSs and their impact on performance.

3.4.1 Structuring the Literature

The qualitative (Section 3.4) and quantitative (Section 3.3) analyzes provide results that we synthesize hereafter in a reference architecture and a taxonomy. The reference architecture presents the components of FIDSs and their interactions, while the taxonomy provides comparison criteria for the selected works.

We build the taxonomy upon different existing ones related to Collaborative Intrusion Detection System (CIDS) [VKF15; ZLK10], ML–based intrusion detection [dCos+19], and FL [Ale+20; LYY20; Mot+21b]. First, we extract classes relevant to the domain of FIDS, before filtering out irrelevant ones by validating the taxonomy against the reference architecture (Figure 3.9). The latter displays both the operation and the design of the system. By confronting the taxonomy and the architecture, we ensure that each item of the taxonomy is related to a component of the architecture, and *vice versa*. Then, we add any commonalities between the selected works that are not already represented in the previous taxonomies. This identifies new criteria on which to compare the selected works.

Reference Architecture

This section presents the reference architecture synthesized from the selected works, as depicted in Figure 3.9. The architecture provides a summary of the components of FIDSs and their interactions, answering Question RQ1-2.a. It can be divided in three parts:

— The *Managed system* represents the monitored system, *e.g.*, Information Technology (IT) network, industrial devices, or health-monitoring wearables. As noticed in Section 3.4.2, collected data can either concern system or environment behavior. The former relates to information generated by the systems, *e.g.*, network traces or

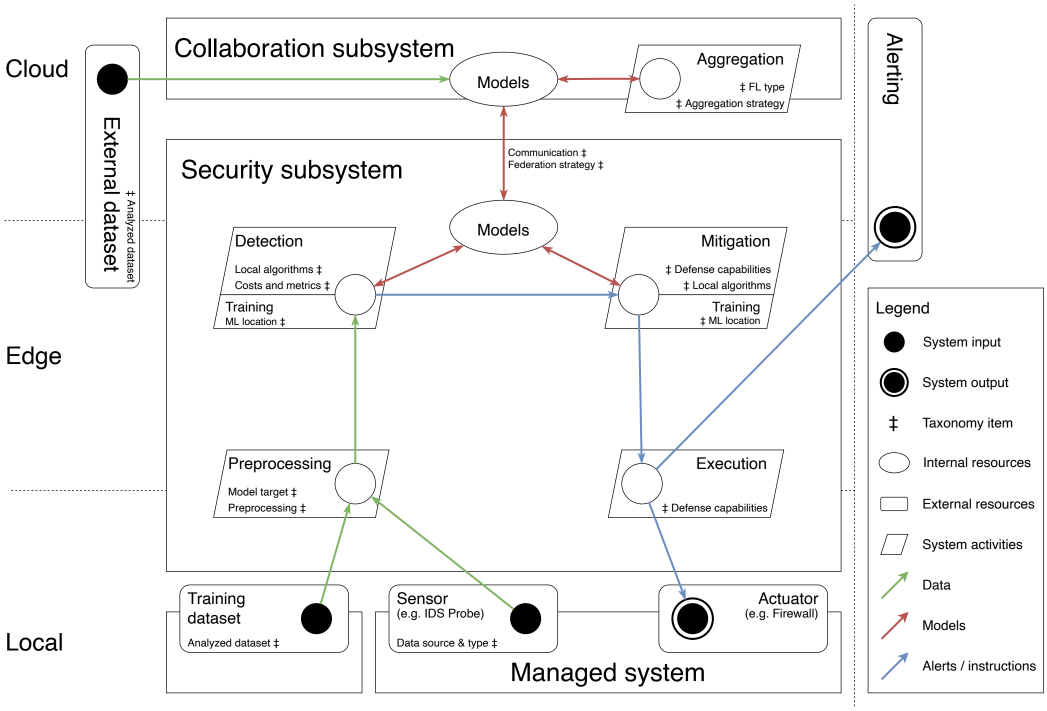


Figure 3.9 – The proposed reference architecture for FIDSs. Figure from Lavaur, Pahl, *et al.* [Lav+22b] © IEEE 2022.

resource consumption. The latter refers to what the monitored system operates on, *e.g.*, health metrics for medical devices of temperature and atmospheric pressure for building management systems.

— The *Security subsystem* is the core of the architecture. It contains all the system's activities, from model training to detection and counter-measures deployment. Depending on the objectives and constraints, this subsystem can either be run locally like [PA18] or [Hei+20], on a dedicated edge-device as in [Li+20a]. In the case of centralized learning, this entire subsystem runs in the cloud. The subsystem is assumed to run a device that embeds enough computing power to perform real-time anomaly detection against ML models. It is also capable of training its own model based on collected data.

— The *Collaboration subsystem* provides the *sharing* feature of the system, essentially model aggregation (Section 3.4.2). It also provides optional training from other sources, like online datasets.

This architecture has similarities with the principles of autonomic systems, as defined by IBM in 2001 [KC03], referred to as Monitor-Analyze-Plan-Execute plus Knowledge (MAPE-K). Classic autonomic systems are local, and therefore use a database to provide *knowledge*. In FIDS, FL fills this role in the reference architecture, as the knowledge is being shared among all agents through model aggregation.

Taxonomy for FIDS

The taxonomy depicted in Figure 3.10 summaries the core components and specificities of FIDSs, as extracted from the selected works and existing related taxonomies. Correlations between the taxonomy items and the system's components can be seen in the reference architecture (Figure 3.9). It also serves as a framework for the comparisons of the selected works. Each class represents a building block, for which multiple approaches exist depending on use case and constraints.

The proposed taxonomy contains 12 classes describing the selected works that span over five main aspects:

— Two classes cover the topic of **Data**: *Data Source and Distribution* and *Preprocessing*. It defines the type of data considered and how it is distributed among clients, how it is collected, and the preprocessing strategies that are used.

— **Local operation** is represented by 3 classes: *Algorithm location*, *Local Algorithm*, and *Defense Capabilities*. It describes the detection and mitigation strategies, how models are built and trained, and where the computing resources are located.

— The **Federation** aspect is covered by 2 classes: *Federation Strategy* and *Communication*. They refer to the communication between the agents and the server, and how data sharing is organized.

— **Aggregation** is also covered by 3 classes: *FL Type*, *Aggregation Strategy*, and *Model Target*. It describes the type of FL used, how the models are merged, in accordance with the objectives of the system.

— Finally, 2 classes address the **Experimentation** topic: *Analyzed Dataset* and *Costs and Metrics*. This meta-category does not relate to the proposed solution, but to how the experiments are performed.

3.4.2 Federated Learning for Intrusion Detection

This section reviews the selected literature. Using the taxonomy as a reference, it details and compares the selected works. Table 3.1 summarizes the information and helps identify differences between the works. It gives partial answers to research questions about the components of FIDSs and how to measure their impact on performance (Questions RQ1-2.a and RQ1-2.b), while Section 3.4.2 replies to Question RQ1-1.b about federation techniques.

Data Source and Distribution

The selected works highlight two main characteristics of the training data that impact the design of FIDSs: the origin of the data and its distribution among clients. The type of data used in the selected works is diverse, ranging from network traffic [CZY20; RWP19] to

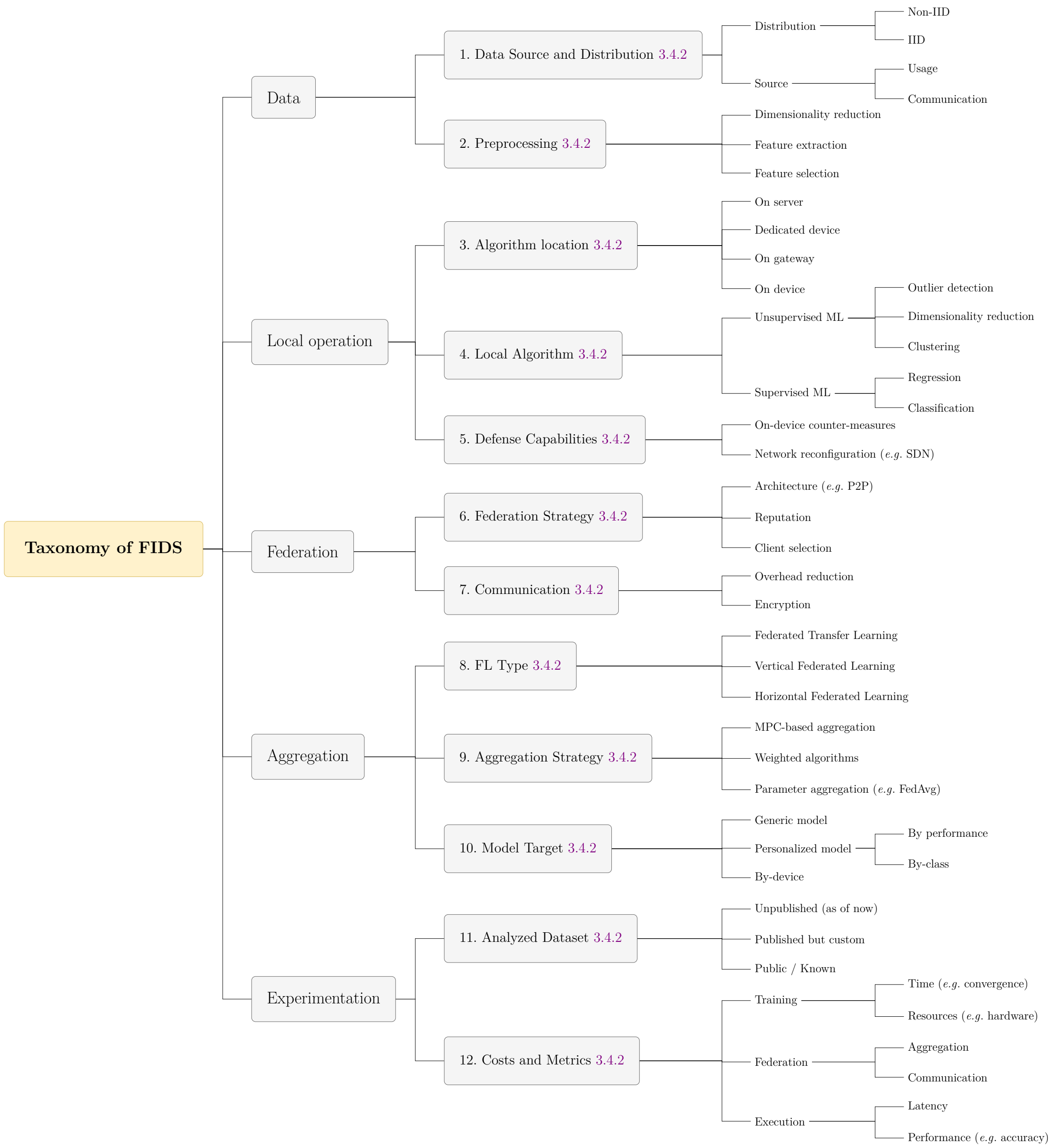


Figure 3.10 – *Proposed taxonomy for FIDS.* Figure from Lavaur, Pahl, *et al.* [Lav+22b] © IEEE 2022.

Table 3.1 – Comparative overview of selected works in the original study—approach and objectives (1/2).

	Ref	Internet of Things	Information Technologies	Cyber Physical Systems	Autonomous Vehicles	Satellite-terrestrial networks	Horizontal FL	Vertical FL	Federated Transfer Learning	Federated MTL	Federated Mimic Learning	Online learning	Semi-supervised	Supervised	Unsupervised	Personalized models	Network-based	Usage-based	Training location	Data type	Strengths
2018	Pahl and Aubet [PA18]	●	○	○	○	○	●	○	○	○	○	●	○	○	●	●	●	○	Device	Abstracted network traffic (middleware)	relatively lightweight, online, no labels
2019	Rathore, Wook Kwon, and Park [RWP19]	○	●	○	○	○	●	○	○	○	○	○	●	○	○	●	●	○	Edge-controller (SDN)	Network traffic (SDN)	offers mitigation, decentralized
2019	Schneble and Thamilarasu [ST19]	●	○	○	○	○	●	○	○	○	○	●	○	○	●	●	●	○	Gateway	IoT network traffic (TCPdump)	online, offers per-class models, no labels
2019	Nguyen, Marchal, *et al.* [Ngu+19]	○	●	○	○	○	○	○	○	●	○	○	●	○	○	●	●	○	Gateway	Encrypted network traffic (CICFlowMeter)	versatile (multi-task)
2019	Zhao *et al.* [Zha+19]	○	○	●	○	○	●	○	○	○	○	●	○	○	●	○	○	●	Gateway	Healthcare sensor values	high adaptability, no labels
2019	Cetin *et al.* [Cet+19]	○	●	○	○	○	●	○	○	○	○	○	●	○	○	○	●	○	Gateway	Network traffic (WIFI)	–
2020	Li, Wu, *et al.* [Li+20a]	●	○	○	○	○	●	○	○	○	○	○	●	○	○	○	○	●	Gateway	Air conditioner sensor values	offers traceability (blockchain)
2020	Chen, Zhang, and Yeo [CZY20]	○	○	●	○	○	●	○	○	○	○	○	●	○	○	○	●	○	Gateway	MODBUS traffic	confidentiality (encryption)
2020	Zhang, Lu, *et al.* [Zha+20a]	○	●	○	○	○	●	○	○	○	○	○	●	○	○	○	●	○	Device	IoT network traffic (TCPdump)	–
2020	Fan *et al.* [Fan+20a]	○	●	○	○	○	●	○	○	○	○	○	○	○	●	○	●	○	Gateway	IoT network traffic (TCPdump)	no labels
2020	Rahman *et al.* [Rah+20]	○	●	○	○	○	●	○	○	○	○	○	●	○	○	●	●	○	Gateway	Network traffic (PCAP)	segmented (performance-based models)
2020	Sun, Ochiai, and Esaki [SOE20]	●	○	○	○	○	○	○	●	○	○	○	●	○	○	●	●	○	Gateway (MEC)	IoT network traffic (TCPdump, CICFlowMeter)	knowledge transfer between public and private datasets
2020	Al-Athba Al-Marri, Ciftler, and Abdallah [ACA20]	○	●	○	○	○	○	○	○	○	●	○	●	○	○	○	●	○	Gateway	Network traffic (TCPdump)	enhanced privacy (mimic learning)
2020	Kim, Cai, *et al.* [Kim+20]	○	●	○	○	○	●	○	○	○	○	○	●	○	○	○	●	○	Gateway	Network traffic (TCPdump)	–
2020	Qin, Poularakis, *et al.* [Qin+20a]	○	●	○	○	○	●	○	○	○	○	○	●	○	○	●	●	○	Gateway (SDN)	Network traffic (SDN)	very lightweight, line-speed classification, P4 language compatible
2020	Chen, Lv, *et al.* [Che+20]	○	●	○	○	○	●	○	○	○	○	○	●	○	○	○	●	○	Gateway	Network traffic (CICFlowMeter)	robust to poisoning, scalable
2020	Hei *et al.* [Hei+20]	○	●	○	○	○	●	○	○	○	○	●	○	●	○	○	●	○	Device	Network traffic (TCPdump)	online, offers traceability (blockchain)
2020	Li, Zhou, *et al.* [Li+20b]	○	●	○	○	●	●	○	○	○	○	○	●	○	○	○	●	○	Gateway	Network traffic (PCAP, CICFlowMeter, Argus)	relatively lightweight, confidentiality (encryption)
2021	Liu, Zhang, Zhang, *et al.* [Liu+21]	●	○	○	○	○	●	○	○	○	○	○	●	○	○	○	●	○	Gateway	IoT network traffic (TCPdump, Argus)	zero-days detection
2021	Popoola, Gui, *et al.* [Pop+21b]	○	●	○	○	○	●	○	○	○	○	○	●	○	○	●	●	○	Device	Network traffic (TCPdump)	relatively lightweight
2021	Qin and Kondo [QK21]	○	○	○	●	○	●	○	○	○	○	○	●	○	○	○	●	○	Device	Network traffic (TCPdump)	decentralized
2021	Sun, Esaki, and Ochiai [SEO21]	○	●	○	○	○	●	○	○	○	○	○	●	○	○	●	●	○	Gateway	Network traffic (PCAP)	segmented (performance-based models)
		Use case					FL type					Training					Approach				

sensor values [Zha+20a; ST19]. The former is significantly more represented, probably due to the availability of public datasets like CICIDS2017 [SHG18] and UNSW-NB15 [MS15] (see Section 3.4.2).

