Federated Learning as enabler for Collaborative Security between not Fully-Trusting Distributed Parties

Léo Lavaur, Benjamin Costé, Marc-Oliver Pahl, Yann Busnel, Fabien Autrel

Federated Learning as enabler for Collaborative Security between not Fully-Trusting Distributed Parties

Léo Lavaur[1,2], Benjamin Costé[3], Marc-Oliver Pahl[1,2], Yann Busnel[1] and Fabien Autrel[1]

[1]IMT-Atlantique, IRISA
[2]Chaire Cybersécurité des Infrastructures Critiques (Cyber CNI)
[3]Airbus CyberSecurity

Abstract

Literature shows that trust typically relies on knowledge about the communication partner. Federated learning is an approach for collaboratively improving machine learning models. It allows collaborators to share Machine Learning models without revealing secrets, as only the abstract models and not the data used for their creation is shared. Federated learning thereby provides a mechanism to create trust without revealing secrets, such as specificities of local industrial systems.

A fundamental challenge, however, is determining how much trust is justified for each contributor to collaboratively optimize the joint models. By assigning equal trust to each contribution, divergence of a model from its optimum can easily happen—caused by errors, bad observations, or cyberattacks. Trust also depends on how much an aggregated model contributes to the objectives of a party. For example, a model trained for an OT system is typically useless for monitoring IT systems.

This paper shows first directions how heterogeneous distributed data sources could be integrated using federated learning methods. With an extended abstract, it shows current research directions and open issues from a cyber-analyst's perspective.

Keywords

Federated learning, cybersecurity, intrusion detection, distributed trust

1. Introduction

A common cybersecurity goal is thwarting attackers through detection of their actions, understanding of their methodologies, and increasing the resilience before their next attempts. However, the considerable variety, thus complexity, of the tools and techniques used by attackers drown their traces in the high volume of legitimate network traffic and endpoints logs. One way used by defenders (often called "Blue teams") to act against threat actors is sharing knowledge about their abuses. However, knowledge gathered during security monitoring or

C&ESAR'22: Computer & Electronics Security Application Rendezvous, Nov. 15-16, 2022, Rennes, France
leo.lavaur@imt-atlantique.fr (L. Lavaur); benjamin.b.coste@airbus.com (B. Costé); marc-oliver.pahl@imt-atlantique.fr (M. Pahl); yann.busnel@imt-atlantique.fr (Y. Busnel); fabien.autrel@imt-atlantique.fr (F. Autrel)
0000-0002-0379-7946 (L. Lavaur); 0000-0002-5631-9120 (B. Costé); 0000-0001-5241-3809 (M. Pahl); 0000-0001-6908-719X (Y. Busnel); 0000-0002-8403-515X (F. Autrel)

incident response is often coupled with private data which cannot be shared for confidentiality reasons (GDPR, NDA, etc.).

Machine Learning (ML) approaches can help here, as they result in abstract models that are typically not reversible to their input data [1]. A fundamental problem for observation-based security is the amount of data needed for having a trustable and reliable impression. Typically, more data allows for better behavior characterization, thus improving either anomaly detection, or event classification. Collecting data requires either a long observation time or many data sources, as comprehensiveness is difficult to reach.

Federated Learning (FL) has been introduced to enable the sharing of local models and to federate them towards better joint models. Each participant computes on its own an ML-model using its own data. The resulting model is aggregated with the ones of other participants, typically by a trusted party, then the new model is shared between each of them.

Consequently, all parties benefit from each other while no one have access to their private data and algorithms. Sharing models however still faces issues as models can be modified with backdoors [2], poisoned with adversarial approaches [3] or simply suffers from bad quality training dataset.

This paper discusses FL sharing approaches through aggregation issues. Section 2 introduces the concept of Machine Learning (ML) based intrusion detection, and Federated Learning (FL) as a collaboration enabler. In Section 3, we summarize the literature around federated intrusion detection, with an extended abstract of a survey. Section 4 outlines an experimental use-case for Federated Intrusion Detection System (FIDS) application. Section 5 discusses open issues and the envisioned solutions. Section 6 concludes our proposition.

2. Background: ML for collaboratively defend cyberattacks

One prominent application of FL in the cybersecurity field is for security monitoring and collaborative intrusion detection. This section defines the basis for the remainder of the paper.

Security monitoring systems frequently use signature-based Intrusion Detection Systems (IDSs) to detect known attacks to safeguard companies [4]. However, this strategy suffers from serious limitations against zero-day and one-day attacks, as well as Advanced Persistent Threats (APTs), such as Stuxnet [5]. Furthermore, the IoT's heterogeneity and irregular traffic cause IDSs to be less effective or inadequate [6]. As a result, researchers began to look into anomaly detection as a way to improve detection systems.

Here, multiple approaches coexist, depending on objectives and available data. On the one hand, anomaly-based detection systems compare monitored events to a baseline profile trained on nominal traffic to assess whether they are malicious or not [7]. On the other hand, pattern-based classification, aims at extracting patterns from known attacks that have been previously labelled as such, and then characterize input data according to the extracted patterns. Therefore, anomaly-based approaches are particularly relevant to detect unknown behaviors, whereas supervised ones are more equipped for threat characterization.

The source of data will also influence the available features for detection. For example, endpoint-oriented sensors such as Sysmon monitor processes, use of windows' registry key or system calls while network-oriented sensors provides a lot of information about protocols,

packet length, or IP addresses. Preprocessing can be used to extract additional features from the raw data, to provide more information. The literature distinguishes three key non-exclusive approaches: feature extraction, feature embedding, and feature selection [8].

In the context of knowledge sharing, the choice of the input feature is critical to transfer knowledge between participants. For instance, IP addresses are specific to the local layout of the network, and might have another meaning in another environment, if they even exist. On the other hand, given one class of devices, inter-arrival time (IAT) should not vary much between networks, thus making potentially making it a good choice for transferring knowledge.

With a designated set of features and an input dataset, one can establish a model of the data. A model is an abstraction that can be used afterward for other tasks, such as characterization. Algorithmically speaking, a model is a set of mathematical parameters that are inferred from input data by an algorithm. It can be the statistical parameters of a distribution function, the condition nodes in a decision tree, or the weights and biases of a neural network. The ability of a model to be shared partly depends on how easily these parameters can be aggregated.