Most papers [CZY20; RWP19; Ngu+19; Li+20a; Rah+20; SOE20; Pop+21a; Hei+20] use similar network features, such as source and destination, local and remote ports, TCP flags, protocol, and packet length. The authors of [Qin+20a] also target network features but at packet-level, all translated to 1D vectors: IP addresses, layer-4 protocol, ports, and IP packet length as a 120-bit input vector. Li, Wu, *et al.* [Li+20a] also explore network-related features in their use case of satellite communications. These values can be completed with preprocessing (see Section 3.4.2) to extract other features from the raw data. For instance, both Pahl and Aubet [PA18] and Nguyen, Marchal, *et al.* [Ngu+19] analyze the periodicity of packets, which is notably useful for volumetric attack detection. By using a middleware to classify the data, Pahl and Aubet [PA18] can train per-class models. Such models are more specialized and thus more accurate, but most communication layers do not provide such metadata. Training per-class models usually requires then a prior classification step, like in [Ngu+19]. The use of specialized models is further discussed in Section 3.4.2.

On the other hand, Zhang, Lu, *et al.* [Zha+20a] and Schneble and Thamilarasu [ST19] use sensor values, such as hearth rate and oxygen saturation. In this case, one does not seek to detect intrusions per se, but rather anomalies in the data that could indicate a malfunction or an attack. The observed data can be seen as a side-channel, leaking information about the actions of potential attackers. More recently, FL has been applied to Host-based Intrusion Detection Systems (HIDSs) [Guo+23], were similar considerations apply, particularly in terms of data distribution.

Finally, even when considering the same data type, use cases introduce significant differences in the available features. For instance, two systems targeting the communication between devices may encounter different protocols, services, and even communication support. In the literature, the most common use cases are (sorted by representation): Information Technology (IT), Internet of Things (IoT), Cyber-Physical System (CPS), and Autonomous Vehicles (AV). While it is unlikely that a system would target multiple use cases, discrepancies in the data distribution can exist within a single use case. Chen, Zhang, and Yeo [CZY20], and partly Hei *et al.* [Hei+20], address the topic of skewed data distribution. A non-Independent and Identically Distributed (IID) data distribution can negatively impact training performance [Yan+19]. However, most real-world scenarios generate non-IID data, which is a major drawback of the selected works, as most of them do not address this issue.

Preprocessing

In addition to the type of data considered, the preprocessing pipeline has a significant impact on the performance of the system. Preprocessing implies the transformation of raw data into a format that can be better leveraged by ML models, either by extracting new features or by reducing the dimensionality of the data. Three main non-exclusive approaches are distinguishable in the selected works: feature extraction, feature embedding, and feature selection:

— *Feature extraction* refers to the computation of numerical characteristics after the data collection; *e.g.* Inter-Arrival Time (IAT) or number of packets per device in the context of traffic monitoring. For instance, both Nguyen, Marchal, *et al.* [Ngu+19] and Pahl and Aubet [PA18] extract periodicity features from the data. Because they only process binary features, Qin, Poularakis, *et al.* [Qin+20a] extract numerical features, and convert them to 1D vectors.

— *Feature embedding* or *dimensionality reduction* is used for algorithms that do not deal efficiently with high-dimensional vectors. We mostly use the term *embedding* when the authors use Deep Learning (DL) techniques, as it implies that the model learns the best representation of the data, such as with autoencoders [CZY20]. Other *dimensionality reduction* techniques include Principal Component Analysis (PCA), used for example by Kim, Cai, *et al.* [Kim+20].

— *Feature selection* relates to the automated selection of relevant features, before learning. For instance, Qin and Kondo [QK21] use a greedy feature-selection algorithm based on accuracy, while logistic regression can be used to eliminate features with a recursive algorithm [ACA20].

The other works [Zha+20a; ST19; Li+20a; RWP19] do not emphasize on their feature selection strategy. Moreover, some papers [Li+20a; ST19; Zha+19] use datasets that contains computed features (3.4.2). For experiments on live prototypes, feature computation is required.

Depending on the use case, additional features after *feature selection* or *extraction* may vary. Network analysis often relies on basic features, such as addresses and ports for source and destination, protocol, data type, packet length, and timestamp. However, these characteristics can also vary regarding their provenance: network capture [Sig99; Tav+09] or abstracted communications [PA18]. Extracted features are very common, such as inter-packet time, bytes sent per host, or bytes per packets [BG16; Cha+19]. For instance, both Nguyen, Marchal, *et al.* [Ngu+19] and Pahl and Aubet [PA18] target IoT devices, which have a sporadic, but periodic and thus more predicable traffic. In this context, anomaly in the packet-sequence, or in the inter-arrival time might indicate an attack.

Usage-based analysis, on the other hand, is entirely dependent on the monitored device. Schneble and Thamilarasu [ST19] monitor health-related features, like arterial blood pressure or the raw ECG signals. The authors of [Zha+20a] focus on air conditioners, and therefore measure related information such as water or air temperature.

Algorithm location

The proposed taxonomy (3.10) considers three types of locations: on-device, on-gateway, and on-server. However, a large majority of the literature concerns either on-device training, or uses a dedicated device acting as a gateway. Most selected works use a dedicated device to perform the analysis, while the others assume the devices can support their own processing. Some hybrid approaches also exist, such as the multi-stage aggregation used by Liu, Zhang, Zhang, *et al.* [Liu+21], where models can be trained and aggregated at different stages of the edge–cloud continuum.

In most cases, it is the use case that dictates the model training location, as each comes with specific constraints. For instance, Zhang, Lu, *et al.* [Zha+20a] focus on a medical use case where the analyzed data solely consists of sensor measurements (Section 3.4.2). Connected sensors are typically lightweight devices unable to process data, so they require a gateway to be usable. Most works [Li+20a; CZY20; ST19; Zha+19; ACA20; Kim+20; CZY20; Pop+21a] rely on gateways because they are more suitable for traffic analysis. It allows to capture all communications, even if the devices communicate on different supports (*e.g.*, IEEE 802.3 *vs.* IEEE 802.11). Gateway-based processing can also be motivated by the architecture of the monitored system. For instance, the authors of [Fan+20a] reuse the existing infrastructure of 5G by exploiting Mobile Edge-Computing (MEC) gateways to capture traffic and perform analysis for a 5G IoT use case. In some specific use cases, like SDN, gateways can even offer additional features that can be leveraged by FIDSs, such as packet re-routing [RWP19] or packet-level analysis [Qin+20b].

Other works [PA18; Hei+20; QK21; Rah+20] assume that end-devices are powerful enough to support their own processing. While this is generally less realistic, it can be the case for some specific use cases, like Autonomous Vehicles (AV) [Liu+21]. Indeed, such vehicles often carry consequent processing abilities for environment recognition alone, and are thus assumed to be able to perform ML training.

Local Algorithm

As discussed in the Background and Preliminaries 2, one key aspects of ML for intrusion detection is its adaptability to the monitored system. Online learning refers to the ability to train a model continuously as data arrives, whereas offline learning refers to a one-shot training on a defined training set. Only four of the selected works adopt online learning [PA18; Ngu+19; ST19; Hei+20]. All online work in the selection use either unsupervised or semi-supervised approaches, as continuously feeding labeled data is impracticable. Offline learning algorithms can be re-trained to adapt to new data, but this is not addressed in the selected works.

Another key difference lies in the type of algorithm used. Neural Networks (NNs), and most particularly Deep Neural Network (DNN), massively outnumber other approaches in the selected works (21 out of 22, see Table 3.2). This is coherent with the state of the art in ML for intrusion detection, as DNNs are also vastly represented in the literature. In the selection, most works rely on either Multilayer Perceptrons (MLPs) (9 out of 22) or Convolutional Neural Networks (CNNs) (4 out of 22). Recurrent Neural Networks (RNNs) are also used in 3 works, but mostly in combination with other architectures.

This sections highlights the predominance of DNNs in the selected works. These finding can be generalized to the literature published since the original study, as confirmed the more recent work of Isma'ila *et al.* [Ism+24]. Indeed, DNNs are particularly well-suited for FL:

— they are parametric models, meaning they can be aggregated using mathematical operations on their parameters (Section 3.4.2);

— their layers learn different levels of abstraction, enabling partial aggregation and specialized training (Section 3.4.2);

— their architecture can be adapted to the monitored system, as they can be trained on different types of data and for different objectives (Section 3.4.2).

Consequently, this choice is both relevant for intrusion detection and as a base model to be used in FL.

Defense Capabilities

Defense strategies are barely covered in the selected works, as only one paper provides actionable counter-measures. Rathore, Wook Kwon, and Park [RWP19] leverage SDN technologies, allowing the SDN controller to modify the network architecture in case of an attack. The proposed solution is tailored for Denial of Service (DoS) or flooding attacks, and therefore only needs to block the responsible traffic flow.

FIDSs could also provide remediation capabilities, providing automated resilience of a monitored system [Gho+07]. To the best of our knowledge, there is no such work in the literature. However, multiple works have been proposed to provide self-healing behaviors to information systems [EA10; Ali+18]. Such functionalities could be considered to enhance FIDS capabilities.

Federation Strategy

Another key aspect of FIDSs is how the federation is organized. This depends on the scale of the system and its architectural constraints, which are both yet again influenced by the use case. To cope with large-scale settings, massive FL applications often implement a client-selection algorithm which only train a subset of participant at each

Table 3.2 – Comparative overview of selected works in the original study—algorithms and performance (2/2).

Ref		Local Algorithm	Federation Algorithm	Accuracy	Precision	Recall	Fall-out	F-Score	K [a]	Dataset
2018	Pahl and Aubet [PA18]	BIRCH K-means	Parameter addition	0.9900	–	0.9600	0.0020	–	7	Generated
2019	Rathore, Wook Kwon, and Park [RWP19]	MLP	Early model fusion	‡ 0.9100	‡ 0.9100	‡ 0.9100	–	‡ 0.9100	15	NSL-KDD [Tav+09]
2019	Schneble and Thamilarasu [ST19]	MLP	Weight and biases average	0.9930	–	–	–	–	64	MIMIC [Joh+16]
2019	Nguyen, Marchal, *et al.* [Ngu+19]	GRU	FedAvg	–	–	0.9543	0	–	15	Generated
2019	Zhao *et al.* [Zha+19]	FC (shared layers) → FC	Weight and biases average	∗ 0.9797	∗ 0.9634	∗ 0.9681	–	–	–	CICIDS2017 [SHG18] ISCXVPN2016 [Dra+16] ISCXTor2016 [Hab+17]
2019	Cetin *et al.* [Cet+19]	SAE	FedAvg	–	–	–	–	–	933	AWID [Kol+16]
2020	Li, Wu, *et al.* [Li+20a]	CNN-GRU → MLP	Homomorphic parameter addition	0.9920	0.9885	0.9745	–	0.9813	7	CPS dataset [MG14]
2020	Chen, Zhang, and Yeo [CZY20]	DAGMM	Parameter addition	–	0.7447	0.9803	–	‡ 0.8700	2 [b]	KDD 99 [Sig99]
2020	Zhang, Lu, *et al.* [Zha+20a]	MLP	CDW_FedAvg	∗‡ 0.8900	∗‡ 0.8600	∗‡ 0.9450	–	∗‡ 0.8500	4	Generated
2020	Fan *et al.* [Fan+20a]	CNN	Parameter aggregation	∗ 0.9100	–	∗‡ 0.9350	∗‡ 0.0020	–	4	CICIDS2017 [SHG18] NSL-KDD [Tav+09] Generated
2020	Rahman *et al.* [Rah+20]	MLP	FedAvg	∗ 0.7731	–	–	–	–	4	NSL-KDD [Tav+09]
2020	Sun, Ochiai, and Esaki [SOE20]	CNN	Parameter aggregation	∗ 0.8710	–	–	–	–	20	LAN-Security Monitoring Project [Hid18]
2020	Al-Athba Al-Marri, Ciftler, and Abdallah [ACA20]	MLP with Dropouts	FedAvg	0.9812	∗ 0.9900	∗ 0.9900	∗ 0.1320	∗ 0.9900	10	NSL-KDD [Tav+09]
2020	Kim, Cai, *et al.* [Kim+20]	MLP	FedAvg	0.9712	–	–	–	–	4	NSL-KDD [Tav+09]
2020	Qin, Poularakis, *et al.* [Qin+20a]	BNN	SignSGD	∗ 0.9640	∗ 0.9555	∗ 0.8645	–	∗ 0.9055	8	CICIDS2017 [SHG18] ISCX Botnet 2014 [Big+14]
2020	Chen, Lv, *et al.* [Che+20]	GRU-SVM	FedAGRU	∗ 0.9905	–	–	∗ 0.0108	∗ 0.9762	20	CICIDS2017 [SHG18] KDD 99 [Sig99] WSN-DS [AAA16]
2020	Hei *et al.* [Hei+20]	MLP	FedAvg	∗‡ 0.8950	∗‡ 0.9750	∗‡ 0.8775	–	∗‡ 0.9225	3	DARPA 1999 [Hai+01]
2020	Li, Zhou, *et al.* [Li+20b]	CNN	Homomorphic parameter addition	∗ 0.8100			∗ 0.1900		4	Generated
2021	Liu, Zhang, Zhang, *et al.* [Liu+21]	MLP	Parameter aggregation	‡ 0.9600	0.9400	0.9500	–	–	6	KDD 99 [Sig99]
2021	Popoola, Gui, *et al.* [Pop+21b]	MLP	FedAvg	∗ 0.9939	∗ 0.9819	∗ 0.9676	–	∗ 0.9728	5	Bot-IoT [Kor+19] N-BaIoT [Mei+18]
2021	Qin and Kondo [QK21]	ONLAD [TKM20] (ELM + AE)	FedAvg	0.7040	–	–	–	–	8	NSL-KDD [Tav+09]
2021	Sun, Esaki, and Ochiai [SEO21]	CNN	Parameter aggregation	–	–	–	–	∗ 0.8930	20	LAN-Security Monitoring Project [Hid18]

Metrics

∗ Value is an average of those provided by the authors.
‡ Value is read from a graph in the article, and may vary a few from the exact value.

[a] K is the highest number of client considered in the experiments.
[b] Chen, Zhang, and Yeo [CZY20] measure how one client performs, by training one other.

round. This reduces the computing load and bandwidth consumption at the expense of a slower convergence due to its stochastic nature—see Chapter 2. The selected works do not discuss this aspect particularly deeply, although some observed positive results from increasing the number of clients [Li+20a; ST19; Ngu+19]. Client selection can even be done dynamically [Zha+20b], even though it is not discussed in the selected works. More recent works leverage client selection, either to improve performances [Che+22], or to mitigate the risk of malicious contributions [Cun+24].

On an architectural perspective, most FL implementations follow a client-server model, where the server acts as an orchestrator distributing training tasks and model updates. This is true for most of the selected works (18 out of 22). While relatively easy to deploy, such approach has caveats, such as the necessity of trusting the central server, or the Single Point-of-Failure (SPoF) in the aggregation process [Ale+20]. To mitigate these issues, some works propose (partly) decentralized approaches, using Distributed Ledger Technologies (DLTs) to store models and updates [RWP19], or enable traceability of the training data using Merkle trees [Zha+20a]. The multi-stage aggregation proposed by Liu, Zhang, Zhang, *et al.* [Liu+21] leverages DLTs to aggregate models between Roadside Units (RSUs)—which connect vehicles to the rest of the world in the V2X paradigm. The vehicles are also able to share their models with each other, in a manner resembling gossip learning. Finally, Hei *et al.* [Hei+20] use the Hyperledger Fabric [And+18] to provide integrity and redundancy.

Communication

FL implies a significant amount of communication between the clients and the server, even though it remains more communication-efficient than distributed Gradient Descent (GD) algorithms [McM+17]. Some selected works try to reduce the communication overhead generated by their solution. Schneble and Thamilarasu [ST19] and Zhang, Lu, *et al.* [Zha+20a] compare the communication used by their system in model sharing, and compare it with the dataset size, which would require to be transferred in non-FL settings. While their results show that the relevance of FL to limit communication usage can be questioned in small datasets, its strength is undeniable with standard use cases—above 10^5 bytes according to [Zha+20a]. The communication overhead is one of the advantages of FL over centralized ML approaches.

The communication between the clients and the server can also be secured. The authors of [Li+20a] and [Li+20b] use homomorphic encryption to aggregate the parameters without the server knowing the generated model. The Paillier crypto-system supports addition [Pai99], which is performed on the server, before the result is disseminated back to the clients. Each client can then decrypt the generated model, and devise the parameters by the number of participants to obtain the averaged biases and weights.

FL Type

As introduced in Chapter 2, most FL implementations use HFL, 18 out of 22. Vertical Federated Learning (VFL) is not represented in the selected works, and only found once in the updated literature selection [NDG22]. As VFL requires having the same samples but different features, it is more difficult to apply to a CIDS use case. Having the same samples would mean that the different participants monitor the same devices, just using different features, which does not follow the motivations of this work. Nevertheless, VFL might be relevant for correlation purposes in a local architecture, or between Computer Emergency Response Teams (CERTs) to share information about common threats.

On the other hand, some papers show that Federated Transfer Learning (FTL) can be used to train models in different but related contexts. For instance, a model trained on the periodicity of specific devices as in [PA18; Ngu+19] would not perform well against devices with behaviors that are too different. However, with FTL, one could quickly train a local model specific to his devices, using the knowledge acquired previously by others, as in [Fan+20a]. Another application of this concept is used by [Zha+19] with Multi-Task Learning (MTL), where a same model is trained simultaneously for multiple tasks. Like in FTL, the model is retrained after the sharing to have personalized behavior.

Al-Athba Al-Marri, Ciftler, and Abdallah [ACA20] implement Federated Mimic Learning (FML) to improve data privacy. Mimic learning is a technique that use two models and two datasets to train and share information afterward. *Teacher* model is trained on the real and sensitive data, and used to label a public dataset. *Student* model is then trained on the newly labeled public dataset, and shared with other participants after that.

Aggregation Strategy

The aggregation strategy is at the core of FL, as it was the original contribution [McM+17] separating it from distributed GD algorithms. More specifically, the base principle of training over multiple epochs locally and aggregating the models (*i.e.*, `FedAvg`) is one of the key components of FL. This approach is the base of most implementations going forward. Naturally, it is also the most represented aggregation algorithm in the selected works, as 6 out of 22 [Ngu+19; Pop+21a; QK21; ACA20; Kim+20; Rah+20] use it directly. Others leverage alternative weighting mechanisms, where `FedAvg` weights models based on the number of samples they have been trained on.

Zhang, Lu, *et al.* [Zha+20a] propose to weight the aggregation according to the centroid distance between the positive and negative classes of the client. They claim it reduces the impact of heterogeneity in the data distribution. Other articles average the parameters of the uploaded models [ST19; CZY20], while not mentioning `FedAvg` explicitly. The aggregation weights also represent an opportunity to balance the clients contributions. This is widely used in the FL literature, whether it is based on the number of samples [McM+17], on the participants' obtained reputation [Wan+22; WK21], or based on model-quality metrics [Den+21].

Because they use Binarized Neural Networks (BNNs) with only binary values, the authors of [Qin+20a] cannot simply average the model parameters. While the last layer of the BNN could be converted to numerical values to be aggregated more easily, the authors prefer the binary approach SignSGD [Ber+18]. This aggregation algorithm relies on majority voting to estimate the best weights for the layers. While their system performs well, the authors point out that updates that do not change the sign of the weights represent a waste of resources, since only two values are possible, $+1$ or -1.

Further, we also considered works in the literature that push the boundaries of the aggregation process, even if it does not suit the formal definition of FL. For instance, Rathore, Wook Kwon, and Park [RWP19] use early model fusion, a technique that concatenates the feature vectors of the models to learn the best feature representation. Pahl and Aubet [PA18] use Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) clusters, which have the particularity of being easily aggregated by simply adding the features of multiple clusters together. Timestamps are also saved to detect the *staleness* of the clusters.

Finally, multiple aggregations can be performed in a single system, whether it is hierarchically [Liu+21] to reduce the communication overhead, or in different clusters in parallel [SOE20; SEO21] to reduce the heterogeneity inside communities. The recent literature contains more works leveraging clustered FL to build more homogeneous communities [Cai+22; Sha+24]. Clustering will be extensively leveraged in Chapter 7 to detect malicious contributions in heterogeneous settings.

Model Target

The existing literature on FIDS highlights how building a generic efficient model is difficult. Nguyen, Marchal, *et al.* [Ngu+19] stress that anomaly detection systems suffer from lower performance when monitoring multiple behaviors at the same time. This especially impacts the false positive rate and sensitivity in their experiments. To address this issue, they propose to use an autonomous classification system [Mar+19] to categorize devices first, and then train per-class models. However, this classification problem is not specific to intrusion detection, and standards have been proposed for devices to advertise such information. Manufacturer Usage Description (MUD) [LDR19] for instance allows devices to signal to the network what type of functionalities and authorizations that they require for operating properly. While they do not rely on an existing standard, Pahl and Aubet [PA18] use a middleware providing similar feature by communicating predefined classes attached with each device's requests.

Fan *et al.* [Fan+20a] leverage FTL to enable model specialization. Each client trains a personalized version of the global model using Transfer Learning (TL). This allows to train models accurately on the singularities of each network, while improving the overall performance of the system. Another approach is to use MTL to train a single model on multiple tasks. Zhao *et al.* [Zha+19] train a unique model for anomaly detection, Virtual Private Network (VPN) traffic classification, and TOR traffic recognition. Qin and Kondo [QK21] propose another way of building different more specific models, by training models depending on the feature set used by the local device. They emit the hypothesis of building models per attack: devices could train a model for *DoS* attacks, others for *Probes*. The other works considered in this survey use a global model for their detection [Zha+20a; ST19; Li+20a; CZY20; Rah+20; ACA20; Kim+20; Qin+20a; Che+20; Hei+20; Li+20a; Pop+21a], regardless of the data type or detection method.

Analyzed Dataset

The literature on IDS has produced various datasets over the years, which are used to evaluate the performance of the proposed systems. Common datasets include KDD'99 [Sig99] and its fixed version NSL-KDD [Tav+09], UNSW-NB15 [MS15], and CICIDS2017 [SHG18]. We will not describe these datasets here, as Chapter 2 already provides an overview of the most common datasets used in intrusion detection. However, it is worth stressing again that these datasets are often criticized, either for their age, their lack of realism, or the different biases they contain. For instance, NSL-KDD fixes multiple issues of the original KDD'99 dataset, such as removing redundant and duplicated records. Likewise, recent works [Lan+22; ERJ21] have demonstrated issues in the CICIDS2017 dataset, *e.g.*, duplicated records, ineffective attacks, and misordered packets.

The initial selection highlights a consequent representation of said datasets: out of the 22 selected works, 3 use KDD'99, 6 use NSL-KDD, and 4 use CIC-IDS2017. There are still some overlaps between the datasets, as some works [Zha+19; Fan+20a; Qin+20b; Che+20] test their approach on multiple datasets (but only one at a time). The other works generally use domain-specific datasets, such as the MIMIC-III dataset [Joh+16] for health-related data [ST19], or the CPS dataset of Morris and Gao [MG14] for FIDSs in industrial settings [Li+20a].

However, this selection highlights a massive drawback of the literature: biased assumptions on the data distribution. Except for Sun, Ochiai, and Esaki who used a dataset collecting data from effectively distributed sources, all selected work use a single dataset and distribute it among the clients. Even worse, most of them either do not mention data-distribution at all, or assume it is IID. As discussed in Section 3.4.2, this is a major drawback, as most real-world scenarios generate non-IID data. This makes most experiments unrealistic, as the clients train their models on data generated on the same network topology, with the same devices, and the same behaviors, even considering Non Independent and Identically Distributed (NIID) settings.

Just after we stopped data collection for the original study, **sarhan_standardfeatureset_2021** proposed a standardized feature set for intrusion detection (see Chapter 2), and converted four known IDS datasets to this format: UNSW-NB15 [MS15], Bot-IoT [Kor+19], ToN_IoT [Mou+20], and CSE-CIC-IDS2018 [SHG18]. The uniform feature set across datasets allows FL-based approaches to evaluate their performance on independently generated datasets [**de_carvalho_bertoli_generalizing_2023**; Pop+21b], closing the gap towards more realistic experiments. In the context of cross-silo FL, each dataset can act as one organization's collected data, which is done by **de_carvalho_bertoli_generalizing_20**

Costs and Metrics

We can divide the metrics used in the selected works into three categories that follow the life cycle of the system: training, federation, and execution. Training-related metrics measure the behavior of the model during the training phase. Federation-related assess the costs and benefits of the FL approach, while execution-related metrics measure the performance of the system in real-time, notably during the detection or classification phase.

Training This includes typical metrics like accuracy and loss used during the training phase, as well as resource-related metrics. They can be used to measure the convergence time of the model, often characterized as obtaining an accuracy above a defined threshold (*e.g.* 90% in [CZY20]), or with the percentage of loss improvement between two epochs (*e.g.* 0.01 in [Kim+20]). Training time also serve as a comparison between approaches [ST19], even though it depends a lot on the underlying hardware architecture. Finally, it can be used as a metric to select other hyper-parameters, such as the number of epochs in [Liu+21]. Algorithm complexity and resource consumption are also relevant metrics to measure local training costs. Constrained use cases like IoT require complex algorithm to run on resource-limited devices. In [PA18], the authors also study complexity to choose BIRCH clusters instead of K-means, as updating the former is easier—$\mathcal{O}(d)$ vs $\mathcal{O}(n * d)$, where d is the dataset size.Hardware-related resources are used by [RWP19; Zha+19], mostly to emphasize differences between their approach and another, often more standard one. These resources often include CPU, disk and memory usage, as well as energy consumption. However, evaluating hardware-related metrics requires experiments to be implemented using the same hardware and software stacks. Hardware- and energy-based metrics are especially relevant in constrained scenarios [Ngu+19; ST19], whereas training time is relevant for most use case, while not a priority. When these measures are collected on reference hardware, it can also be used to evaluate the feasibility of the approach, as in [Ngu+19], if the hardware matches the deployment constraints of the study.

Federation Federation-related metrics are heavily tied to the communication between clients, or with a server. The communication overhead is a core metric of FIDSs, as high bandwidth consumption is a drawback of CIDSs (see **??**), especially in constrained environments [Qin+20a]. The overhead is often measured in bytes, either per packets [PA18], or for the total of all communications [ST19; Zha+20a]. Metrics must be adapted to the specificities of each solution, for instance when adding a feature. Consequently, Zhang, Lu, *et al.* [Zha+20a] add specific metrics in their evaluation to measure the impact of using the blockchain, like the time of the *blockchain encoring process*. Some works [RWP19; Li+20a; CZY20; Fan+20a; Rah+20; ACA20; Pop+21a; SOE20], on the other hand, do not cover federation-related metrics in their evaluation, which is questionable as it is a critical part of FIDSs.

Execution Finally, execution-related metrics are mostly focused on performance, and often come from the ML community. As shown in Table 3.2, *accuracy* is used by almost all reviewed works, followed by *precision* and *recall*. More generally, all metrics issued from the confusion matrix can be used, but the literature emphasizes on metrics that focus on the detection of anomalies, like *recall* and *precision*, or the *F1-score* which combines the two. Researchers often use these metrics to compare their results with related works. Other execution metrics like execution time are considered, as it can be critical for intrusion detection tasks. Latency allows a comparison between different architectures, especially *centralized*, *distributed*, and *decentralized* [RWP19]. Latency is also relevant for highly constrained setups, as in [Qin+20a]. As pointed out in Section 3.4.2, *ML location* can have an impact on data collection, but also on detection latency, if data need to travel over network to be analyzed. Execution metrics are only relevant when comparing works that share implementation. Such comparison is often performed by reimplementing a selection of related works. They can also be used to highlight differences between approaches, like between *local*, *federated*, and *ideal* models [Li+20a; RWP19].

3.5 Related Work

At the time of writing this literature review, the literature on FL for IDS was still scarce. Only a handful of reviews had been published on the topic [Ala+21; Agr+22; Cam+22]. Therefore, we extended our search of related works to related topics that were susceptible to share similar challenges or conclusions. This extended selection can be divided into three main categories: (a) security information sharing, (b) intrusion detection, and (c) collaborative ML. Table 3.3 provides a summary of this selection, grouped by topic and sorted by publication date. In addition to the initial selection, we also included more recent surveys on the topic [FNS22; GR22; Ism+24], whose number highlights the massive interest in the community.

Table 3.3 – *Related literature reviews, their topics, contributions, and number of citations according to Google Scholar (Apr. 2024).* Works marked $*$ were originally available as preprints, and were only published afterward. Works marked ‡ are added for the sake of completeness, but were not included in the initial selection.

Domain	**Year**	**Authors**	**Contributions**							**Cited**	**Ref.**
			Qualitative analysis	Quantitative analysis	Taxonomy	Reference architecture	Performance evaluation	Research directions	Systematic Literature Review		
Security information sharing	2016	Skopik *et al.*	●	○	○	○	○	●	○	291	[SSF16]
	2018	Tounsi *et al.*	●	●	○	○	○	●	○	448	[TR18]
	2019	Wagner *et al.*	●	●	○	○	○	●	○	240	[Wag+19]
	2019	Pala *et al.*	●	●	○	●	○	●	○	63	[PZ19]
ML for intrusion detection	2016	Buczak *et al.*	●	○	○	○	○	◐	○	3105	[BG16]
	2018	Meng *et al.*	●	○	○	○	○	●	○	562	[Men+18]
	2019	Chaabouni *et al.*	●	○	◐	○	○	●	○	790	[Cha+19]
	2019	da Costa *et al.*	●	○	○	○	○	●	○	492	[dCos+19]
Collaborative detection	2010	Zhou *et al.*	●	○	○	○	○	●	○	517	[ZLK10]
	2015	Vasilomanolakis *et al.*	●	○	●	○	○	●	○	379	[VKF15]
Federated learning	2020	Aledhari *et al.*	●	○	○	○	○	○	○	517	[Ale+20]
	2020	Lyu *et al.* $*$	●	○	○	○	○	●	○	436	[LYY20]
	2020	Shen *et al.*	●	○	○	○	○	●	○	69	[She+20]
	2021	Mothukuri *et al.*	●	○	●	○	○	●	○	376	[Mot+21a]
	2021	Lo *et al.*	●	●	○	○	○	●	●	158	[Lo+21]
FL for intrusion detection	2021	Agrawal *et al.* $*$	●	○	○	○	○	●	○	142	[Agr+22]
	2021	Alazab *et al.*	●	○	○	○	○	●	○	158	[Ala+21]
	2021	Campos *et al.* $*$	●	○	○	○	●	●	○	123	[Cam+22]
	2022	**Lavaur *et al.***	●	●	●	●	○	●	●	22	[Lav+22b]
	2022	Fedorchenko *et al.* ‡	◐	○	○	○	○	○	○	22	[FNS22]
	2022	Ghimire *et al.* ‡	●	○	○	○	○	●	○	208	[GR22]
	2024	Isma'ila *et al.* ‡	●	●	○	○	○	●	●	0	[Ism+24]

● covers topic; ◐ partly addresses topic; ○ does not cover topic.

Common issues of collaborative systems, such as the need for trust, privacy, and security, can also apply to FL-based collaboration systems. Therefore, we include four surveys [SSF16; TR18; Wag+19; PZ19] where the authors discuss the challenges and opportunities of sharing security-related information. They highlight the need for standardization, automation, and incentives, to achieve efficient and effective collaboration. The topic of trust is a clearly identified challenge in these works [Wag19; TR18]. The present study rather focuses on FL as a technical mean for collaboration, but such as trust or incentives are also relevant in this context.

Because ML-based IDS can be considered as a key component of FIDS, we review existing surveys on the topic [BG16; Men+18; Cha+19; dCos+19]. These work cover a wide range of solutions, from traditional ML (Support Vector Machine (SVM), Decision Tree (DT) and Random Forest (RF), among others) to more recent approaches, such as deep learning, the latter being overrepresented in the literature of FIDSs. They also provide a good overview of the existing datasets and evaluation metrics, which can be useful for the evaluation of FL-based IDS. However, as noted in Section 3.6.2, typical IDS datasets present limitations that can hinder the evaluation of FL-based IDS.

FL is obviously another critical aspect of FIDSs. Consequently, related works include surveys on the collaborative aspects of ML (b) and FL [Ale+20; Lo+21]. They discuss FL approaches to work with distributed architectures. The security of FL is also heavily reviewed by [She+20; LYY20; Mot+21b]. They identify security threats like communication bottleneck, poisoning, and Distributed Denial of Service (DDoS) attacks, that could endanger FL-based systems. While the IDS use case can be seen as an application of FL, we argue that it raises specific concerns in terms of privacy, latency, and adaptability.

Zhou, Leckie, and Karunasekera [ZLK10] and Vasilomanolakis, Karuppayah, and Fischer [VKF15] survey the evolution of CIDS—at the merge of intrusion detection (b) and collaborative ML (c). Their works are however older and thus, cannot offer a comprehensive view of CIDS, as FL-based approaches did not exist at the time of their writing. Hence, the authors focus on collaboration in the sense of *detection+correlation*, whereas the analysis presented in this chapter (Section 3.4) surveys the use of FL in IDSs.