Over the years, ML have been applied to intrusion detection with substantial results. Three approaches coexist:

(1) Anomaly detection becomes a binary classification problem with supervised learning. For effective training, a balanced labeled dataset is required. However, because local training data is infrequently labeled and models can be affected by unbalanced class distribution, supervised learning is more difficult to apply in real-world scenarios [9].

(2) For unlabeled data, unsupervised learning is more appropriate. In the case of IDS, we assume that (i) benign traffic is substantially more common in the testing set than anomalies [10]; and (ii) abnormal packets are statistically different from normal packets.

(3) Semi-supervised learning is a hybrid method that labels only a portion of the training data. It can be used to bootstrap a detection model with a publicly labeled dataset before training it on locally collected data afterward.

However, ML algorithms also require a lot of training data to avoid learning biased model, from the lack of exhaustively or from an overrepresentation of a class in the training set.

To cope with the limitations of ML, especially when training data is locally collected, collaborative IDSs have emerged in the literature. However, they are almost always built in a centralized manner, which induces its own set of issues: (i) centralizing a system typically introduce a single point-of-failure (SPoF) [11]; (ii) centralized IDSs imply sending local data for training or detection, increasing the risk of information disclosure [12]; (iii) communicating data over networks also increase bandwidth consumption and latency, which are critical for intrusion detection [13].

Introduced in 2016 by Google [14], FL promises to cope with these issues. In FL, model learning is distributed among the participants of the federation. Therefore, local data stay in the participant's system, and collaboration is achieved by sharing and aggregating the generated models. Aggregation can be done by a server [6], [15]–[17], but it might lead to concerns with trust and privacy. Due to challenges in terms of traceability, integrity, privacy, and trust, recent research has favored the usage of trusted distributed ledgers [18], [19], multi-party computation (MPC) [20], and privacy-preserving mechanisms like Differential Privacy (DP) [21].

3. The example of collaborative intrusion detection

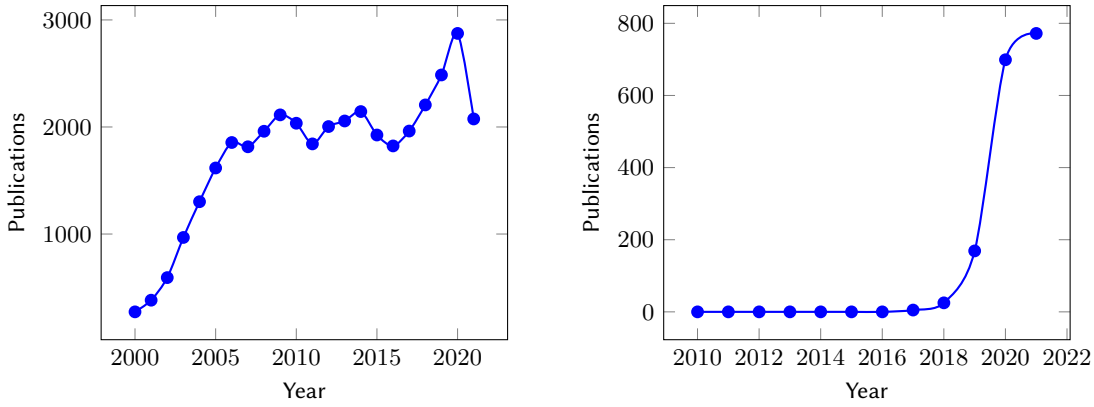
Since its introduction, FL has been applied to multiple domains, such as intrusion detection, whose presence is increasing in the literature. In this context, FL allows local detection of attacks—thus offering low latency and bandwidth—while jointly enriching participant's models and preserving data privacy.

This section is an extended abstract of the study published in *IEEE Transactions on Network and Service Management* (TNSM) [8]. Contributions of the study are as follows: (1) it examines the application of FL to the detection and mitigation of attacks; (2) it proposes a reference architecture that generalizes the selected work and can serve as a starting point for the design of future FIDSs; (3) it establishes a taxonomy of FIDSs, which provides a framework for comparing the selected works in this study; and (4) it highlights open questions regarding FIDSs and identifies associated research directions. The findings presented in this study can easily be extrapolated to other cybersecurity applications of FL, such as automated forensic analysis, or malware detection and classification.

Section 3.1 presents the results of the study and the established taxonomy, whereas Section 3.2 reviews identified research directions.

3.1. Literature review

The study performs two analyses of the literature. The quantitative analysis is based on objective metrics that can be extracted from the literature, such as publication date and venue. Figures 1a and 1b show the evolution of the literature on FL and IDS over time, both being building blocks of FIDSs.



(a) Keywords: `intrusion detection system` (b) Keywords: `federated learning`

Figure 1: Evolution of related domains until 08/10/2021, data from Microsoft Academic [22]
—Figures from Lavaur *et al.* [8] © 2022 IEEE

The qualitative analysis is based on the comparison of existing approaches, using the proposed taxonomy. They can be grouped into four categories: data, local operation, federation, and aggregation. A fifth meta-category is dedicated to the implementation and evaluation of the approach.

The taxonomy (Figure 2) provides twelve characteristics on which all approaches can be compared:

1. *Data source and type*: what data is collected and how; heavily depends on the use case.
2. *Preprocessing*: strategies for data curation, such as normalization and feature selection.
3. *ML location*: where the ML model is trained and executed.
4. *Local algorithms*: how the ML model is trained, and its impact on performance.
5. *Defense capabilities*: the ability of the approach to mitigate attacks.
6. *Federation strategy*: how the federation is organized; *e.g.* client selection, architecture.
7. *Communication*: how data (*i.e.* models) is exchanged between participants, including protection mechanisms.
8. *FL type*: type of FL strategy, depending on the objectives and available data.
9. *Aggregation strategy*: how models are aggregated, especially with heterogeneous clients.
10. *Model target*: *i.e.* the balance between specialization and generalization.
11. *Analyzed dataset*: the dataset used to evaluate the approach; often Information Technology (IT)-focused, and not always available.
12. *Costs and metrics*: how the approach is evaluated, depending on the use case and objectives.

The structure provided by the taxonomy also allows comparing the selected works. Table 1 summaries the results of the comparison. It shows that most approaches focus on IT network traffic, using a gateway to collect data and host both learning and detection processes. Most also use Deep Learnings (DLs) to perform supervised learning and classify traffic. A majority use unmodified `FedAvg` for the aggregation, which is the initial FL algorithm that was proposed by Google [14].

3.2. Open issues and research directions

As the FL topic gets more mature, research tends to focus on secondary aspects, such as security and privacy, or on the application of FL to other use cases. This subsection summaries the open issues and research directions; more details are available in [8].