In addition to the above, recent work (*i.e.*, contemporary to the writing of the initial study) have reviewed the use of FL for intrusion detection [Agr+22; Cam+22; Ala+21]. Alazab *et al.* [Ala+21] address the wider topic of FL for cybersecurity, which only includes intrusion detection as an application. Their paper is explanatory and provides an overview of FL applications in information security. Like this work, Agrawal *et al.* [Agr+22] focus on FIDSs, but have different methodology. The authors list existing FIDSs and detail their approaches, and identify open issues. On the other hand, Campos *et al.* [Cam+22] review a subset of FIDSs by focusing on IoT use case, and the impact of non-IID (Independent and Identically Distributed) data on performance. While all identify challenges and research directions, this work also performs quantitative (Section 3.3) and qualitative

(Section 3.4) analyses of existing FIDSs, and extracts reference architecture and taxonomy. The existence of these papers emphasizes the importance and relevance of FIDSs for the research community.

The more recent works on the topic [FNS22; GR22; Ism+24] confirm these observations. The work of Fedorchenko, Novikova, and Shulepov [FNS22] is of little interest, as it only lists and details existing works with close to no added value. Ghimire and Rawat [GR22] provide a more convincing study, closer to the method applied by Alazab *et al.* [Ala+21], but with a focus on the IoT. Finally, Isma'ila *et al.* [Ism+24] provide a comprehensive review, with up-to-date literature leveraging the SLR methodology, but still focuses on the IoT.

3.6 Discussion

This section first discusses the limitations of this study, notably the number of selected papers and the methodology used. We then answer Question RQ1-3.c by identifying the open issues and according research directions, and associate them with recent publications.

3.6.1 Limitations of this Study

The original review reviewed 22 technical papers about FIDSs, selected using SLR methodology. This ensures that the selected papers are representative of existing works in this field. Other surveys in similar but broader fields worked with bigger quantities of papers; 231 in [Lo+21] about FL, or 95 in [dCos+19] for ML-based IDSs. Therefore, all conclusion extracted from the selected works must be put in perspective of the number of analyzed papers.

Furthermore, SLR methodology guarantees the exhaustive aspect of the selection. However, relevant papers may have been missed; especially, edge–use-cases and unusual wording can exclude papers from the selection process. We expect the steps presented in Section 3.2 to mitigate this risk, notably snowballing.

Moreover, the selected metrics give insight on the quality of the predictions, and more importantly the comparison between FIDS and local detection, when provided. As the selected works target different use cases with different objectives, a performance metric-based comparison is less relevant. Using the same datasets, hardware and network configuration, and coding frameworks, a thorough reimplementation of the reviewed papers could provide significant contributions.

Finally, the selected papers are from 2016 to 2021, and the field of FIDS has been evolving rapidly. As noted in Section 3.2.3, a significant number of papers were published since the end of data collection of the original review, including new literature reviews (see Section 3.5). Therefore, the conclusions of this study may not be up-to-date. However, the conclusions of this study have provided a solid foundation for the other contributions of this thesis, and most identified trends and research directions are still relevant.

3.6.2 Open Issues and Future Directions

As FL is becoming more mature, new research tend focus either on side-aspects like security and privacy [Bon+17; FYB20; Don+20; Ngu+20; Mot+21b] or on its application to a specific use case, as do the works selected in this survey. This section reviews open questions identified by literature, and the proposed according research directions. Additionally, for each identified open issue, we provide relevant publications that have been published since the original review to complement the discussion. Some of these issues depend on works from other related fields, such as ML for performance or FL for scalability. However, the specificities of FIDSs require dedicated research. Especially, the topics of security, trust, and heterogeneity are critical for a collaborative security use case.

Performance Like any detection system, FIDSs are looking for an absolute performance: a system with a perfect classification score, producing no false positives or negatives. To this end, several research directions have been identified by the selected works, such as the use of Generative Adversarial Networkss (GANs) [ST19] or the improvement of feature selection as input to the model [SEO21]. More generally, there is a need for a better understanding of the impact of hyper- and meta-parameters on the performance of the system. This is especially true for FL, where the aggregation process can be seen as an optimization problem in itself [CK21]; a problem for which the right parameters need to be inferred. Both GANs [Jin+24] and meta-learning for local data-sampling [HDH23] have been reviewed as potential solutions to this problem.

Adaptability Constrained environments like low-bandwidth networks, or low-powered devices, may also impact the ability of FL to provide detection in a timely fashion (Chapter 2). Since the security of constrained devices is a growing concern, the selected works identify relevant research directions in this area, such as implementing compression algorithms [Fan+20a] or globally reducing the number of computation rounds [Rah+20]. Moreover, as time goes by, the training data can be easily become outdated. Updating strategies need to be studied to provide accurate results as time goes [Fan+20a], and adapt to changes in the traffic behavior [QK21]. This topic has been especially tackled in FIDSs using incremental learning [Jin+23].

Scalability Distributed systems such as FL are often used to cope with resource limitations, especially in terms of computation and bandwidth. However, as pointed out by several selected works, FL faces limitations when dealing with too many clients [RWP19;

Fan+20a]. Therefore, FIDSs require further research regarding client selection: performance-, time-, or reputation-based [Cun+24]. Moreover, in massively distributed federations, the aggregation process can become a bottleneck. In such settings, researchers and practitioners might consider using hierarchical aggregation or even complete decentralization of the system. A few decentralized FIDSs approaches have been proposed since [Fri+23].

Heterogeneity and Transferability The approaches presented in the initial review mostly consider that all local models share the same architecture and hyper-parameters, use data from the same domain, and that all clients possess similar resources. These limitations hurdle convergence, and more generally make current FIDSs less versatile and transferable. Hence, open issues include allowing the federation of cross-domain clients [Li+20a]. As pointed out in Section 3.4.2, the features selected for model training have to be applicable to multiple environments. Transfer learning [SA21; She+20] and its federated variant FTL [Che+20; Fan+20a] have been applied to similar domains in the past, and might also represent a favorable direction for future research in terms of adaptability. Since the submission of this study, multiple papers [OWN21; Kho+21; Che+22] have been published in this direction.

Security and Privacy The broad attack surface of FL directly applies to FIDSs, raising concerns about poisoning, inference, or model extraction. The selected works already address some of these issues by leveraging homomorphic encryption to secure the aggregation process [Li+20a; Li+20b]. Others identify this aspect as potential future works [Che+20], with countermeasures like MPC or Differential Privacy (DP). Furthermore, as ML, and especially DL, lacks explainability, the content of a model is difficult to infer. It complicates the detection of poisoning attacks, as it is hard to distinguish between a model that has been poisoned and one that has been trained on a different dataset. Poisoning has recieved a lot of attention in the literature of FL [FYB20; Ngu+20; Ngu+22], and some works have been published in the context of intrusion detection too [Yan+23; Mer+23].

Trust and Reputation Following the same line of thought, the trustworthiness of the participants is a critical aspect of FIDSs. Malicious participants can indeed impact the model and the detection process. More generally, the quality of the participants' contributions must be controlled to ensure the quality of the aggregated model. Zhang, Lu, *et al.* [Zha+20a] identify assessing the trustworthiness of the participants as a future research direction. Inspiration should be taken from the state of the art of collaboration systems and information-sharing platforms, which address problems such as trust or reputations [Wag+19; SSF16], which are relevant for FIDS. Since the submission of this study, works have been published on the topic of trust via client selection [Cun+24].

Self-defense and self-healing As highlighted in Section 3.4.2, current research on FIDS is focused on intrusion detection and attack classification. Mitigation is barely represented in the literature [RWP19]. However, technologies like SDN offer quick mitigation capabilities, and recent works study the effectiveness of such defense mechanisms [BG17; SB20]. New emerging applications like self-defense and self-healing systems could benefit from FIDS and other FL-based technologies. A handful of works have been published on the topic of attack mitigation and reaction [Pan+22; dCal+23; PG22], corroborating our survey's findings.

Evaluation Finally, the topic of evaluation raise two major issues in the selected works. First, reproducibility is a major concern, as few are the works that provide the code or the datasets used for the experiments. This is a common issue in the field of ML, which has long been criticized for its lack of reproducibility [Baj+17; Arp+22]. The same issue is present in the field of FIDSs, as most of the selected works do not provide enough information to reproduce the experiments. Some do not even disclose the datasets they used, such as Nguyen, Marchal, *et al.* for DÏoT [Ngu+19]. Further, existing public datasets are not representative of FIDSs deployment environments. Indeed, they are often datasets produced for traditional ML-based IDSs, but split for federated purposes. This makes most of the literature biased, as all samples are related to the same original event. Recent works have tried to partially address this issue by providing standardized datasets [SLP22] or dedicated ones [Fer+22], although the problem remains unsolved.

3.7 Conclusion and takeaways

FL comes solves two main challenges: (1) it breaks isolated architectures by allowing learning over distributed data without compromising privacy; and (2) it speeds up training and reduces communications compared to existing distributed learning approaches, and even more so when compared to centralized learning. Applied on intrusion detection, FL allows to leverage the knowledge of multiple actors to improve the detection of attacks, while preserving the privacy of each organization's data. This is particularly relevant to fit with the injunctions of security agencies and regulations, which call for collaboration and intelligence sharing, while also demanding strong privacy requirements. Based on the literature reviewed, we can define FIDSs as follows:

Definition 3.1: Federated Intrusion Detection Systems (FIDSs)

Distributed IDSs with privacy-preserving federated knowledge. FIDSs leverage FL or similar distributed learning techniques [1] to share and aggregate the models trained locally with other members of the federation. Federations can be closed (*i.e.*, all participants are identified and trusted) or open (*i.e.*, participants can join and leave the federation at any time). Depending on the ML model used locally, training can happen offline on labelled data, online, or with a combination of both.

This review highlighted eight main challenges that need to be addressed to build efficient and secure FIDSs, ranging from pure performance to scalability and mitigation mechanisms. In the following chapters, we will especially focus on three of these challenges: (i) *FIDSs in Heterogeneous Environments*; (ii) *Malicious Contributions and Trust*; and (iii) *Evaluation and Dataset Representativity.* More specifically, we will address the following points:

— We address *Heterogeneity* and *Dataset Representativity* in Chapter 5, where we propose a novel approach to generate network topologies that are both realistic and heterogeneous.

— Chapter 6 reviews the impact of *Malicious Contributions* over FIDSs in IID settings, with an emphasis on reproducibility.

— We propose in Chapter 7 a novel approach to mitigate such effects in *heterogeneous* environments, leveraging *reputation* systems to assess the quality of the participants' contributions and their *trustworthiness.*

1. We take some liberty in this definition by not imposing the use of FL as a requirement. While it is definitely the most popular approach and the motivation behind this study, other privacy-preserving distributed learning techniques have been used in the literature [PA18; RWP19]. Further, the formal definition of FL is still debated (see Chapter 2), as the term is often used to describe a broader set of techniques.

Chapter 4

~~An Overview of~~ FIDSs' Performance and Limitations

of FIDSs

C'est une proposition pour éviter le "FIDSs'". A toi de voir.

4.1 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 4.2 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].

4.2 A Practical Use Case for FIDSs

Les ensembles sont en majuscules.

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 $p, k \in [\![1, n]\!]$ at each round r. Participants 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.~~0~~.

C'est ce que tu avais mis (cf. p 24)

P_k

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, such as the 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.

Preciser que, par contre, les données peuvent être particulièrement hétérogènes dans ce cas (pour justifier l'aspect réaliste)

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 possess es a local dataset $d_k = (X_i, Y_i)$ that is not shared with the others. Because organizations in CIDS may have different network configurations [ZLK10], the distribution of each local dataset d_k can vary considerably, independently of the associated labels. 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].

4.3 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 4.3.2 to 4.3.4) target different heterogeneity scenarios, ranging from homogeneous dataset partitioning to completely independent data sources. The last scenario (Section 4.3.5) focuses on poisoning attacks against FL, where malicious participants try to degrade the performance of the global model.

4.3.1 Setup

To evaluate the performance of FL in the context of CIDSs, and especially evaluate the feasibility of the scenario presented in Section 4.2, we need datasets that are representative of the traffic that can be observed in real-world networks. 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 Section 2.2.2). 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.

Rajouter la ref vers le papier demo@ICDCS

To generate the different scenarios, we build an evaluation framework for FL called Eiffel [1], which relies on Flower [Beu+20], a modular FL framework. Eiffel is a 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 locally, 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 4.1 summarizes the parameters used for all scenarios, with the notations defined in Section 2.4.

only on local data

4.3.2 Scenario 1: IID Data

which ones? Ça semble être presque moins bon.

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 4.1 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 few benefits to using FL in this scenario.

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

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

Table 4.1 – 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

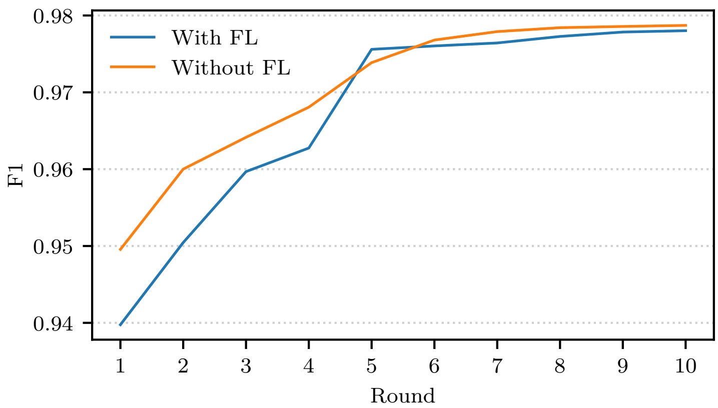


Figure 4.1 – Global model performance in IID.

Ça-ai ;-)

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.

4.3.3 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 4.3.2, 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 4.2 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

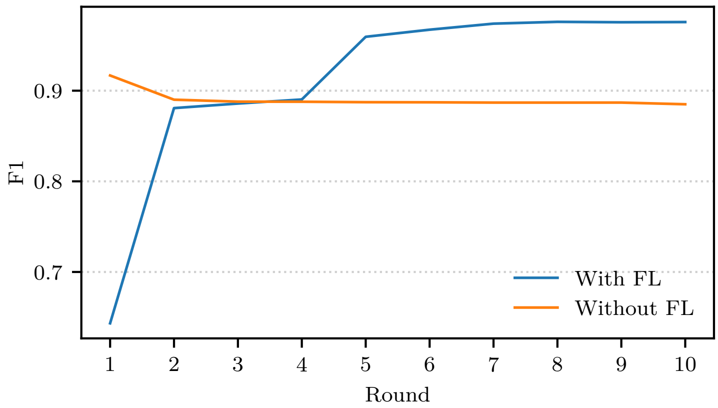


Figure 4.2 – Global model performance in NIID (same source).

Table 4.2 – 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

On avait pas de point de comparaison avec l'algo sur le dataset complet ? Car 20% ça fait un peu faible. (dû à la rareté de l'attaque)

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 4.2 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.

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.

4.3.4 Scenario 3: NIID Data from Different Sources

While we highlight in Section 4.3.3 that FL can benefit from having different datasets per client, to the point where it can share knowledge between participants, the third

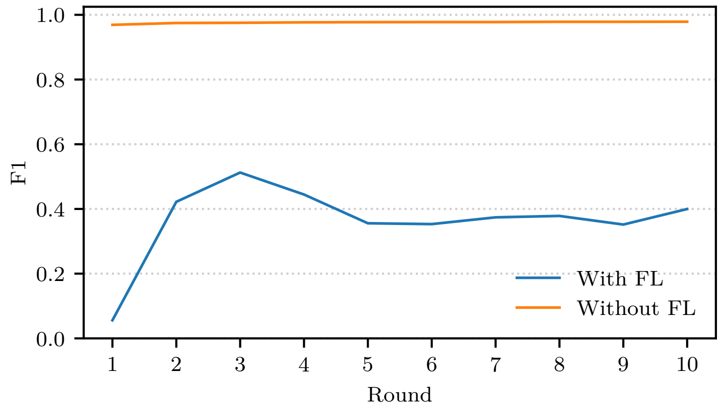


Figure 4.3 – Global model performance in NIID (different sources).

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 4.3.2 and 4.3.3, 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 4.3.1), 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.

The results presented in Figure 4.3 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, Gui, *et al.* [Pop+21b] for instance.

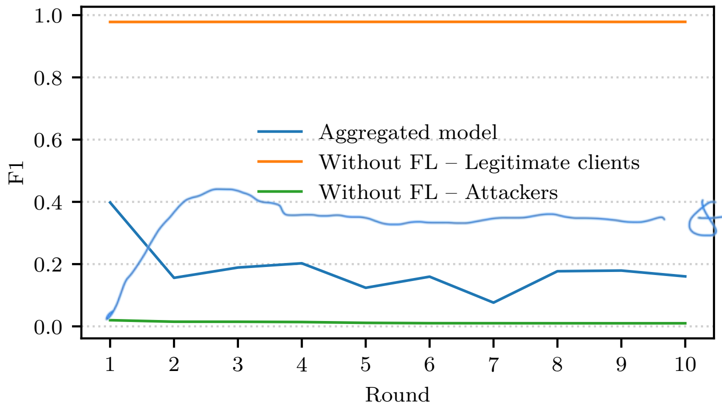
Tu n'expliques pas, en 2.3 mots, leur approche ?

4.3.5 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 4.4) that participants identified as



Remet la courbe de la Fig. 4.3 pour comparer avec et sans attaque ?

Figure 4.4 – Global model performance in poisoning attacks.

"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 4.3.2, with a high DR on their test set.

In FL however, the global model is impacted by the malicious participants, as illustrated in Figure 4.4. The participants cannot converge towards a stable global model, as the malicious participants' updates are too different from the others. Due to the missclassification 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 means that the aggregated model either misses a lot of attacks and misclassifies a lot of benign samples.

4.4 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 Federated Intrusion Detection Systems (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 Chapter 5, and the impact of poisoning attacks in Chapter 6. Finally, we will present some solutions to these limitations in Chapter 7 and Chapter 8.

Part II

Quantifying the Limitations of FIDSs

Chapter 5

Studying Heterogeneity in Distributed Intrusion Detection with Topology Generation

Chapter 6

Assessing the Impact of Label-Flipping Attacks on FL-based IDSs

Part III

Providing Solutions

Chapter 7

Model Quality Assessment for Reputation-aware Collaborative Federated Learning

Chapter 8

Solutions for the Future of FIDSs

Chapter 9

Conclusion

Bibliography

[16] *Directive (EU) 2016/1148 of 6 July 2016 Concerning Measures for a High Common Level of Security of Network and Information Systems across the Union*, 2016, URL: https://eur-lex.europa.eu/eli/dir/2016/1148/oj.

[22] *Directive (EU) 2022/2555 of the European Parliament and of the Council of 14 December 2022 on Measures for a High Common Level of Cybersecurity across the Union, Amending Regulation (EU) No 910/2014 and Directive (EU) 2018/1972, and Repealing Directive (EU) 2016/1148 (NIS 2 Directive)*, Dec. 14, 2022, URL: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32022L2555 (visited on 05/26/2024).

[AAA16] Iman Almomani, Bassam Al-Kasasbeh, and Mousa AL-Akhras, « WSN-DS: A Dataset for Intrusion Detection Systems in Wireless Sensor Networks », *in*: *Journal of Sensors* 2016 (2016), pp. 1–16, ISSN: 1687-725X, 1687-7268, DOI: 10.1155/2016/4731953, URL: https://www.hindawi.com/journals/js/2016/4731953/ (visited on 10/25/2021).

[ACA20] Noor Ali Al-Athba Al-Marri, Bekir S. Ciftler, and Mohamed M. Abdallah, « Federated Mimic Learning for Privacy Preserving Intrusion Detection », *in*: *2020 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, 2020 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom), May 2020, pp. 1–6, DOI: 10.1109/BlackSeaCom48709.2020.9234959, URL: https://ieeexplore.ieee.org/document/9234959 (visited on 04/12/2024).

[Agr+22] Shaashwat Agrawal *et al.*, « Federated Learning for Intrusion Detection System: Concepts, Challenges and Future Directions », *in*: *Computer Communications* 195 (Nov. 1, 2022), pp. 346–361, ISSN: 0140-3664, DOI: 10.1016/j.comcom.2022.09.012, URL: https://www.sciencedirect.com/science/article/pii/S0140366422003516 (visited on 06/09/2024).

[Ala+21] Mamoun Alazab *et al.*, « Federated Learning for Cybersecurity: Concepts, Challenges and Future Directions », *in*: *IEEE Transactions on Industrial Informatics* (2021), pp. 1–1, ISSN: 1551-3203, 1941-0050, DOI: 10/gnm4dj, URL: https://ieeexplore.ieee.org/document/9566732/ (visited on 12/02/2021).

[Ale+20] Mohammed Aledhari *et al.*, « Federated Learning: A Survey on Enabling Technologies, Protocols, and Applications », *in*: *IEEE Access* 8 (2020), pp. 140699–140725, ISSN: 2169-3536, DOI: 10.1109/ACCESS.2020.3013541, URL: https://ieeexplore.ieee.org/document/9153560/.

[Ali+18] Janne Ali-Tolppa *et al.*, « SELF-HEALING AND RESILIENCE IN FUTURE 5G COGNITIVE AUTONOMOUS NETWORKS », *in*: *2018 ITU Kaleidoscope: Machine Learning for a 5G Future (ITU K)*, 2018 ITU Kaleidoscope: Machine Learning for a 5G Future (ITU K), Santa Fe: IEEE, Nov. 2018, pp. 1–8, ISBN: 978-92-61-26921-0, DOI: 10.23919/ITU-WT.2018.8598115, URL: https://ieeexplore.ieee.org/document/8598115/ (visited on 03/31/2022).

[And+18] Elli Androulaki *et al.*, « Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains », *in*: *Proceedings of the Thirteenth EuroSys Conference*, New York, NY, USA: ACM, Apr. 23, 2018, pp. 1–15, ISBN: 978-1-4503-5584-1, DOI: 10.1145/3190508.3190538, URL: https://dl.acm.org/doi/10.1145/3190508.3190538.

[Aou+22] Ons Aouedi, Kandaraj Piamrat, Guillaume Muller, *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).

[APB20] Ons Aouedi, Kandaraj Piamrat, and Dhruvjyoti Bagadthey, « A Semi-supervised Stacked Autoencoder Approach for Network Traffic Classification », *in*: *2020 IEEE 28th International Conference on Network Protocols (ICNP)*, 2020 IEEE 28th International Conference on Network Protocols (ICNP), Oct. 2020, pp. 1–6, DOI: 10.1109/ICNP49622.2020.9259390, URL: https://ieeexplore.ieee.org/document/9259390 (visited on 06/19/2024).

[Arp+22] Daniel Arp *et al.*, « Dos and Don'ts of Machine Learning in Computer Security », *in*: *31st USENIX Security Symposium (USENIX Security 22)*, Boston, MA: USENIX Association, Aug. 2022, pp. 3971–3988, ISBN: 978-1-939133-31-1, URL: https://www.usenix.org/conference/usenixsecurity22/presentation/arp.

[Baj+17] Vaibhav Bajpai *et al.*, « Challenges with Reproducibility », *in*: *Proceedings of the Reproducibility Workshop*, SIGCOMM '17: ACM SIGCOMM 2017 Conference, Los Angeles CA USA: ACM, Aug. 11, 2017, pp. 1–4, ISBN: 978-1-4503-5060-0, DOI: 10.1145/3097766.3097767, URL: https://dl.acm.org/doi/10.1145/3097766.3097767 (visited on 08/12/2022).

[Ber+18] Jeremy Bernstein *et al.*, « signSGD: Compressed Optimisation for Non-Convex Problems », *in*: *Proceedings of the 35th International Conference on Machine Learning*, ed. by Jennifer Dy and Andreas Krause, vol. 80, Proceedings of Machine Learning Research, PMLR, July 10–15, 2018, pp. 560–569, URL: https://proceedings.mlr.press/v80/bernstein18a.html.

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

[BG16] Anna L. Buczak and Erhan Guven, « A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection », *in*: *IEEE Communications Surveys Tutorials* 18.*2* (2016), pp. 1153–1176, ISSN: 1553-877X, DOI: 10.1109/COMST.2015.2494502.

[BG17] Suman Sankar Bhunia and Mohan Gurusamy, « Dynamic Attack Detection and Mitigation in IoT Using SDN », *in*: *2017 27th International Telecommunication Networks and Applications Conference (ITNAC)*, 2017 27th International Telecommunication Networks and Applications Conference (ITNAC), Melbourne, VIC: IEEE, Nov. 2017, pp. 1–6, ISBN: 978-1-5090-6796-1, DOI: 10.1109/ATNAC.2017.8215418, URL: http://ieeexplore.ieee.org/document/8215418/ (visited on 04/09/2022).

[Bha+19] Arjun Nitin Bhagoji *et al.*, « Analyzing Federated Learning through an Adversarial Lens », *in*: *Proceedings of the 36th International Conference on Machine Learning*, International Conference on Machine Learning, PMLR, May 24, 2019, pp. 634–643, URL: https://proceedings.mlr.press/v97/bhagoji19a.html (visited on 02/23/2023).

[Big+14] Elaheh Biglar Beigi *et al.*, « Towards Effective Feature Selection in Machine Learning-Based Botnet Detection Approaches », *in*: *2014 IEEE Conference on Communications and Network Security*, 2014 IEEE Conference on Communications and Network Security (CNS), San Francisco, CA, USA: IEEE, Oct. 2014, pp. 247–255, ISBN: 978-1-4799-5890-0, DOI: 10.1109/CNS.2014.6997492, URL: https://ieeexplore.ieee.org/document/6997492 (visited on 10/25/2021).

[Bon+17] Keith Bonawitz, Vladimir Ivanov, *et al.*, « Practical Secure Aggregation for Privacy-Preserving Machine Learning », *in*: *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, New York, NY, USA: ACM, Oct. 30, 2017, pp. 1175–1191, ISBN: 978-1-4503-4946-8, DOI: 10.1145/3133956.3133982, URL: https://dl.acm.org/doi/10.1145/3133956.3133982.

[Bon+19] Keith Bonawitz, Hubert Eichner, *et al.*, « Towards Federated Learning at Scale: System Design », *in*: *arXiv* (Feb. 4, 2019), URL: http://arxiv.org/abs/1902.01046.

[Cai+22] Luxin Cai *et al.*, « Cluster-Based Federated Learning Framework for Intrusion Detection », *in*: *2022 IEEE 13th International Symposium on Parallel Architectures, Algorithms and Programming (PAAP)*, 2022 IEEE 13th International Symposium on Parallel Architectures, Algorithms and Programming (PAAP), Nov. 2022, pp. 1–6, DOI: 10.1109/PAAP56126.2022.10010553, URL: https://ieeexplore.ieee.org/abstract/document/10010553 (visited on 04/12/2024).

[Cam+22] Enrique Mármol Campos *et al.*, « Evaluating Federated Learning for Intrusion Detection in Internet of Things: Review and Challenges », *in*: *Computer Networks* 203 (Feb. 11, 2022), p. 108661, ISSN: 1389-1286, DOI: 10.1016/j.comnet.2021.108661, URL: https://www.sciencedirect.com/science/article/pii/S1389128621005405 (visited on 04/25/2024).

[CBK09] Varun Chandola, Arindam Banerjee, and Vipin Kumar, « Anomaly Detection: A Survey », *in*: *ACM Computing Surveys* 41.*3* (July 2009), pp. 1–58, ISSN: 0360-0300, 1557-7341, DOI: 10.1145/1541880.1541882, URL: https://dl.acm.org/doi/10.1145/1541880.1541882 (visited on 03/20/2022).

[Cet+19] Burak Cetin *et al.*, « Federated Wireless Network Intrusion Detection », *in*: *2019 IEEE International Conference on Big Data (Big Data)*, 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA: IEEE, Dec. 2019, pp. 6004–6006, ISBN: 978-1-72810-858-2, DOI: 10.1109/BigData47090.2019.9005507, URL: https://ieeexplore.ieee.org/document/9005507/ (visited on 10/25/2021).

[Cha+19] Nadia Chaabouni *et al.*, « Network Intrusion Detection for IoT Security Based on Learning Techniques », *in*: *IEEE Communications Surveys & Tutorials* 21.*3* (2019), pp. 2671–2701, ISSN: 1553-877X, DOI: 10.1109/COMST.2019.2896380, URL: https://ieeexplore.ieee.org/document/8629941/.

[Che+20] Zhuo Chen, Na Lv, *et al.*, « Intrusion Detection for Wireless Edge Networks Based on Federated Learning », *in*: *IEEE Access* 8 (2020), pp. 217463–217472, ISSN: 2169-3536, DOI: 10.1109/ACCESS.2020.3041793, URL: https://ieeexplore.ieee.org/document/9274294/ (visited on 10/25/2021).

[Che+22] Yanyu Cheng *et al.*, « Federated Transfer Learning With Client Selection for Intrusion Detection in Mobile Edge Computing », *in*: *IEEE Communications Letters* 26.*3* (Mar. 2022), pp. 552–556, ISSN: 1558-2558, DOI: 10.1109/

LCOMM.2022.3140273, URL: https://ieeexplore.ieee.org/abstract/document/9668958 (visited on 04/12/2024).

[CJ20] Davide Chicco and Giuseppe Jurman, « The Advantages of the Matthews Correlation Coefficient (MCC) over F1 Score and Accuracy in Binary Classification Evaluation », *in*: *BMC Genomics* 21.*1* (Dec. 2020), p. 6, ISSN: 1471-2164, DOI: 10.1186/s12864-019-6413-7, URL: https://bmcgenomics.biomedcentral.com/articles/10.1186/s12864-019-6413-7 (visited on 03/11/2022).

[CK21] Zachary Charles and Jakub Konecny, « Convergence and Accuracy Trade-Offs in Federated Learning and Meta-Learning », *in*: (2021), p. 11.

[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.

[Cun+24] Helio N. Cunha Neto *et al.*, « FedSBS: Federated-Learning Participant-Selection Method for Intrusion Detection Systems », *in*: *Computer Networks* 244 (May 1, 2024), p. 110351, ISSN: 1389-1286, DOI: 10.1016/j.comnet.2024.110351, URL: https://www.sciencedirect.com/science/article/pii/S138912862400183X (visited on 04/12/2024).

[CZY20] Yang Chen, Junzhe Zhang, and Chai Kiat Yeo, « Network Anomaly Detection Using Federated Deep Autoencoding Gaussian Mixture Model », *in*: *Machine Learning for Networking*, ed. by Selma Boumerdassi, Éric Renault, and Paul Mühlethaler, Cham: Springer International Publishing, 2020, pp. 1–14, ISBN: 978-3-030-45778-5.

[DAF18] Rohan Doshi, Noah Apthorpe, and Nick Feamster, « Machine Learning DDoS Detection for Consumer Internet of Things Devices », *in*: *2018 IEEE Security and Privacy Workshops (SPW)* (Ml Apr. 11, 2018), pp. 29–35, DOI: 10.1109/SPW.2018.00013, URL: https://ieeexplore.ieee.org/document/8424629/.

[dCal+23] Francisco Lopes de Caldas Filho *et al.*, « Botnet Detection and Mitigation Model for IoT Networks Using Federated Learning », *in*: *Sensors* 23.*14* (14 Jan. 2023), p. 6305, ISSN: 1424-8220, DOI: 10.3390/s23146305, URL: https://www.mdpi.com/1424-8220/23/14/6305 (visited on 04/12/2024).

[dCos+19] Kelton A.P. da Costa *et al.*, « Internet of Things: A Survey on Machine Learning-Based Intrusion Detection Approaches », *in*: *Computer Networks* 151 (Mar. 2019), pp. 147–157, ISSN: 13891286, DOI: 10.1016/j.comnet.2019.01.023, URL: https://doi.org/10.1016/j.comnet.2019.01.023.

[Den+21] Yongheng Deng *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).

[Don+20] Ye Dong *et al.*, « EaSTFLy: Efficient and Secure Ternary Federated Learning », *in*: *Computers & Security* 94 (July 2020), p. 101824, ISSN: 01674048, DOI: 10.1016/j.cose.2020.101824, URL: https://linkinghub.elsevier.com/retrieve/pii/S0167404820300985 (visited on 05/18/2021).

[Dou02] John R. Douceur, « The Sybil Attack », *in*: *Peer-to-Peer Systems*, ed. by Peter Druschel, Frans Kaashoek, and Antony Rowstron, Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 2002, pp. 251–260, ISBN: 978-3-540-45748-0, DOI: 10.1007/3-540-45748-8_24.

[Dra+16] Gerard Draper-Gil *et al.*, « Characterization of Encrypted and VPN Traffic Using Time-related Features: » *in*: *Proceedings of the 2nd International Conference on Information Systems Security and Privacy*, 2nd International Conference on Information Systems Security and Privacy, Rome, Italy: SCITEPRESS - Science and and Technology Publications, 2016, pp. 407–414, ISBN: 978-989-758-167-0, DOI: 10.5220/0005740704070414, URL: http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0005740704070414 (visited on 10/15/2021).

[Duy+21] Phan The Duy *et al.*, « Federated Learning-Based Intrusion Detection in SDN-enabled IIoT Networks », *in*: *2021 8th NAFOSTED Conference on Information and Computer Science (NICS)*, 2021 8th NAFOSTED Conference on Information and Computer Science (NICS), Dec. 2021, pp. 424–429, DOI: 10.1109/NICS54270.2021.9701525, URL: https://ieeexplore.ieee.org/abstract/document/9701525 (visited on 04/12/2024).