(i) **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 avenues have been identified in the literature, such as the use of Generative Adversarial Networkss (GANs) or the improvement of feature selection as input to the model. Moreover, the link between the performance of the system and its hyper- and meta-parameters is not yet established.

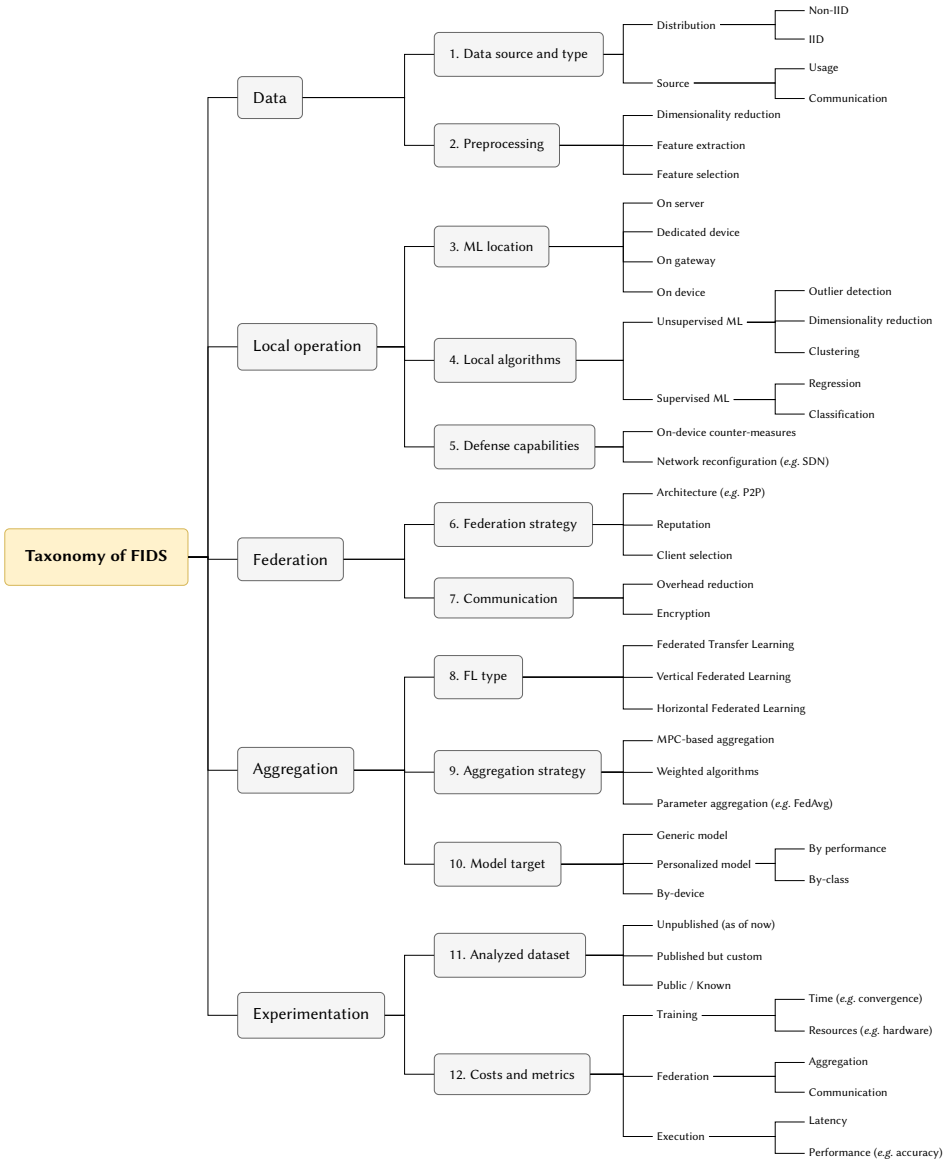


Figure 2: Proposed taxonomy for FIDS—Figure from Lavaur *et al.* [8] © IEEE

(ii) **Adaptability and 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 numerous clients. Therefore, further research is needed on FIDSs client selection: dynamic fusion based on score or time, reputation, number of detected attacks, etc.

(iii) **Knowledge transfer** — Current solutions focus on federating training and detection for devices and resources that belong in the same domain. Therefore, open issues include the

Table 1
Comparative table of selected works—Data from Lavaur *et al.* [8] © 2022 IEEE