[EA10] Muna Elsadig and Azween Abduallah, « Biological Intrusion Prevention and Self-Healing Model for Network Security », *in*: *2010 Second International Conference on Future Networks*, 2010 Second International Conference on Future Networks (ICFN 2010), Sanya, Hainan: IEEE, Jan. 2010, pp. 337–342, ISBN: 978-1-4244-5666-6 978-0-7695-3940-9, DOI: 10.1109/ICFN.2010.103, URL: https://ieeexplore.ieee.org/document/5431824/ (visited on 03/31/2022).

[ENI14] ENISA, *Actionable Information for Security Incident Response*, 2014, pp. 1–79.

[EO11] Huwaida Tagelsir Elshoush and Izzeldin Mohamed Osman, « Alert Correlation in Collaborative Intelligent Intrusion Detection Systems—A Survey », *in*: *Applied Soft Computing*, Soft Computing for Information System Security 11.7 (Oct. 1, 2011), pp. 4349–4365, ISSN: 1568-4946, DOI: 10.1016/j.asoc.2010.12.004, URL: https://www.sciencedirect.com/science/article/pii/S156849461000311X (visited on 06/22/2024).

[ERJ21] Gints Engelen, Vera Rimmer, and Wouter Joosen, « Troubleshooting an Intrusion Detection Dataset: The CICIDS2017 Case Study », *in*: *2021 IEEE Security and Privacy Workshops (SPW)*, 2021 IEEE Security and Privacy Workshops (SPW), May 2021, pp. 7–12, DOI: 10.1109/SPW53761.2021.00009, URL: https://ieeexplore.ieee.org/document/9474286 (visited on 06/14/2024).

[Fan+20a] Yulin Fan *et al.*, « IoTDefender: A Federated Transfer Learning Intrusion Detection Framework for 5G IoT », *in*: *2020 IEEE 14th International Conference on Big Data Science and Engineering (BigDataSE)*, 2020 IEEE 14th International Conference on Big Data Science and Engineering (BigDataSE), Guangzhou, China: IEEE, Dec. 2020, pp. 88–95, ISBN: 978-1-66540-396-2, DOI: 10.1109/BigDataSE50710.2020.00020, URL: https://ieeexplore.ieee.org/document/9343358/ (visited on 10/04/2021).

[Fan+20b] Minghong Fang *et al.*, « Local Model Poisoning Attacks to Byzantine-Robust Federated Learning », *in*: 29th USENIX Security Symposium (USENIX Security 20), 2020, pp. 1605–1622, ISBN: 978-1-939133-17-5, URL: https://www.usenix.org/conference/usenixsecurity20/presentation/fang (visited on 02/23/2023).

[Far+20] Omair Faraj *et al.*, « Taxonomy and Challenges in Machine Learning-Based Approaches to Detect Attacks in the Internet of Things », *in*: *Proceedings of the 15th International Conference on Availability, Reliability and Security*, ARES 2020: The 15th International Conference on Availability, Reliability and Security, Virtual Event Ireland: ACM, Aug. 25, 2020, pp. 1–10, ISBN: 978-1-4503-8833-7, DOI: 10.1145/3407023.3407048, URL: https://dl.acm.org/doi/10.1145/3407023.3407048 (visited on 06/23/2021).

[Fer+22] Mohamed Amine Ferrag *et al.*, « Edge-IIoTset: A New Comprehensive Realistic Cyber Security Dataset of IoT and IIoT Applications for Centralized and Federated Learning », *in*: *IEEE Access* 10 (2022), pp. 40281–40306, ISSN: 2169-3536, DOI: 10.1109/ACCESS.2022.3165809, URL: https://ieeexplore.ieee.org/document/9751703 (visited on 04/12/2024).

[FNS22] Elena Fedorchenko, Evgenia Novikova, and Anton Shulepov, « Comparative Review of the Intrusion Detection Systems Based on Federated Learning: Advantages and Open Challenges », *in*: *Algorithms* 15.*7* (7 July 2022), p. 247, ISSN: 1999-4893, DOI: 10.3390/a15070247, URL: https://www.mdpi.com/1999-4893/15/7/247 (visited on 04/24/2024).

[FPS10] Gianluigi Folino, Clara Pizzuti, and Giandomenico Spezzano, « An Ensemble-Based Evolutionary Framework for Coping with Distributed Intrusion Detection », *in*: *Genetic Programming and Evolvable Machines* 11.*2* (June 1, 2010), pp. 131–146, ISSN: 1573-7632, DOI: 10.1007/s10710-010-9101-6, URL: https://doi.org/10.1007/s10710-010-9101-6 (visited on 06/23/2024).

[Fri+23] Othmane Friha *et al.*, « 2DF-IDS: Decentralized and Differentially Private Federated Learning-Based Intrusion Detection System for Industrial IoT », *in*: *Computers & Security* 127 (Apr. 1, 2023), p. 103097, ISSN: 0167-4048, DOI: 10.1016/j.cose.2023.103097, URL: https://www.sciencedirect.com/science/article/pii/S016740482300007X (visited on 04/12/2024).

[FS16] Gianluigi Folino and Pietro Sabatino, « Ensemble Based Collaborative and Distributed Intrusion Detection Systems: A Survey », *in*: *Journal of Network and Computer Applications* 66 (May 2016), pp. 1–16, ISSN: 10848045, DOI: 10.1016/j.jnca.2016.03.011, URL: https://linkinghub.elsevier.com/retrieve/pii/S1084804516300248 (visited on 06/22/2024).

[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.

[Gar+09] P. García-Teodoro *et al.*, « Anomaly-Based Network Intrusion Detection: Techniques, Systems and Challenges », *in*: *Computers & Security* 28.*1-2* (Feb. 2009), pp. 18–28, ISSN: 01674048, DOI: 10.1016/j.cose.2008.08.003, URL: https://linkinghub.elsevier.com/retrieve/pii/S0167404808000692.

[Gho+07] Debanjan Ghosh *et al.*, « Self-Healing Systems — Survey and Synthesis », *in*: *Decision Support Systems* 42.*4* (Jan. 2007), pp. 2164–2185, ISSN: 01679236, DOI: 10.1016/j.dss.2006.06.011, URL: https://linkinghub.elsevier.com/retrieve/pii/S0167923606000807 (visited on 03/31/2022).

[GR22] Bimal Ghimire and Danda B. Rawat, « Recent Advances on Federated Learning for Cybersecurity and Cybersecurity for Federated Learning for Internet of Things », *in*: *IEEE Internet of Things Journal* 9.*11* (June 2022), pp. 8229–8249, ISSN: 2327-4662, DOI: 10.1109/JIOT.2022.3150363, URL: https://ieeexplore.ieee.org/abstract/document/9709603 (visited on 04/12/2024).

[Guo+23] Wei Guo *et al.*, « A New Federated Learning Model for Host Intrusion Detection System Under Non-IID Data », *in*: *2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Oct. 2023, pp. 494–500, DOI: 10.1109/SMC53992.2023.10393972, URL: https://ieeexplore.ieee.org/document/10393972 (visited on 06/11/2024).

[Hab+17] Arash Habibi Lashkari *et al.*, « Characterization of Tor Traffic Using Time Based Features: » *in*: *Proceedings of the 3rd International Conference on Information Systems Security and Privacy*, 3rd International Conference on Information Systems Security and Privacy, Porto, Portugal: SCITEPRESS - Science and Technology Publications, 2017, pp. 253–262, ISBN: 978-989-758-209-7, DOI: 10.5220/0006105602530262, URL: http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0006105602530262 (visited on 10/15/2021).

[Hai+01] J W Haines *et al.*, « 1999 DARPA Intrusion Detection Evaluation: Design and Procedures », *in*: (2001), p. 188.

[Han+24] Weixiang Han *et al.*, « Heterogeneous Data-Aware Federated Learning for Intrusion Detection Systems via Meta-Sampling in Artificial Intelligence of Things », *in*: *IEEE Internet of Things Journal* 11.*8* (Apr. 2024), pp. 13340–13354, ISSN: 2327-4662, DOI: 10.1109/JIOT.2023.3337755, URL: https://ieeexplore.ieee.org/abstract/document/10334467 (visited on 04/12/2024).

[Har+17] Stephen Hardy *et al.*, « Private Federated Learning on Vertically Partitioned Data via Entity Resolution and Additively Homomorphic Encryption », Nov. 28, 2017, arXiv: 1711.10677 [cs], URL: http://arxiv.org/abs/1711.10677 (visited on 10/04/2021).

[HC18] David Hand and Peter Christen, « A Note on Using the F-measure for Evaluating Record Linkage Algorithms », *in*: *Statistics and Computing* 28.*3* (May 2018), pp. 539–547, ISSN: 0960-3174, 1573-1375, DOI: 10/gfw6dw, URL: http://link.springer.com/10.1007/s11222-017-9746-6 (visited on 11/10/2021).

[HDH23] Suli He, Chengwen Du, and M. Shamim Hossain, « 6G-enabled Consumer Electronics Device Intrusion Detection with Federated Meta-Learning and Digital Twins in a Meta-Verse Environment », *in*: *IEEE Transactions on Consumer Electronics* (2023), pp. 1–1, ISSN: 1558-4127, DOI: 10.1109/TCE.2023.3321846, URL: https://ieeexplore.ieee.org/abstract/document/10271257 (visited on 04/12/2024).

[Hei+20] Xinhong Hei *et al.*, « A Trusted Feature Aggregator Federated Learning for Distributed Malicious Attack Detection », *in*: *Computers & Security* 99 (Dec. 2020), p. 102033, ISSN: 01674048, DOI: 10.1016/j.cose.2020.102033, URL: https://linkinghub.elsevier.com/retrieve/pii/S0167404820303060 (visited on 10/25/2021).

[Hel+22] Stijn Heldens *et al.*, « Litstudy: A Python Package for Literature Reviews », *in*: *SoftwareX* 20 (Dec. 1, 2022), p. 101207, ISSN: 2352-7110, DOI: 10.1016/j.softx.2022.101207, URL: https://www.sciencedirect.com/science/article/pii/S235271102200125X (visited on 06/06/2024).

[Hid18] Ochiai Hideya, *LAN-Security Monitoring Project*, Whitepaper, 2018, URL: https://lan-security.net/whitepaper.pdf (visited on 10/22/2021).

[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).

[Ism+24] Umar Audi Isma'ila *et al.*, « Review on Approaches of Federated Modeling in Anomaly-Based Intrusion Detection for IoT Devices », *in*: *IEEE Access* 12 (2024), pp. 30941–30961, ISSN: 2169-3536, DOI: 10.1109/ACCESS.2024.3369915, URL: https://ieeexplore.ieee.org/document/10445150 (visited on 04/24/2024).

[Jin+23] Dong Jin, Shuangwu Chen, *et al.*, « Federated Incremental Learning Based Evolvable Intrusion Detection System for Zero-Day Attacks », *in*: *IEEE Network* 37.*1* (Jan. 2023), pp. 125–132, ISSN: 1558-156X, DOI: 10.1109/MNET.018.2200349, URL: https://ieeexplore.ieee.org/abstract/document/10110017 (visited on 04/12/2024).

[Jin+24] Zhigang Jin, Junyi Zhou, *et al.*, « FL-IIDS: A Novel Federated Learning-Based Incremental Intrusion Detection System », *in*: *Future Generation Computer Systems* 151 (Feb. 1, 2024), pp. 57–70, ISSN: 0167-739X, DOI: 10.1016/j.future.2023.09.019, URL: https://www.sciencedirect.com/science/article/pii/S0167739X23003503 (visited on 04/12/2024).

[Joh+16] Alistair E.W. Johnson *et al.*, « MIMIC-III, a Freely Accessible Critical Care Database », *in*: *Scientific Data* 3.*1* (May 24, 2016), p. 160035, ISSN: 2052-4463, DOI: 10.1038/sdata.2016.35, URL: https://doi.org/10.1038/sdata.2016.35.

[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).

[Kar+20] Sai Praneeth Karimireddy *et al.*, « SCAFFOLD: Stochastic Controlled Averaging for Federated Learning », *in*: *Proceedings of the 37th International Conference on Machine Learning*, International Conference on Machine Learning, PMLR, Nov. 21, 2020, pp. 5132–5143, URL: https://proceedings.mlr.press/v119/karimireddy20a.html (visited on 06/23/2024).

[KC03] J.O. Kephart and D.M. Chess, « The Vision of Autonomic Computing », *in*: *Computer* 36.*1* (Jan. 2003), pp. 41–50, ISSN: 0018-9162, DOI: 10.1109/MC.2003.1160055, URL: http://ieeexplore.ieee.org/document/1160055/.

[KC07] B. Kitchenham and S Charters, *Guidelines for Performing Systematic Literature Reviews in Software Engineering*, EBSE-2007-01, 2007.

[Kes07] S. Keshav, « How to Read a Paper », *in*: *acm special interest group on data communication* 37.*3* (2007), pp. 83–84.

[KGS21] Muah Kim, Onur Gunlu, and Rafael F Schaefer, « Federated Learning with Local Differential Privacy: Trade-offs between Privacy, Utility, and Communication », *in*: (2021).

[Kho+21] Tran Viet Khoa *et al.*, « Deep Transfer Learning: A Novel Collaborative Learning Model for Cyberattack Detection Systems in IoT Networks », Dec. 2, 2021, arXiv: 2112.00988 [cs], URL: http://arxiv.org/abs/2112.00988 (visited on 01/31/2022).

[Kim+20] Seongwoo Kim, He Cai, *et al.*, « Collaborative Anomaly Detection for Internet of Things Based on Federated Learning », *in*: *2020 IEEE/CIC International Conference on Communications in China (ICCC)*, 2020 IEEE/CIC International Conference on Communications in China (ICCC), Chongqing, China: IEEE, Aug. 9, 2020, pp. 623–628, ISBN: 978-1-72817-327-6, DOI: 10.1109/ICCC49849.2020.9238913, URL: https://ieeexplore.ieee.org/document/9238913/ (visited on 10/25/2021).

[Kol+16] Constantinos Kolias *et al.*, « Intrusion Detection in 802.11 Networks: Empirical Evaluation of Threats and a Public Dataset », *in*: *IEEE Communications Surveys & Tutorials* 18.*1* (2016), pp. 184–208, ISSN: 1553-877X, DOI: 10.1109/COMST.2015.2402161, URL: http://ieeexplore.ieee.org/document/7041170/.

[Kon+16a] Jakub Konený, H. Brendan McMahan, Daniel Ramage, *et al.*, « Federated Optimization: Distributed Machine Learning for On-Device Intelligence », *in*: (Oct. 8, 2016), pp. 1–38, URL: http://arxiv.org/abs/1610.02527.

[Kon+16b] Jakub Konený, H. Brendan McMahan, Felix X. Yu, *et al.*, « Federated Learning: Strategies for Improving Communication Efficiency », *in*: (Oct. 18, 2016), pp. 1–10, URL: http://arxiv.org/abs/1610.05492.

[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).

[Lan+22] Maxime Lanvin *et al.*, « Errors in the CICIDS2017 Dataset and the Significant Differences in Detection Performances It Makes », *in*: *CRiSIS 2022 - International Conference on Risks and Security of Internet and Systems*, Sousse, Tunisia, Dec. 2022, pp. 1–16, URL: https://hal.archives-ouvertes.fr/hal-03775466 (visited on 09/29/2022).

[Lav+22a] Leo Lavaur, Benjamin Coste, *et al.*, « Federated Learning as Enabler for Collaborative Security between Not Fully-Trusting Distributed Parties », *in*: *Proceedings of the 29th Computer & Electronics Security Application Rendezvous (C&ESAR): Ensuring Trust in a Decentralized World*, 2022, pp. 65–80, URL: http://ceur-ws.org/Vol-3329/paper-04.pdf.

[Lav+22b] Leo Lavaur, Marc-Oliver Pahl, *et al.*, « The Evolution of Federated Learning-based Intrusion Detection and Mitigation: A Survey », *in*: *IEEE Transactions on Network and Service Management*, Special Issue on Network Security Management (June 2022).

[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)*, July 2024.

[LDR19] Eliot Lear, Ralph Droms, and Dan Romascanu, *Manufacturer Usage Description Specification*, RFC 8520, RFC Editor, Mar. 2019, DOI: 10.17487/RFC8520, URL: https://rfc-editor.org/rfc/rfc8520.txt.

[Lec+98] Y. Lecun *et al.*, « Gradient-Based Learning Applied to Document Recognition », *in*: *Proceedings of the IEEE* 86.*11* (Nov. 1998), pp. 2278–2324, ISSN: 1558-2256, DOI: 10.1109/5.726791, URL: https://ieeexplore.ieee.org/document/726791 (visited on 11/09/2023).

[Li+20a] Beibei Li, Yuhao Wu, *et al.*, « DeepFed: Federated Deep Learning for Intrusion Detection in Industrial Cyber-Physical Systems », *in*: *IEEE Transactions on Industrial Informatics* 3203.*c* (2020), pp. 1–1, ISSN: 1551-3203, DOI: 10.1109/TII.2020.3023430, URL: https://ieeexplore.ieee.org/document/9195012/.

[Li+20b] Kun Li, Huachun Zhou, *et al.*, « Distributed Network Intrusion Detection System in Satellite-Terrestrial Integrated Networks Using Federated Learning », *in*: *IEEE Access* 8 (2020), pp. 214852–214865, ISSN: 2169-3536, DOI: 10.1109/ACCESS.2020.3041641, URL: https://ieeexplore.ieee.org/document/9274426/ (visited on 10/25/2021).

[Li+20c] Tian Li, Anit Kumar Sahu, *et al.*, « Federated Optimization in Heterogeneous Networks », Apr. 21, 2020, arXiv: 1812.06127 [cs, stat], URL: http://arxiv.org/abs/1812.06127 (visited on 09/20/2021).

[Liu+20] Lumin Liu, Jun Zhang, S.H. Song, *et al.*, « Client-Edge-Cloud Hierarchical Federated Learning », *in*: *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, ICC 2020 - 2020 IEEE International Conference on Communications (ICC), June 2020, pp. 1–6, DOI: 10.1109/ICC40277.2020.9148862, URL: https://ieeexplore.ieee.org/document/9148862 (visited on 06/23/2024).

[Liu+21] Hong Liu, Shuaipeng Zhang, Pengfei Zhang, *et al.*, « Blockchain and Federated Learning for Collaborative Intrusion Detection in Vehicular Edge Computing », *in*: *IEEE Transactions on Vehicular Technology* 70.*6* (June 2021), pp. 6073–6084, ISSN: 0018-9545, 1939-9359, DOI: 10.1109/TVT.2021.3076780, URL: https://ieeexplore.ieee.org/document/9420262/ (visited on 10/04/2021).

[LL19] Hongyu Liu and Bo Lang, « Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey », *in*: *Applied Sciences* 9.*20* (Oct. 17, 2019), p. 4396, ISSN: 2076-3417, DOI: 10.3390/app9204396, URL: https://www.mdpi.com/2076-3417/9/20/4396 (visited on 03/11/2022).

[LMK22] Wenjuan Li, Weizhi Meng, and Lam For Kwok, « Surveying Trust-Based Collaborative Intrusion Detection: State-of-the-Art, Challenges and Future Directions », *in*: *IEEE Communications Surveys & Tutorials* 24.*1* (2022), pp. 280–305, ISSN: 1553-877X, DOI: 10.1109/COMST.2021.3139052, URL: https://ieeexplore.ieee.org/document/9663537 (visited on 06/22/2024).

[Lo+21] Sin Kit Lo *et al.*, « A Systematic Literature Review on Federated Machine Learning: From A Software Engineering Perspective », *in*: *ACM Computing Surveys* 54.*5* (June 2021), pp. 1–39, ISSN: 0360-0300, 1557-7341, DOI: 10.1145/3450288, arXiv: 2007.11354, URL: http://arxiv.org/abs/2007.11354 (visited on 10/04/2021).

[LYY20] Lingjuan Lyu, Han Yu, and Qiang Yang, « Threats to Federated Learning: A Survey », *in*: *arXiv* (Mar. 4, 2020), URL: http://arxiv.org/abs/2003.02133.

[Mar+19] Samuel Marchal *et al.*, « AuDI: Toward Autonomous IoT Device-Type Identification Using Periodic Communication », *in*: *IEEE Journal on Selected Areas in Communications* 37.*6* (June 2019), pp. 1402–1412, ISSN: 0733-8716, 1558-0008, DOI: 10.1109/JSAC.2019.2904364, URL: https://ieeexplore.ieee.org/document/8664655/ (visited on 06/04/2021).

[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.

[Mei+18] Yair Meidan *et al.*, « N-BaIoT: Network-based Detection of IoT Botnet Attacks Using Deep Autoencoders », *in*: *IEEE Pervasive Computing* 17.*3* (July 2018), pp. 12–22, ISSN: 1536-1268, 1558-2590, DOI: 10.1109/MPRV.2018.03367731, arXiv: 1805.03409, URL: http://arxiv.org/abs/1805.03409 (visited on 10/23/2021).

[Men+15] Guozhu Meng, Yang Liu, *et al.*, « Collaborative Security: A Survey and Taxonomy », *in*: *ACM Computing Surveys* 48.*1* (July 22, 2015), 1:1–1:42, ISSN: 0360-0300, DOI: 10.1145/2785733, URL: https://doi.org/10.1145/2785733 (visited on 06/22/2024).

[Men+18] Weizhi Meng, Elmar Wolfgang Tischhauser, *et al.*, « When Intrusion Detection Meets Blockchain Technology: A Review », *in*: *IEEE Access* 6 (2018), pp. 10179–10188, ISSN: 2169-3536, DOI: 10.1109/ACCESS.2018.2799854, URL: http://ieeexplore.ieee.org/document/8274922/.

[Mer+23] Mohamed Amine Merzouk *et al.*, « Parameterizing Poisoning Attacks in Federated Learning-Based Intrusion Detection », *in*: *Proceedings of the 18th International Conference on Availability, Reliability and Security*, ARES '23, New York, NY, USA: Association for Computing Machinery, Aug. 29, 2023, pp. 1–8, ISBN: 9798400707728, DOI: 10.1145/3600160.3605090, URL: https://dl.acm.org/doi/10.1145/3600160.3605090 (visited on 01/29/2024).

[MG14] Thomas Morris and Wei Gao, « Industrial Control System Traffic Data Sets for Intrusion Detection Research », *in*: *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, ed. by Eduardo Bayro-Corrochano and Edwin Hancock, vol. 8827, Cham: Springer International Publishing, 2014, pp. 65–78, ISBN: 978-3-319-12567-1 978-3-319-12568-8, DOI: 10.1007/978-3-662-45355-1_5, URL: http://link.springer.com/10.1007/978-3-662-45355-1_5 (visited on 06/10/2021).

[ML15] Stuart Murdoch and Nick Leaver, « Anonymity vs. Trust in Cyber-Security Collaboration », *in*: *Proceedings of the 2nd ACM Workshop on Information Sharing and Collaborative Security*, New York, NY, USA: ACM, Oct. 12, 2015, pp. 27–29, ISBN: 978-1-4503-3822-6, DOI: 10.1145/2808128.2808134, URL: https://dl.acm.org/doi/10.1145/2808128.2808134.

[Mot+21a] Viraaji Mothukuri, Prachi Khare, *et al.*, « Federated Learning-based Anomaly Detection for IoT Security Attacks », *in*: *IEEE Internet of Things Journal* (2021), pp. 1–1, ISSN: 2327-4662, DOI: 10/gmhhmw.

[Mot+21b] Viraaji Mothukuri, Reza M. Parizi, *et al.*, « A Survey on Security and Privacy of Federated Learning », *in*: *Future Generation Computer Systems* 115 (Feb. 2021), pp. 619–640, ISSN: 0167739X, DOI: 10.1016/j.future.2020.10.007, URL: https://doi.org/10.1016/j.future.2020.10.007.

[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.

[Mou21] Nour Moustafa, « A New Distributed Architecture for Evaluating AI-based Security Systems at the Edge: Network TON_IoT Datasets », *in*: *Sustainable Cities and Society* 72 (Sept. 1, 2021), p. 102994, ISSN: 2210-6707, DOI: 10.1016/j.scs.2021.102994, URL: https://www.sciencedirect.com/science/article/pii/S2210670721002808 (visited on 06/21/2024).

[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).

[Nat24] National Institute of Standards and Technology, *The NIST Cybersecurity Framework (CSF) 2.0*, NIST CSWP 29, Gaithersburg, MD: National Institute of Standards and Technology, Feb. 26, 2024, NIST CSWP 29, DOI: 10.6028/NIST.CSWP.29, URL: https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.29.pdf (visited on 05/23/2024).

[NDG22] Evgenia Novikova, Elena Doynikova, and Sergey Golubev, « Federated Learning for Intrusion Detection in the Critical Infrastructures: Vertically Partitioned Data Use Case », *in*: *Algorithms* 15.*4* (4 Mar. 23, 2022), p. 104, ISSN: 1999-4893, DOI: 10.3390/a15040104, URL: https://www.mdpi.com/1999-4893/15/4/104 (visited on 07/05/2022).

[Ngu+19] Thien Duc Nguyen, Samuel Marchal, *et al.*, « DÏoT: A Federated Self-learning Anomaly Detection System for IoT », *in*: *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, vol. 2019-July, IEEE, July 2019, pp. 756–767, ISBN: 978-1-72812-519-0, DOI: 10.1109/ICDCS.2019.00080, URL: https://ieeexplore.ieee.org/document/8884802/.

[Ngu+20] Thien Duc Nguyen, Phillip Rieger, Markus Miettinen, *et al.*, « Poisoning Attacks on Federated Learning-based IoT Intrusion Detection System », *in*: *Proceedings 2020 Workshop on Decentralized IoT Systems and Security*, Workshop on Decentralized IoT Systems and Security, San Diego, CA: Internet Society, 2020, ISBN: 978-1-891562-64-8, DOI: 10.14722/diss.2020.23003, URL: https://www.ndss-symposium.org/wp-content/uploads/2020/04/diss2020-23003-paper.pdf (visited on 01/29/2024).

[Ngu+21] Thien Duc Nguyen, Phillip Rieger, Hossein Yalame, *et al.*, « FLGUARD: Secure and Private Federated Learning », Jan. 21, 2021, arXiv: 2101.02281 [cs], URL: http://arxiv.org/abs/2101.02281 (visited on 05/18/2021).

[Ngu+22] Thien Duc Nguyen, Phillip Rieger, Huili Chen, *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).

[OWN21] Yazan Otoum, Yue Wan, and Amiya Nayak, « Federated Transfer Learning-Based IDS for the Internet of Medical Things (IoMT) », *in*: *2021 IEEE Globecom Workshops (GC Wkshps)*, 2021 IEEE Globecom Workshops (GC Wkshps), Madrid, Spain: IEEE, Dec. 2021, pp. 1–6, ISBN: 978-1-66542-390-8, DOI: 10/gpbg4z, URL: https://ieeexplore.ieee.org/document/9682118/ (visited on 01/31/2022).

[PA18] Marc-Oliver Pahl and Francois Xavier Aubet, « All Eyes on You: Distributed Multi-Dimensional IoT Microservice Anomaly Detection », *in*: *14th International Conference on Network and Service Management, CNSM 2018, 1st Workshop on Segment Routing and Service Function Chaining* (2018), pp. 72–80.

[Pai99] Pascal Paillier, « Public-Key Cryptosystems Based on Composite Degree Residuosity Classes », *in*: *Advances in Cryptology — EUROCRYPT '99*, ed. by Jacques Stern, vol. 1592, Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 223–238, ISBN: 978-3-540-65889-4, DOI: 10.1007/3-540-48910-X_16, URL: http://link.springer.com/10.1007/3-540-48910-X_16 (visited on 07/05/2021).

[Pan+22] Dinelka Panagoda *et al.*, « Application of Federated Learning in Health Care Sector for Malware Detection and Mitigation Using Software Defined Networking Approach », *in*: *2022 2nd Asian Conference on Innovation in Technology (ASIANCON)*, 2022 2nd Asian Conference on Innovation in Technology (ASIANCON), Aug. 2022, pp. 1–6, DOI: 10.1109/ASIANCON55314.2022.9909488, URL: https://ieeexplore.ieee.org/abstract/document/9909488 (visited on 04/12/2024).

[PG22] Trung V. Phan and Tri Gia Nguyen, « FEAR: Federated Cyber-Attack Reaction in Distributed Software-Defined Networks with Deep Q-Network », *in*: *2022 Wireless Telecommunications Symposium (WTS)*, 2022 Wireless Telecommunications Symposium (WTS), Apr. 2022, pp. 1–7, DOI: 10.1109/WTS53620.2022.9768169.

[Pop+21a] Segun I. Popoola, Ruth Ande, *et al.*, « Federated Deep Learning for Zero-Day Botnet Attack Detection in IoT Edge Devices », *in*: *IEEE Internet of Things Journal* (2021), pp. 1–1, ISSN: 2327-4662, 2372-2541, DOI: 10.1109/JIOT.2021.3100755, URL: https://ieeexplore.ieee.org/document/9499122/ (visited on 10/01/2021).

[Pop+21b] Segun I. Popoola, Guan Gui, *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.

[Pop+22] Segun Popoola, Bamidele Adebisi, *et al.*, *Optimizing Deep Learning Model Hyperparameters for Botnet Attack Detection in IoT Networks*, preprint, Apr. 8, 2022, DOI: 10.36227/techrxiv.19501885.v1, URL: https://www.techrxiv.org/articles/preprint/Optimizing_Deep_Learning_Model_Hyperparameters_for_Botnet_Attack_Detection_in_IoT_Networks/19501885/1 (visited on 04/14/2022).

[Pop+23] Segun I. Popoola, Agbotiname L. Imoize, *et al.*, « Federated Deep Learning for Intrusion Detection in Consumer-Centric Internet of Things », *in*: *IEEE Transactions on Consumer Electronics* (2023), pp. 1–1, ISSN: 1558-4127, DOI: 10.1109/TCE.2023.3347170, URL: https://ieeexplore.ieee.org/abstract/document/10373897 (visited on 04/12/2024).

[PZ19] Ali Pala and Jun Zhuang, « Information Sharing in Cybersecurity: A Review », *in*: *Decision Analysis* 16.*3* (Sept. 2019), pp. 172–196, ISSN: 1545-8490, DOI: 10.1287/deca.2018.0387, URL: http://pubsonline.informs.org/doi/10.1287/deca.2018.0387.

[Qin+20a] Qiaofeng Qin, Konstantinos Poularakis, *et al.*, « Line-Speed and Scalable Intrusion Detection at the Network Edge via Federated Learning », *in*: *2020 IFIP Networking Conference (Networking)*, 2020 IFIP Networking Conference (Networking), June 2020, pp. 352–360, URL: https://ieeexplore.ieee.org/document/9142704 (visited on 04/12/2024).

[Qin+20b] Qiaofeng Qin, Konstantinos Poularakis, *et al.*, « Line-Speed and Scalable Intrusion Detection at the Network Edge via Federated Learning », *in*: (June 2020), p. 9.

[QK21] Yang Qin and Masaaki Kondo, « Federated Learning-Based Network Intrusion Detection with a Feature Selection Approach », *in*: *2021 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*, 2021 International Conference on Electrical, Communication, and Computer Engineering (ICECCE), Kuala Lumpur, Malaysia: IEEE, June 12, 2021, pp. 1–6, ISBN: 978-1-66543-897-1, DOI: 10.1109/ICECCE52056.2021.9514222, URL: https://ieeexplore.ieee.org/document/9514222/ (visited on 10/04/2021).

[Quy+22] Nguyen Huu Quyen *et al.*, « Federated Intrusion Detection on Non-IID Data for IIoT Networks Using Generative Adversarial Networks and Reinforcement Learning », *in*: *Information Security Practice and Experience*, ed. by Chunhua Su, Dimitris Gritzalis, and Vincenzo Piuri, Cham: Springer International Publishing, 2022, pp. 364–381, ISBN: 978-3-031-21280-2, DOI: 10.1007/978-3-031-21280-2_20.

[Rah+20] Sawsan Abdul Rahman *et al.*, « Internet of Things Intrusion Detection: Centralized, On-Device, or Federated Learning? », *in*: *IEEE Network* 34.*6* (Nov. 2020), pp. 310–317, ISSN: 0890-8044, 1558-156X, DOI: 10.1109/MNET.011.2000286, URL: https://ieeexplore.ieee.org/document/9183799/ (visited on 06/01/2021).

[Rin+17a] Markus Ring *et al.*, « Creation of Flow-Based Data Sets for Intrusion Detection », *in*: *Journal of Information Warfare* 16.*4* (2017), pp. 41–54, ISSN: 14453312, 14453347, JSTOR: 26504117, URL: https://www.jstor.org/stable/26504117.

[Rin+17b] Markus Ring *et al.*, « Flow-Based Benchmark Data Sets for Intrusion Detection », *in*: *Proceedings of the 16th European Conference on Cyber Warfare and Security (ECCWS)* (2017), pp. 361–369, ISSN: 20488610.