	Ref	Internet of Things	Information Technologies	Cyber Physical systems	Autonomous Vehicles	Satellite-terrestrial networks	Federated Transfer Learning	Horizontal FL	Vertical FL	Federated Mimic Learning	Federated MTL	Online learning	Semi-supervised	Supervised	Unsupervised	Personalized models	Network-based	Usage-based	Training location	Data type	Local Algorithm	Federation Algorithm	Dataset	Strengths
2018	Pahl *et al.* [23]	●	○	○	○	○	●	○	○	○	○	●	○	○	●	●	●	○	Device	Abstracted network traffic (middleware)	BIRCH K-means	Parameter addition	Generated	relatively lightweight, online, no labels
2019	Rathore *et al.* [11]	○	●	○	○	○	●	○	○	○	○	○	●	○	○	●	●	○	Edge-controller (SDN)	Network traffic (SDN)	ANN	Vector concatenation	NSL-KDD	offers mitigation, decentralized
2019	Nguyen, Marchal, *et al.* [6]	●	○	○	○	○	●	○	○	○	○	●	○	○	●	●	●	○	Gateway	IoT network traffic (TCPdump)	MLP	Weight and biases average	MIMIC	online, offers per-class models, no labels
2019	Zhao *et al.* [24]	○	●	○	○	○	○	○	○	●	○	○	●	○	●	●	●	○	Gateway	Encrypted network traffic (CICFlowMeter)	GRU	FedAvg	Generated	versatile (multi-task)
2019	Schneble *et al.* [25]	○	○	●	○	○	●	○	○	○	○	●	○	○	●	○	○	●	Gateway	Healthcare sensor values	FC (shared layers) → FC	Weight and biases average	CICIDS2017 ISCXVPN2016 ISCXTor2016	high adaptability, no labels
2019	Cetin *et al.* [26]	○	●	○	○	○	●	○	○	○	○	○	●	○	○	○	●	○	Gateway	Network traffic (WIFI)	SAE	FedAvg	AWID	–
2020	Zhang *et al.* [18]	●	○	○	○	○	●	○	○	○	○	○	●	○	○	○	○	●	Gateway	Air conditioner sensor values	CNN-GRU → MLP	Homomorphic parameter addition	CPS dataset	offers traceability (blockchain)
2020	Li, Wu, *et al.* [15]	○	○	●	○	○	●	○	○	○	○	○	●	○	○	○	●	○	Gateway	MODBUS traffic	DAGMM	Parameter addition	KDD 99	confidentiality (encryption)
2020	Rahman *et al.* [27]	○	●	○	○	○	●	○	○	○	○	○	●	○	○	○	●	○	Device	IoT network traffic (TCPdump)	ANN	CDW_FedAvg	Generated	–
2020	Chen, Zhang, *et al.* [28]	○	●	○	○	○	●	○	○	○	○	○	○	○	●	○	●	○	Gateway	IoT network traffic (TCPdump)	CNN	Parameter aggregation	CICIDS2017 NSL-KDD Generated	no labels
2020	Sun, Ochiai, *et al.* [29]	○	●	○	○	○	●	○	○	○	○	○	●	○	○	●	●	○	Gateway	Network traffic (PCAP)	ANN	FedAvg	NSL-KDD	segmented (performance-based models)
2020	Fan *et al.* [30]	●	○	○	○	○	○	○	●	○	○	○	●	○	○	●	●	○	Gateway (MEC)	IoT network traffic (TCPdump, CICFlowMeter)	CNN	Parameter aggregation	LAN-Security Monitoring Project	knowledge transfer between public and private datasets
2020	Al-Marri *et al.* [31]	○	●	○	○	○	○	○	○	○	●	○	●	○	○	○	●	○	Gateway	Network traffic (TCPdump)	ANN	FedAvg	NSL-KDD	enhanced privacy (mimic learning)
2020	Kim, Cai, *et al.* [32]	○	●	○	○	○	●	○	○	○	○	○	●	○	○	○	●	○	Gateway	Network traffic (TCPdump)	MLP	FedAvg	NSL-KDD	–
2020	Qin, Poularakis, *et al.* [33]	○	●	○	○	○	●	○	○	○	○	○	●	○	○	●	●	○	Gateway (SDN)	Network traffic (SDN)	BNN	SignSGD	CICIDS2017 ISCX Botnet 2014	very lightweight, line-speed classification, P4 language compatible
2020	Chen, Lv, *et al.* [34]	○	●	○	○	○	●	○	○	○	○	○	●	○	○	○	●	○	Gateway	Network traffic (CICFlowMeter)	GRU-SVM	FedAGRU	CICIDS2017 KDD 99 WSN-DS	robust to poisoning, scalable
2020	Hei *et al.* [35]	○	●	○	○	○	●	○	○	○	○	●	○	●	○	○	●	○	Device	Network traffic (TCPdump)	MLP	FedAvg	DARPA 1999	online, offers traceability (blockchain)
2020	Li, Zhou, *et al.* [36]	○	●	○	○	●	●	○	○	○	○	○	●	○	○	○	●	○	Gateway	Network traffic (PCAP, CICFlowMeter, Argus)	CNN	Homomorphic parameter addition	Generated	relatively lightweight, confidentiality (encryption)
2021	Popoola *et al.* [37]	●	○	○	○	○	●	○	○	○	○	○	●	○	○	○	●	○	Gateway	IoT network traffic (TCPdump, Argus)	MLP	Parameter aggregation	KDD 99	zero-days detection
2021	Qin and Kondo [38]	○	●	○	○	○	●	○	○	○	○	○	●	○	○	●	●	○	Device	Network traffic (TCPdump)	ANN	FedAvg	Bot-IoT N-BaIoT	relatively lightweight
2021	Liu *et al.* [39]	○	○	○	●	○	●	○	○	○	○	○	●	○	○	○	●	○	Device	Network traffic (TCPdump)	ELM + AE	FedAvg	NSL-KDD	decentralized
2021	Sun, Esaki, *et al.* [40]	○	●	○	○	○	●	○	○	○	○	○	●	○	○	●	●	○	Gateway	Network traffic (PCAP)	CNN	Parameter aggregation	LAN-Security Monitoring Project	segmented (performance-based models)
			Use case				FL type					Training				Approach								

ability to federate clients across different domains. In addition, current methods often consider that all local models share the same architecture and hyper-parameters. This limitation makes current FIDSs less versatile and transferable.

(iv) **Security and trust** — Using ML or FL to detect intrusions can introduce new threats to the system, such as poisoning. Several works have examined the vulnerabilities of FL systems and proposed countermeasures. With FL, poisoning becomes easier, as any participant can theoretically impact everyone else's model. Structure of models depends on the architecture of the underlying algorithm, models trained cannot be aggregated easily [41]. Furthermore, as ML, and especially DL, lacks explainability, the content of a model is difficult to infer. Its aggregation with others is therefore made more risky. Future works are required in this direction to properly assess the content of a model before aggregation with others. Current solutions require an increase in the trust attributable to clients for model aggregation, inspired by the state of the art in collaboration systems and information sharing platforms.

(v) **Self-defense and self-healing** — Current research on FIDSs focuses on intrusion detection and attack classification. Defense is barely represented in the literature. However,

technologies such as Software-defined networking (SDN) offer rapid resiliency capabilities; and recent work studies the effectiveness of such defense mechanisms. New emerging applications such as self-defense and self-healing systems could benefit from FIDSs and other FL-based technologies.

(vi) **Model convergence** — Models can differ from one client to another, especially in heterogeneous contexts like intrusion detection. Consequently, model convergence is made more difficult. Current research focus on optimizing parameters by considering aggregation as an optimization problem. For instance, Charles *et al.* [42] use meta-learning to infer the right parameters, thus optimizing the aggregation afterward. Weighting mechanisms are also present in the literature to improve the convergence [43].

(vii) **Dataset representativity** — Existing public datasets are not representative of FIDSs potential deployment environments. Indeed, they are often datasets produced for traditional machine learning algorithms but split for federated purposes, like NSL-KDD [44], UNSW-NB15 [45], or CIC-IDS2017 [46]. However, this approach introduces biases, as features or times series are all related to the same original event. Similar approaches using adversarial examples for malware analysis which modify features instead of original binaries faces the inverse feature-mapping problem [47], [48].

Some of these issues depend on works from other related fields, such as ML for performance or FL for scalability. However, specificities of the FIDSs use case require more concrete research questions. Especially, the topics of security, trust, and resilience, are critical for a collaborative security use case.