[Rod+23] Nuria Rodríguez-Barroso *et al.*, « Survey on Federated Learning Threats: Concepts, Taxonomy on Attacks and Defences, Experimental Study and Challenges », *in*: *Information Fusion* 90 (Feb. 1, 2023), pp. 148–173, ISSN: 1566-2535, DOI: 10.1016/j.inffus.2022.09.011, URL: https://www.sciencedirect.com/science/article/pii/S1566253522001439 (visited on 09/29/2023).

[RWP19] Shailendra Rathore, Byung Wook Kwon, and Jong Hyuk Park, « BlockSecIoTNet: Blockchain-based Decentralized Security Architecture for IoT Network », *in*: *Journal of Network and Computer Applications* 143 (December 2018 Oct. 2019), pp. 167–177, ISSN: 10848045, DOI: 10.1016/j.jnca.2019.06.019, URL: https://doi.org/10.1016/j.jnca.2019.06.019.

[SA21] Sudipan Saha and Tahir Ahmad, « Federated Transfer Learning: Concept and Applications », Mar. 6, 2021, arXiv: 2010.15561 [cs], URL: http://arxiv.org/abs/2010.15561 (visited on 10/04/2021).

[SB20] Jagdeep Singh and Sunny Behal, « Detection and Mitigation of DDoS Attacks in SDN: A Comprehensive Review, Research Challenges and Future Directions », *in*: *Computer Science Review* 37 (Aug. 2020), p. 100279, ISSN: 15740137, DOI: 10.1016/j.cosrev.2020.100279, URL: https://linkinghub.elsevier.com/retrieve/pii/S1574013720301647 (visited on 04/09/2022).

[SEO21] Yuwei Sun, Hiroshi Esaki, and Hideya Ochiai, « Adaptive Intrusion Detection in the Networking of Large-Scale LANs With Segmented Federated Learning », *in*: *IEEE Open Journal of the Communications Society* 2 (2021), pp. 102–112, ISSN: 2644-125X, DOI: 10.1109/OJCOMS.2020.3044323, URL: https://ieeexplore.ieee.org/document/9296578/ (visited on 10/04/2021).

[Sha+24] Yao Shan *et al.*, « CFL-IDS: An Effective Clustered Federated Learning Framework for Industrial Internet of Things Intrusion Detection », *in*: *IEEE Internet of Things Journal* 11.*6* (Mar. 2024), pp. 10007–10019, ISSN: 2327-4662, DOI: 10.1109/JIOT.2023.3324302, URL: https://ieeexplore.ieee.org/abstract/document/10285326 (visited on 04/12/2024).

[She+20] Sheng Shen *et al.*, « From Distributed Machine Learning To Federated Learning: In The View Of Data Privacy And Security », *in*: *Concurrency and Computation: Practice and Experience* (Sept. 23, 2020), cpe.6002, ISSN: 1532-0626, 1532-0634, DOI: 10.1002/cpe.6002, arXiv: 2010.09258, URL: http://arxiv.org/abs/2010.09258 (visited on 10/04/2021).

[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).

[Sig99] SigKDD, *KDD Cup 1999 Dataset*, 1999, URL: https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html (visited on 04/26/2021).

[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).

[Sna+92] Steven R. Snapp *et al.*, « The DIDS (Distributed Intrusion Detection System) Prototype », *in*: *USENIX Summer 1992 Technical Conference (USENIX Summer 1992 Technical Conference)*, San Antonio, TX: USENIX Association, June 1992, URL: https://www.usenix.org/conference/usenix-summer-1992-technical-conference/dids-distributed-intrusion-detection-system.

[SOE20] Yuwei Sun, Hideya Ochiai, and Hiroshi Esaki, « Intrusion Detection with Segmented Federated Learning for Large-Scale Multiple LANs », *in*: *2020 International Joint Conference on Neural Networks (IJCNN)*, 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, United Kingdom: IEEE, July 2020, pp. 1–8, ISBN: 978-1-72816-926-2, DOI: 10.1109/IJCNN48605.2020.9207094, URL: https://ieeexplore.ieee.org/document/9207094/ (visited on 10/01/2021).

[SSF16] Florian Skopik, Giuseppe Settanni, and Roman Fiedler, « A Problem Shared Is a Problem Halved: A Survey on the Dimensions of Collective Cyber Defense through Security Information Sharing », *in*: *Computers & Security* 60 (2016), pp. 154–176, ISSN: 01674048, DOI: 10.1016/j.cose.2016.04.003.

[ST19] William Schneble and Geethapriya Thamilarasu, « Attack Detection Using Federated Learning in Medical Cyber-Physical Systems », *in*: *2019 28th International Conference on Computer Communication and Networks (ICCCN)*, International Conference on Computer Communications and Networks, Aug. 2019.

[Sun+20] Wen Sun, Shiyu Lei, *et al.*, « Adaptive Federated Learning and Digital Twin for Industrial Internet of Things », Oct. 31, 2020, arXiv: 2010.13058 [cs], URL: http://arxiv.org/abs/2010.13058 (visited on 10/04/2021).

[Tan+23] Zhenheng Tang *et al.*, « GossipFL: A Decentralized Federated Learning Framework With Sparsified and Adaptive Communication », *in*: *IEEE Transactions on Parallel and Distributed Systems* 34.*3* (Mar. 2023), pp. 909–922, ISSN: 1558-2183, DOI: 10.1109/TPDS.2022.3230938, URL: https://ieeexplore.ieee.org/document/9996127 (visited on 06/23/2024).

[Tav+09] Mahbod Tavallaee *et al.*, « A Detailed Analysis of the KDD CUP 99 Data Set », *in*: *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, IEEE, July 2009, pp. 1–6, ISBN: 978-1-4244-3763-4, DOI: 10.1109/CISDA.2009.5356528, URL: http://ieeexplore.ieee.org/document/5356528/.

[Thi+22] Huynh Thai Thi *et al.*, « Federated Learning-Based Cyber Threat Hunting for APT Attack Detection in SDN-Enabled Networks », *in*: *2022 21st International Symposium on Communications and Information Technologies (ISCIT)*, 2022 21st International Symposium on Communications and Information Technologies (ISCIT), Sept. 2022, pp. 1–6, DOI: 10.1109/ISCIT55906.2022.9931222, URL: https://ieeexplore.ieee.org/abstract/document/9931222 (visited on 04/12/2024).

[TKM20] Mineto Tsukada, Masaaki Kondo, and Hiroki Matsutani, « A Neural Network-Based On-device Learning Anomaly Detector for Edge Devices », *in*: *IEEE Transactions on Computers* (2020), pp. 1–1, ISSN: 0018-9340, 1557-9956, 2326-3814, DOI: 10.1109/TC.2020.2973631, URL: https://ieeexplore.ieee.org/document/9000710/ (visited on 10/23/2021).

[Tol+20] Vale Tolpegin *et al.*, « Data Poisoning Attacks Against Federated Learning Systems », *in*: *Computer Security – ESORICS 2020*, ed. by Liqun Chen *et al.*, Lecture Notes in Computer Science, Cham: Springer International Publishing, 2020, pp. 480–501, ISBN: 978-3-030-58951-6, DOI: 10.1007/978-3-030-58951-6_24.

[TR18] Wiem Tounsi and Helmi Rais, « A Survey on Technical Threat Intelligence in the Age of Sophisticated Cyber Attacks », *in*: *Computers & Security* 72 (Jan. 2018), pp. 212–233, ISSN: 01674048, DOI: 10.1016/j.cose.2017.09.001, URL: https://doi.org/10.1016/j.cose.2017.09.001.

[VKF15] Emmanouil Vasilomanolakis, Shankar Karuppayah, and Mathias Fischer, « Taxonomy and Survey of Collaborative Intrusion Detection », *in*: *ACM Computing Surveys* 47.4 (May 2015), p. 33, DOI: 10.1145/2716260.

[Vy+21] Nguyen Chi Vy *et al.*, « Federated Learning-Based Intrusion Detection in the Context of IIoT Networks: Poisoning Attack and Defense », *in*: *Network and System Security*, ed. by Min Yang, Chao Chen, and Yang Liu, vol. 13041, Cham: Springer International Publishing, 2021, pp. 131–147, ISBN: 978-3-030-92707-3 978-3-030-92708-0, DOI: 10.1007/978-3-030-92708-0_8, URL: https://link.springer.com/10.1007/978-3-030-92708-0_8 (visited on 03/05/2024).

[Wag+19] Thomas D. Wagner *et al.*, « Cyber Threat Intelligence Sharing: Survey and Research Directions », *in*: *Computers & Security* 87 (2019), p. 101589, ISSN: 01674048, DOI: 10.1016/j.cose.2019.101589.

[Wag19] Thomas D Wagner, « Cyber Threat Intelligence for "Things" », *in*: *2019 International Conference on Cyber Situational Awareness, Data Analytics And Assessment (Cyber SA)*, IEEE, June 2019, pp. 1–2, ISBN: 978-1-72810-232-0, DOI: 10.1109/CyberSA.2019.8899384, URL: https://ieeexplore.ieee.org/document/8899384/.

[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.

[Yad19] Omry Yadan, *Hydra - A Framework for Elegantly Configuring Complex Applications*, Github, 2019, URL: https://github.com/facebookresearch/hydra.

[Yan+19] Qiang Yang, Yang Liu, *et al.*, « Federated Machine Learning: Concept and Applications », *in*: *ACM Transactions on Intelligent Systems and Technology* 10.*2* (Feb. 28, 2019), pp. 1–19, ISSN: 2157-6904, DOI: 10.1145/3298981, URL: https://dl.acm.org/doi/10.1145/3298981.

[Yan+23] Run Yang, Hui He, *et al.*, « Dependable Federated Learning for IoT Intrusion Detection against Poisoning Attacks », *in*: *Computers & Security* 132 (Sept. 1, 2023), p. 103381, ISSN: 0167-4048, DOI: 10.1016/j.cose.2023.103381, URL: https://www.sciencedirect.com/science/article/pii/S0167404823002912 (visited on 03/04/2024).

[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).

[Zha+19] Ying Zhao *et al.*, « Multi-Task Network Anomaly Detection Using Federated Learning », *in*: *Proceedings of the Tenth International Symposium on Information and Communication Technology - SoICT 2019*, The Tenth International Symposium, Hanoi, Ha Long Bay, Viet Nam: ACM Press, 2019, pp. 273–279, ISBN: 978-1-4503-7245-9, DOI: 10.1145/3368926.3369705, URL: http://dl.acm.org/citation.cfm?doid=3368926.3369705 (visited on 06/07/2021).

[Zha+20a] Weishan Zhang, Qinghua Lu, *et al.*, « Blockchain-Based Federated Learning for Device Failure Detection in Industrial IoT », *in*: *IEEE Internet of Things Journal* (Sept. 6, 2020), pp. 1–1, ISSN: 2327-4662, DOI: 10.1109/JIOT.2020.3032544, URL: https://ieeexplore.ieee.org/document/9233457/.

[Zha+20b] Weishan Zhang, Tao Zhou, *et al.*, « Dynamic Fusion Based Federated Learning for COVID-19 Detection », *in*: *arXiv* 14.*8* (Sept. 22, 2020), pp. 1–9, ISSN: 23318422, URL: http://arxiv.org/abs/2009.10401.

[Zha+22] Yuemeng Zhang, Yong Zhang, *et al.*, « Evaluation of Data Poisoning Attacks on Federated Learning-Based Network Intrusion Detection System », *in*: *2022 IEEE 24th Int Conf on High Performance Computing & Communications; 8th Int Conf on Data Science & Systems; 20th Int Conf on Smart City; 8th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys)*, 2022 IEEE 24th Int Conf on High Performance Computing & Communications; 8th Int Conf on Data Science & Systems; 20th Int Conf on Smart City; 8th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys), Dec. 2022, pp. 2235–2242, DOI: 10.1109/HPCC-DSS-SmartCity-DependSys57074.2022.00330, URL: https://ieeexplore.ieee.org/document/10074658 (visited on 10/31/2023).

[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

A Additional figures

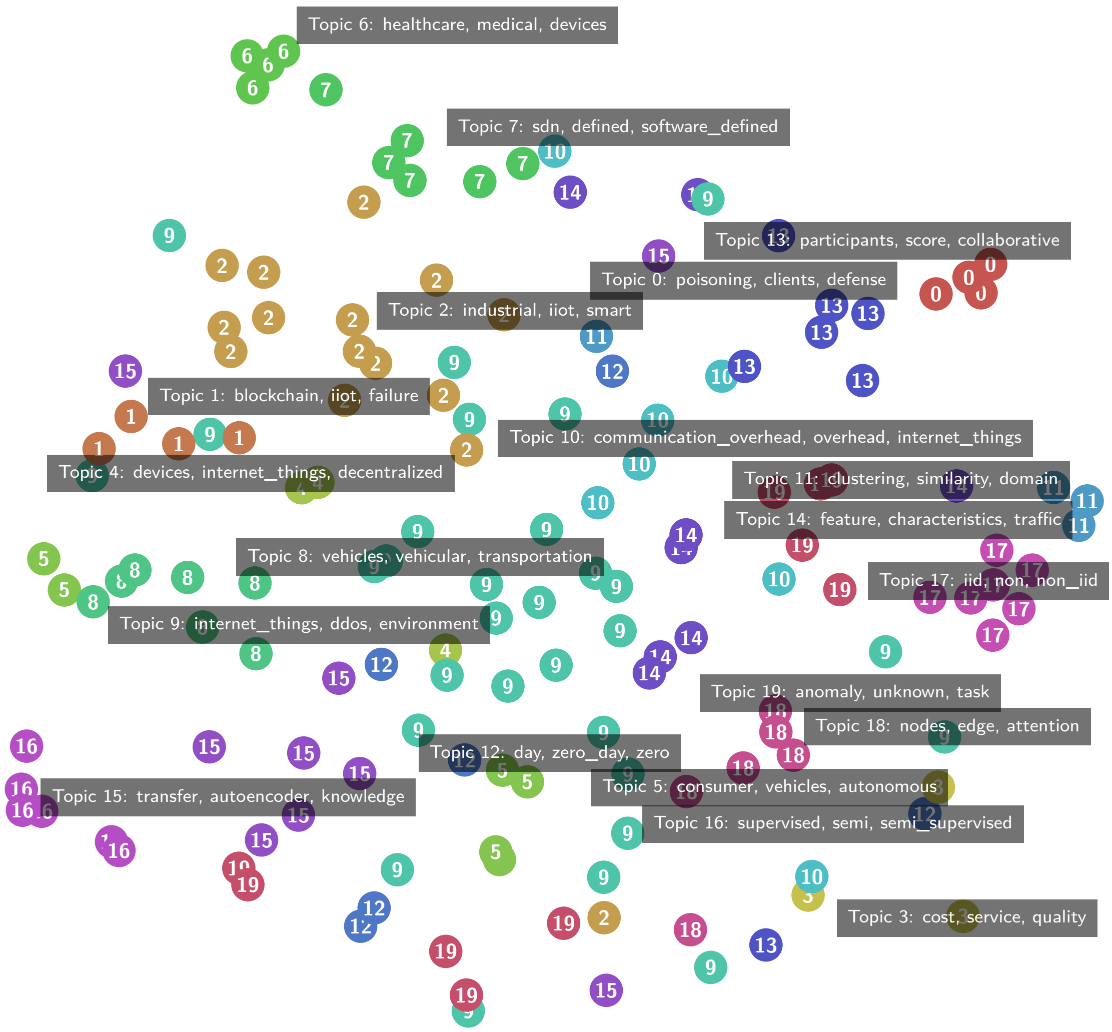


Figure 9.1 – Topic embedding of the Federated Intrusion Detection System (FIDS) literature using a Non-negative Matrix Factorization (NMF) model with 20 topics. Each point represents a paper, and each are labelled with the topic they are the most associated with.

B Résumé en français de la thèse

COLLEGES SCIENCES
BRETAGNE POUR L'INGENIEUR
LOIRE ET LE NUMERIQUE

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

Titre : Améliorer la détection d'intrusions dans des systèmes distribués grâce à l'apprentissage fédéré

Mot clés : apprentissage automatique, apprentissage fédéré, détection d'intrusions, collaboration, 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 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 distribués 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 distribuées 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 l'impact de la qualité des données dans des contextes hétérogènes, certains types d'attaques par empoisonnement, et proposons des outils et des méthodologies pour améliorer l'évaluation de ce type d'algorithmes distribués.

Title: Improving Intrusion Detection in Distributed Systems with Federated Learning

Keywords: machine learning, federated learning, intrusion detection, collaboration, 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, 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 the impact of data quality in heterogeneous contexts, some types of poisoning attacks, and propose tools and methodologies to improve the evaluation of these types of distributed algorithms.