4. Ensuring trust and personalization in FL-based intrusion detection

As introduced in Section 2, FL can be used in multiple cybersecurity settings and use cases, from distributed statistic inference [49] to authentication [50]. Even in the more specific context of intrusion detection (*i.e.* FIDS), numerous use cases are studied, *e.g.* IT networks [29], Industrial Internet of Things (IIoT) [15], or smart healthcare [51]. This variety in use cases comes with various data types, architectures, and constraints. Making knowledge transferable among heterogeneous use cases and data is a long-term goal for FIDSs (see Section 3.2).

However, heterogeneity is currently a major challenge for FL [52]. Therefore, we focus on a specific use case, namely FIDS in IT networks. We study the impact of heterogeneity on FL in this setting, and research solutions to mitigate it.

4.1. Use case definition

We consider a typical use case inspired from the industry, where actors invested in collaboration are organizations aiming at improving their local detection. We assume that each organization has interests in sharing information, but has highly sensitive data that cannot be shared. For example, Security Operation Centers (SOCs) perform security monitoring through the

processing of customer data (which can contain personal identifiable information) that cannot be shared. On the opposite, with the rise of cyber-criminal services [53], attackers tend to lead similar attacks against different information systems. Two SOCs in such situation would share their Indicators of Compromise (IoCs) or corresponding ML-models to detect all attacks after the first that succeed. In this context, FL can be used to train a FIDS model on a distributed dataset, while preserving the privacy of each organization. For instance, existing structures such as Information Sharing and Analysis Centers (ISACs) or inter-SOCs could benefit from such a system, which enables collaboration while protecting company secrets.

This setting is called cross-silo [52], as opposed to cross-device. It is worth noting that FIDSs do not exclude cross-device settings, for instance in endpoint detection [5]. However, cross-silo is more relevant for our use case, as it is more relatable for the industry, and it is easier to implement in a testing infrastructure. In cross-silo, fewer clients (10–1000) operate with more data each, as well as more powerful computation capabilities. Participants are also deemed more reliable in terms of availability, as the local learning process is performed on a dedicated device. Hence, it is not dependent on whether the hosting device (*e.g.* an employee computer) is turned on or off. Finally, organizations often operate with higher-stake privacy requirements, as they process data from their customers, their employees, and themselves.

We firstly also focus on horizontal federated learning (HFL), where participants have the same features, but different samples. In HFL, participants have similar objectives (*i.e.* ML tasks) and want to improve their models, but cannot build a centralized dataset due to privacy or legal concerns. This is the most common setting in FIDSs, as it serves the goal of improving behavior characterization, and having access to knowledge that cannot be inferred with only local data. We also start with unsupervised learning algorithms, as the presence of labels not guaranteed in real-world settings.

Figure 3 depicts an example topology, inspired by IT networks from the industry. We assume that each organization has a dedicated IT network, that might vary in terms of architecture, probe location, or services (see Section 4.2).

4.2. Experiment presentation

The chosen use case inherently highlights two of the seven open issues identified in the literature: heterogeneity and trust. Furthermore, as mentioned in Section 3.2, the lack of a dataset that is representative of this use case undermines existing works on the topic. Therefore, we first focus on two complementary tasks: (1) the creation of a representative, distributed dataset; and (2) the development of solutions to mitigate heterogeneity and provide trust between distributed participants.

To cope with the lack of appropriated dataset in the literature, we propose to build a new dataset generation platform with federation and distributed systems in mind. The platform relies on virtualization topologies to generate normal traffic [54], and defined attack scenarios to evaluate the detection performance. These topologies will be notably used to generate benign traffic and train local unsupervised learning algorithms—literature typically use autoencoders for this task [28], [38], [55]. The traffic generated at this step must be representative of a real IT-focused network with users, internal resources (such as file sharing and web applications), administration and supervision services, and access to the Internet.

We then aim to evaluate existing approaches on this more realistic and demanding use case. We expect most of them to yield less promising results, even when they claim to be able to deal with heterogeneity. On the other hand, FL literature abounds of works on heterogeneous data, and aggregation algorithms, such as FedProx [56] or Fed+ [57], might provide good results. In this case, proving empirically that such approaches are able to cope with the realistic heterogeneity of our use case remains a major contribution.

At the same time, we develop a new FL approach that deal with heterogeneity and trust. Section 5 presents strategies envisioned to achieve this goal. When considered as a collaboration system, FL is highly connected to research on trust and reputation systems. Hence, we plan to leverage existing works on the topic, such as [58], to provide a reputation-aware FL system. This approach will be evaluated on the new dataset, and compared to existing strategies.

4.2.1. Parameters

To measure the ways in which heterogeneity can manifest itself, we define varying parameters that are used to generate the topologies. This can be used to generate a worst-case scenario, where all the parameters are set differently in each topology.

In regard to the use case presented in Section 4.1, we consider the following parameters:

(i) **architecture** — the network architecture of the topology defines how services are interconnected, how the traffic is captured, and where data collection is performed. For example, a topology with a single main gateway which captures traffic, and several services on the same network, will produce a different dataset when compared with a star-shaped topology with multiple subnets. Appropriated metrics are required to characterize the impact of these differences, *e.g.* size (number of hosts, of subnets), mean number of hops between a service and the last gateway, and so on.

(ii) **services** — different services can rely on different protocols, and therefore generate different kind of data, with different behaviors. For example, a service using TCP will induce connection establishment, and therefore a lot of traffic back-and-forth, whereas something based on UDP will produce a more continuous stream of data. Therefore, different services (and protocols) might have different normal behaviors, causing heterogeneity among participants. The list of considered services must be adapted depending on the considered attack scenarios.

(iii) **maturity** — security practices vary between organizations, depending on their threat model, previous expertise, and budget. For example, a company might have a dedicated security team, and therefore be able to implement a more mature security policy, whereas a small company might not have the resources to do so. This parameter is important to consider, as it can impact the quality of the dataset, *e.g.* by having unseen attacks in the training data, supposed to be benign traffic.

(iv) **probe location** — while we assume that all participants extract the same features (due to HFL settings, see Section 4.1), the location of the probes can vary. In the same architecture, one collection point at the gateway, or distributed probes in each subnet, will produce different datasets.

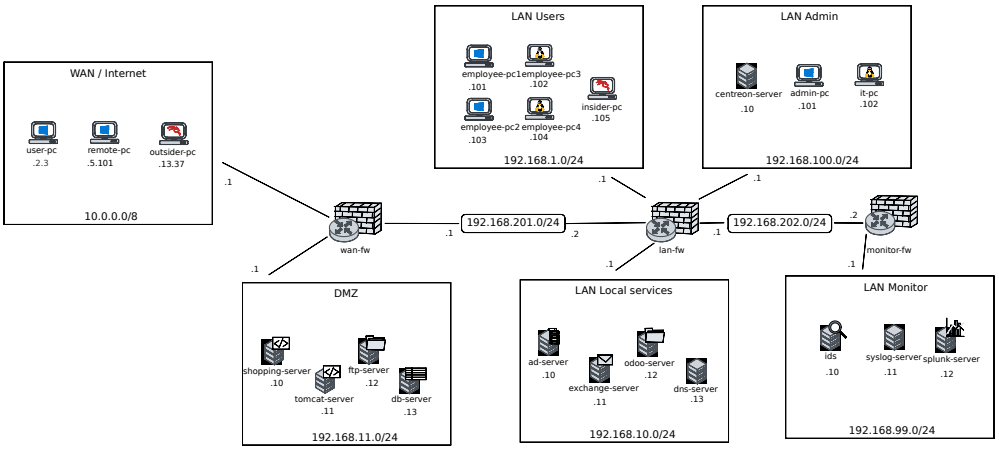


Figure 3: Example topology of a IT network

Furthermore, some services are required to build working topologies, such as NTP, DHCP, or DNS. We use a typical SME topology as a starting point for the experiments, which is presented in Figure 3. This topology is composed of a dozen of machines, divided between employee workstations, servers, etc. Variations of this topology will be also generated, and will be used as other participants.

4.2.2. Evaluation

To evaluate approaches on any dateset, we measure their ability to detect relevant attack scenarios. Typical attacks targeting such infrastructures are automated, like ransomware and botnets. A lone attacker doing recognition and enumeration in the network is also to be considered, but more advanced attacks such as APTs are out of the scope of this use case.

To make a rigorous evaluation, we rely on the state of the art to select attacks, such as the MITRE ATT&CK framework [59], or the most used attacks in dataset literature. The list of attacks that are considered on a given topology is deeply correlated to the services available in this topology. For instance, cache poisoning requires cache-based service in the topology, such as a DNS relay-server. On the other hand, organizations establish threat models depending on the services they host or use. Considered attacks may include: Distributed Denial of Service (DDoS) on the internet-facing web services, host and port scan,or web vulnerability exploitation (injections, ...).

Experiments are conducted on a private infrastructure. The test bed consist of three servers, two for the virtualization, and one for computation that will be dedicated to executing the required ML algorithms. While in real-world settings, the computation is performed on the participants' devices, we offload the computation to a dedicated server. This allows us to focus on the aggregation and trust aspects of the problem, and to avoid the complexity of the distributed computation. To enable sound experiments [60], we plan to provide access to all produced artifacts, including the dataset, the code, and the topology specifications, as well as to the testbed itself.

5. Discussion

As the literature shows, FIDSs falter at aggregating heterogeneous models. Therefore, the experiments detailed in section 4 provide means to measure and quantify the issue. While literature on FL contains works on dealing with heterogeneous clients, this is still an issue in the context of intrusion detection. Hence, we expect state-of-the-art approaches to perform rather poorly, when compared to the results obtained on a homogeneous dataset. We consider several approaches to cope with these issues.

First, metric-based model weighting could be used to give less importance to models that deviate too much from the others. Zhang *et al.* [18] use a centroid-distance weighting algorithm that cope with the heterogeneity in IIoT. More generally, weighting algorithms can help to merge only relevant models, depending on a set of defined metrics. Other metrics could be used to further tune the model aggregation, like a numerical estimation of how much information the model can bring.

Another related topic is the measurement of the training data's quality. In fact, as the learned model is an abstraction of the data upon which it was trained, low-quality data would lead to a low-quality model. Bringing a low-quality model in a federation could undermine every one's security. However, we must define what makes training data of quality, and how this quality can be measured. Furthermore, weighting models according to their quality means the system needs to be able to compare them, thus having access to the other's data.

Moreover, existing works on federated learning rely on participant clustering [58], [61] to improve model specialization. Often, these approaches aim at either reducing heterogeneity between clients, or detecting and excluding malicious participants. Part of the challenge here reside in the metrics to chose to build clusters. Aforementioned metrics such as data quality, or information estimation, could be also used in clustering. In the context of intrusion detection, the formation of clusters can have an indirect impact on participants' security.

Finally, the lack of information inherent to the abstract nature of the model is a major hurdle to estimate its value for aggregation. Therefore, we also consider adding metadata around models, describing what the model contains without giving out too much information about local data and configurations Such metadata would allow choosing which models one client is interested in, depending on its use case. We believe an *à-la-carte* model aggregation would help to cope with heterogeneity issues.

6. Conclusion

In this paper, we focused on the application of federated learning to the cybersecurity field. Federated learning increases trust among partners as they do not need to share data, contrarily to traditional ML-approaches. However, federated learning faces some limitations when federating models built on heterogeneous data. We therefore offer to address this issue within experiments conducted on a platform with appropriate generated datasets. This platform displays several use-cases with different partners in order to study aggregation-models parameters. We then outline envisioned solutions which will be subjects of future work.

These contributions can have a significant impact on increasing the security of organizations.

In fact, while existing research has addressed the privacy aspects of collaboration through federated learning and other privacy-preserving mechanisms, maintaining trust in heterogeneous collaboration is still a challenge. We showed that federated learning can help with the creation of a common trusted security model for systems that are inherently distributed. Our approach enables non-trusting parties to collaborate for the joint goal of increasing their cybersecurity without revealing critical internals.

References

[1] M. Catillo, A. D. Vecchio, A. Pecchia, and U. Villano, "A critique on the use of machine learning on public datasets for intrusion detection," in *Communications in Computer and Information Science*, Springer International Publishing, 2021.

[2] S. Goldwasser, M. P. Kim, V. Vaikuntanathan, and O. Zamir, "Planting undetectable backdoors in machine learning models," 2022. arXiv: 2204.06974 [cs.LG].

[3] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014. arXiv: 1412.6572 [stat.ML].

[4] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, and P. Faruki, "Network Intrusion Detection for IoT Security Based on Learning Techniques," *IEEE Communications Surveys & Tutorials*, 2019.

[5] G. Karantzas and C. Patsakis, "An Empirical Assessment of Endpoint Detection and Response Systems against Advanced Persistent Threats Attack Vectors," en, *Journal of Cybersecurity and Privacy*, 2021. (visited on 07/20/2021).

[6] T. D. Nguyen, S. Marchal, M. Miettinen, H. Fereidooni, N. Asokan, and A.-R. Sadeghi, "DÏoT: A Federated Self-learning Anomaly Detection System for IoT," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, IEEE, 2019.

[7] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *Computers & Security*, 2009.

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

[9] E. M. Campos, P. F. Saura, A. González-Vidal, J. L. Hernández-Ramos, J. B. Bernabe, G. Baldini, and A. Skarmeta, "Evaluating Federated Learning for Intrusion Detection in Internet of Things: Review and Challenges," en, *arXiv:2108.00974 [cs]*, 2021. (visited on 12/02/2021).

[10] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," en, *ACM Computing Surveys*, 2009. (visited on 03/20/2022).

[11] S. Rathore, B. Wook Kwon, and J. H. Park, "BlockSecIoTNet: Blockchain-based decentralized security architecture for IoT network," *Journal of Network and Computer Applications*, 2019.

[12] C. V. Zhou, C. Leckie, and S. Karunasekera, "A survey of coordinated attacks and collaborative intrusion detection," en, *Computers & Security*, 2010. (visited on 07/21/2021).

[13] ENISA, "Actionable Information for Security Incident Response," Tech. Rep., 2014.

[14] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th international conference on artificial intelligence and statistics*, A. Singh and J. Zhu, Eds., ser. Proceedings of machine learning research, PMLR, 2017.

[15] B. Li, Y. Wu, J. Song, R. Lu, T. Li, and L. Zhao, "DeepFed: Federated Deep Learning for Intrusion Detection in Industrial Cyber-Physical Systems," *IEEE Transactions on Industrial Informatics*, 2020.

[16] J. Ren, H. Wang, T. Hou, S. Zheng, and C. Tang, "Federated Learning-Based Computation Offloading Optimization in Edge Computing-Supported Internet of Things," *IEEE Access*, 2019.

[17] K. Bonawitz, V. Ivanov, B. Kreuter, *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, 2017.

[18] W. Zhang, Q. Lu, Q. Yu, *et al.*, "Blockchain-based Federated Learning for Device Failure Detection in Industrial IoT," *IEEE Internet of Things Journal*, 2020.

[19] U. Majeed and C. S. Hong, "FLchain: Federated Learning via MEC-enabled Blockchain Network," in *2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, IEEE, 2019.

[20] T. D. Nguyen, P. Rieger, H. Yalame, *et al.*, "FLGUARD: Secure and Private Federated Learning," en, *arXiv:2101.02281 [cs]*, 2021. (visited on 05/18/2021).

[21] M. Kim, O. Gunlu, and R. F. Schaefer, "Federated Learning with Local Differential Privacy: Trade-offs between Privacy, Utility, and Communication," 2021.

[22] A. Sinha, Z. Shen, Y. Song, H. Ma, D. Eide, B.-J. (Hsu, and K. Wang, "An Overview of Microsoft Academic Service (MAS) and Applications," en, in *Proceedings of the 24th International Conference on World Wide Web*, Florence Italy: ACM, 2015. (visited on 10/22/2021).

[23] M.-O. Pahl and F. X. Aubet, "All Eyes on You: Distributed Multi-Dimensional IoT Microservice Anomaly Detection," *14th International Conference on Network and Service Management, CNSM 2018, 1st Workshop on Segment Routing and Service Function Chaining*, 2018.

[24] Y. Zhao, J. Chen, D. Wu, J. Teng, and S. Yu, "Multi-Task Network Anomaly Detection using Federated Learning," en, in *Proceedings of the Tenth International Symposium on Information and Communication Technology - SoICT 2019*, Hanoi, Ha Long Bay, Viet Nam: ACM Press, 2019. (visited on 06/07/2021).

[25] W. Schneble and G. Thamilarasu, "Attack detection using federated learning in medical cyber-physical systems," en, 2019.

[26] B. Cetin, A. Lazar, J. Kim, A. Sim, and K. Wu, "Federated Wireless Network Intrusion Detection," en, in *2019 IEEE International Conference on Big Data (Big Data)*, Los Angeles, CA, USA: IEEE, 2019. (visited on 10/25/2021).

[27] S. A. Rahman, H. Tout, C. Talhi, and A. Mourad, "Internet of Things Intrusion Detection: Centralized, On-Device, or Federated Learning?" en, *IEEE Network*, 2020. (visited on 06/01/2021).

[28] Y. Chen, J. Zhang, and C. K. Yeo, "Network anomaly detection using federated deep autoencoding gaussian mixture model," in *Machine learning for networking*, S. Boumerdassi, É. Renault, and P. Mühlethaler, Eds., Cham: Springer International Publishing, 2020.

[29] Y. Sun, H. Ochiai, and H. Esaki, "Intrusion Detection with Segmented Federated Learning for Large-Scale Multiple LANs," en, in *2020 International Joint Conference on Neural Networks (IJCNN)*, Glasgow, United Kingdom: IEEE, 2020. (visited on 10/01/2021).

[30] Y. Fan, Y. Li, M. Zhan, H. Cui, and Y. Zhang, "IoTDefender: A Federated Transfer Learning Intrusion Detection Framework for 5G IoT," en, in *2020 IEEE 14th International Conference on Big Data Science and Engineering (BigDataSE)*, Guangzhou, China: IEEE, 2020. (visited on 10/04/2021).

[31] N. A. A.-A. Al-Marri, B. S. Ciftler, and M. Abdallah, "Federated Mimic Learning for Privacy Preserving Intrusion Detection," en, *arXiv:2012.06974 [cs]*, 2020. (visited on 10/25/2021).

[32] S. Kim, H. Cai, C. Hua, P. Gu, W. Xu, and J. Park, "Collaborative Anomaly Detection for Internet of Things based on Federated Learning," en, in *2020 IEEE/CIC International Conference on Communications in China (ICCC)*, Chongqing, China: IEEE, 2020. (visited on 10/25/2021).

[33] Q. Qin, K. Poularakis, K. K. Leung, and L. Tassiulas, "Line-Speed and Scalable Intrusion Detection at the Network Edge via Federated Learning," en, 2020.

[34] Z. Chen, N. Lv, P. Liu, Y. Fang, K. Chen, and W. Pan, "Intrusion Detection for Wireless Edge Networks Based on Federated Learning," en, *IEEE Access*, 2020. (visited on 10/25/2021).

[35] X. Hei, X. Yin, Y. Wang, J. Ren, and L. Zhu, "A trusted feature aggregator federated learning for distributed malicious attack detection," en, *Computers & Security*, 2020. (visited on 10/25/2021).

[36] K. Li, H. Zhou, Z. Tu, W. Wang, and H. Zhang, "Distributed Network Intrusion Detection System in Satellite-Terrestrial Integrated Networks Using Federated Learning," en, *IEEE Access*, 2020. (visited on 10/25/2021).

[37] S. I. Popoola, R. Ande, B. Adebisi, G. Gui, M. Hammoudeh, and O. Jogunola, "Federated Deep Learning for Zero-Day Botnet Attack Detection in IoT Edge Devices," en, *IEEE Internet of Things Journal*, 2021. (visited on 10/01/2021).

[38] Y. Qin and M. Kondo, "Federated Learning-Based Network Intrusion Detection with a Feature Selection Approach," en, in *2021 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*, Kuala Lumpur, Malaysia: IEEE, 2021. (visited on 10/04/2021).

[39] H. Liu, S. Zhang, P. Zhang, X. Zhou, X. Shao, G. Pu, and Y. Zhang, "Blockchain and Federated Learning for Collaborative Intrusion Detection in Vehicular Edge Computing," en, *IEEE Transactions on Vehicular Technology*, 2021. (visited on 10/04/2021).

[40] Y. Sun, H. Esaki, and H. Ochiai, "Adaptive Intrusion Detection in the Networking of Large-Scale LANs With Segmented Federated Learning," en, *IEEE Open Journal of the Communications Society*, 2021. (visited on 10/04/2021).

[41] X. Cao, Z. Li, H. Yu, and G. Sun, "CoFED: Cross-silo Heterogeneous Federated Multi-task Learning via Co-training," en, *arXiv:2202.08603 [cs]*, 2022. (visited on 02/25/2022).

[42] Z. Charles and J. Konecny, "Convergence and Accuracy Trade-Offs in Federated Learning and Meta-Learning," en, 2021.

[43] H. T. Nguyen, V. Sehwag, S. Hosseinalipour, C. G. Brinton, M. Chiang, and H. Vincent Poor, "Fast-Convergent Federated Learning," en, *IEEE Journal on Selected Areas in Communications*, 2021. (visited on 05/25/2022).

[44] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, IEEE, 2009.

[45] S. A. V. Jatti and V. J. Kishor Sontif, "UNSW-NB15: A Comprehensive Data set for Network Intrusion Detection Systems," *International Journal of Recent Technology and Engineering*, 2019.

[46] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," en, in *Proceedings of the 4th International Conference on Information Systems Security and Privacy*, Funchal, Madeira, Portugal: SCITEPRESS - Science and Technology Publications, 2018. (visited on 10/14/2021).

[47] F. Pierazzi, F. Pendlebury, J. Cortellazzi, and L. Cavallaro, "Intriguing properties of adversarial ml attacks in the problem space," 2019. arXiv: 1911.02142 [cs.CR].

[48] D. Park and B. Yener, "A survey on practical adversarial examples for malware classifiers," in *Reversing and Offensive-oriented Trends Symposium*, ACM, 2020. arXiv: 2011.05973 [cs.CR].

[49] J. R. Trocoso-Pastoriza, A. Mermoud, R. Bouyé, F. Marino, J.-P. Bossuat, V. Lenders, and J.-P. Hubaux, *Orchestrating Collaborative Cybersecurity: A Secure Framework for Distributed Privacy-Preserving Threat Intelligence Sharing*, en, 2022. (visited on 09/12/2022).

[50] M. Alazab, S. P. R M, P. M, P. Reddy, T. R. Gadekallu, and Q.-V. Pham, "Federated Learning for Cybersecurity: Concepts, Challenges and Future Directions," en, *IEEE Transactions on Industrial Informatics*, 2021. (visited on 12/02/2021).

[51] E. Ashraf, N. F. F. Areed, H. Salem, E. H. Abdelhay, and A. Farouk, "FIDChain: Federated Intrusion Detection System for Blockchain-Enabled IoT Healthcare Applications," en, *Healthcare*, 2022. (visited on 07/05/2022).

[52] P. Kairouz, H. B. McMahan, B. Avent, *et al.*, "Advances and Open Problems in Federated Learning," en, *arXiv:1912.04977 [cs, stat]*, 2021. (visited on 04/01/2022).

[53] E. Brumaghin, A. Khodjibaev, M. Thaxton, and A. Zobec, *Attackers leveraging dark utilities "c2aas" platform in malware campaigns*, 2022.

[54] A. Dey, B. Costé, É. Totel, and A. Bécue, "Realistic simulation of users for it systems in cyber ranges," 2021.

[55] G. d. C. Bertoli, L. A. P. Junior, A. L. d. Santos, and O. Saotome, *Generalizing intrusion detection for heterogeneous networks: A stacked-unsupervised federated learning approach*, en, 2022. (visited on 09/12/2022).

[56] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated Optimization in Heterogeneous Networks," en, *arXiv:1812.06127 [cs, stat]*, 2020. (visited on 09/20/2021).

[57] P. Yu, A. Kundu, L. Wynter, and S. H. Lim, "Fed+: A Unified Approach to Robust Personalized Federated Learning," en, *arXiv:2009.06303 [cs, math, stat]*, 2021. (visited on 01/31/2022).

[58] Z. Chen, P. Tian, W. Liao, and W. Yu, "Zero Knowledge Clustering Based Adversarial Mitigation in Heterogeneous Federated Learning," *IEEE Transactions on Network Science and Engineering*, 2021.

[59] B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels, A. G. Pennington, and C. B. Thomas, "Mitre att&ck™: Design and philosophy," MITRE CORP BEDFORD MA, Tech. Rep., 2018.

[60] R. Uetz, C. Hemminghaus, L. Hackländer, P. Schlipper, and M. Henze, "Reproducible and Adaptable Log Data Generation for Sound Cybersecurity Experiments," en, in *Annual Computer Security Applications Conference*, Virtual Event USA: ACM, 2021. (visited on 08/08/2022).

[61] Z. Li, T. Wang, and N. Li, *Differentially Private Vertical Federated Clustering*, en, 2022. (visited on 08/16/2022).